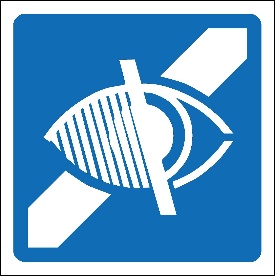


Cairo University

Faculty of Engineering

Department of Computer Engineering

**Pocket Lens**



A Graduation Project Report Submitted

to

Faculty of Engineering, Cairo University

in Partial Fulfillment of the requirements of the degree

of

Bachelor of Science in Computer Engineering.

**Presented by**

Ahmed Mohamed Ismail Moaz Mohamed El Sehbini

Mostafa Ashraf Ahmed Nader Youhanna Adib Khalil

**Supervised by**

Assoc. Prof. Dr. Mona Farouk

12/06/2023

All rights reserved. This report may not be reproduced in whole or in part, by photocopying or other means, without the permission of the authors/department.

**General Guidelines**

* The document is intended to be a template for the graduation project report in the department of Computer Engineering, Cairo University
* After reading this page, you have to remove it from the report
* Red texts should be removed and replaced by similar ones, related to your project
* The color of the cover page should be green
* Throughout your text, use the font type, font size, and spacing, as in this template. In general, Arial font should be used. Chapter headings should be of size 24. Sections should be of size 18, and regular text should be of size 12. Your text should be justified on both left and right sides.
* The reference list should be written using a font size of 10. Ensure that the references are written correctly and all fields are included. References should be ordered according to their appearance in the text “[1], [2], [3] … etc”
* The table of content is a tentative one. You could add more sections as required. However, the mentioned sections should be included in your report
* For the appendices, add any appendix you see necessary. Remove any appendix that is not applicable to your project. However, the feasibility study and user guide should be included
* Ensure that the report is clear and self-contained, such that any future interested reader could completely understand your project “to the extent of building another one similar to yours”
* Use figures as much as possible to clarify and enrich your discussion. You have to draw all figures yourself. Ensure that the figures are clear and their size is suitable.
* Any figure caption should be inserted below the figure. Figures within any chapter should be numbered starting from 1. For example, the first figure of chapter 2 should be “Figure 2.1”. Similarly, the fourth figure of chapter 3 should be “Figure 3.4”
* Any table caption should be inserted above the table. Tables within any chapter should be numbered starting from 1. For example, the first table of chapter 4 should be “Table 4.1”. Similarly, the seventh table of chapter 5 should be “Table 5.7”
* Copy and paste from any other source is not allowed by any shape. Even for the background knowledge, you have to use your own wording.
* The complete report should be submitted 48 hours before the final project demonstration day. Ensure that you would meet this deadline to avoid any late penalty

# Abstract

Even though many mobile devices today include accessibility features available for visually impaired and blind users, many of these users are reluctant to use them. This is because either the features are not very beneficial for the user or the interface is mainly designed for sighted people. The latter is caused by the fact that the main input and output methods on mobile devices are tactile or visual in nature. However, in recent years, there have been many innovative applications that assist VIB users in navigating their environment. Programmers have made use of technological advances regarding gyroscope sensors and vibration feedback to make communication possible.

The proposed system relies on input images and videos provided by the user’s device camera to allow daily life navigation without the need to use such sensors. It makes communication between VIB users and their devices possible using speech/text conversion techniques.

The approach that is followed to solve this problem is to use artificial intelligence to analyze images captured by the device's camera, and provide feedback to the user through speech synthesis. The output of this project is a mobile app that can run on both Android and iOS devices, and that can be customized according to the user's preferences and needs.

The summary of testing results shows that the app is effective, accurate, and reliable in performing the intended functions, and that it has a positive impact on the user's independence and quality of life.

# الملخص

على الرغم من أن العديد من الأجهزة المحمولة اليوم تتضمن ميزات للمستخدمين المكفوفين وضعاف البصر، إلا أن العديد من هؤلاء المستخدمين يترددون في استخدامها. هذا لأن الميزات ليست مفيدة جدًا للمستخدم أو أن الواجهة مصممة بشكل رئيسي للأشخاص ذوي البصر السليم. مشكلة الواجهات المصممة هي كون طرق الإدخال والإخراج الرئيسية على الأجهزة المحمولة تكون عن طريق اللمس و استخدام حاسة البصر. ومع ذلك، في السنوات الأخيرة، ظهرت العديد من التطبيقات المبتكرة التي تساعد المستخدمين المكفوفين وضعاف البصر في التنقل في بيئتهم. قام المبرمجون باستخدام التطورات التكنولوجية المتعلقة بحساسات الجايروسكوب والاهتزاز لجعل التواصل ممكنًا. النظام المقترح يعتمد على صور وفيديوهات إدخال يقدمها كاميرا جهاز المستخدم للسماح بالتنقل في الحياة اليومية دون الحاجة إلى استخدام مثل هذه الحساسات. يجعل التواصل بين مستخدمي VIB وأجهزتهم ممكانًا باستخدام تقنيات التحويل من/إلى نص/كلام.

الطريقة التى يتبعها النظام لحل هذه المشكلة هى استخدام الذكاء الإصطناعى لتحليل الصور التى يلتقطها كاميرا جهاز وتقديم ملاحظات للمستخدم من خلال توليف الكلام. ناتج هذا المشروع هو تطبيق جوال يعمل على كلاً من أجهزة Android و iOS، والذى يمكن تخصيصه وفقًا لتفضيلات واحتياجات المستخدم. تظهر ملخص نتائج اختبارات أن التطبيق فعال و دقیق و موثوق في أداء الوظائف المقصودة، وأن لديه تأثير إيجابى على استقلالية المستخدم و جودة حياته.

# ACKNOWLEDGMENT

We would like to express our sincere gratitude to Allah for giving us the opportunity and the strength to complete this graduation project.

We would also like to thank Dr. Mona Farouk, our supervisor and mentor, for her invaluable guidance, feedback and encouragement throughout this journey. She has been a source of inspiration and motivation for us, and we have learned a lot from her expertise and experience. We are truly grateful for her support and kindness.

Ahmed, Moaz, Mostafa and Nader

# Table Of Contents

[Abstract iii](#_Toc136835446)

[الملخص iv](#_Toc136835447)

[ACKNOWLEDGMENT v](#_Toc136835448)

[Table Of Contents vi](#_Toc136835449)

[List of Figures ix](#_Toc136835450)

[List of Tables xi](#_Toc136835451)

[List of Abbreviation xii](#_Toc136835452)

[List of Symbols xiii](#_Toc136835453)

[Contacts xiv](#_Toc136835454)

[Chapter 1: Introduction 1](#_Toc136835455)

[1.1 Motivation and Justification 1](#_Toc136835456)

[1.2 The Essential Question 1](#_Toc136835457)

[1.3 Project Objectives and Problem Definition 1](#_Toc136835458)

[1.4 Project Outcomes 2](#_Toc136835459)

[1.5 Document Organization 2](#_Toc136835460)

[Chapter 2: Market Visibility Study 3](#_Toc136835461)

[2.1 Targeted Customers 3](#_Toc136835462)

[2.2 Market Survey 3](#_Toc136835463)

[2.2.1 Competitive Project 1 3](#_Toc136835464)

[2.2.2 Competitive Project 2 3](#_Toc136835465)

[2.3 Business Case and Financial Analysis 3](#_Toc136835466)

[Chapter 3: Literature Survey 5](#_Toc136835467)

[3.1 Background on Topic 1 5](#_Toc136835468)

[3.2 Background on Topic 2 5](#_Toc136835469)

[3.3 Comparative Study of Previous Work 5](#_Toc136835470)

[3.4 Implemented Approach 5](#_Toc136835471)

[Chapter 4: System Design and Architecture 7](#_Toc136835472)

[4.1 Overview and Assumptions 7](#_Toc136835473)

[4.2 System Architecture 7](#_Toc136835474)

[4.2.1 Block Diagram 7](#_Toc136835475)

[4.3 Module 1 8](#_Toc136835476)

[4.3.1 Functional Description 8](#_Toc136835477)

[4.3.2 Modular Decomposition 9](#_Toc136835478)

[4.3.3 Stage 2: Creating and Training the CNN Model 11](#_Toc136835479)

[4.3.4 Stage 3: Post-processing and Text Output 15](#_Toc136835480)

[4.3.5 Design Constraints 19](#_Toc136835481)

[4.3.6 Other Description of Module 1 20](#_Toc136835482)

[4.4 Module 2 20](#_Toc136835483)

[4.4.1 Functional Description 21](#_Toc136835484)

[4.4.2 Modular Decomposition 22](#_Toc136835485)

[4.4.3 Design Constraints 22](#_Toc136835486)

[4.4.4 Other Description of Module 2 22](#_Toc136835487)

[4.5 Face Detection 23](#_Toc136835488)

[4.5.1 Functional Description 23](#_Toc136835489)

[4.5.2 Modular Decomposition 24](#_Toc136835490)

[4.5.2.1 Feature Extraction 24](#_Toc136835491)

[4.5.2.2 Feature Selection and Classifier Construction 24](#_Toc136835492)

[4.5.2.3 Detection 25](#_Toc136835493)

[4.5.3 Design Constraints 26](#_Toc136835494)

[4.6 Emotion Detection 26](#_Toc136835495)

[4.6.1 Functional Description 26](#_Toc136835496)

[4.6.2 Modular Decomposition 27](#_Toc136835497)

[4.6.3 Design Constraints 29](#_Toc136835498)

[4.7 Retial Product Identifier 29](#_Toc136835499)

[4.7.1 Functional Description 29](#_Toc136835500)

[4.7.2 Modular Decomposition 30](#_Toc136835501)

[4.7.3 Design Constraints 30](#_Toc136835502)

[4.8 Apparel Recommender 30](#_Toc136835503)

[Chapter 5: System Testing and Verification 31](#_Toc136835504)

[5.1 Testing Setup 31](#_Toc136835505)

[5.2 Testing Plan and Strategy 31](#_Toc136835506)

[5.2.1 Module Testing 31](#_Toc136835507)

[5.2.2 Integration Testing 31](#_Toc136835508)

[5.3 Testing Schedule 31](#_Toc136835509)

[5.4 Comparative Results to Previous Work 31](#_Toc136835510)

[Chapter 6: Conclusions and Future Work 32](#_Toc136835511)

[6.1 Faced Challenges 32](#_Toc136835512)

[6.2 Gained Experience 32](#_Toc136835513)

[6.3 Conclusions 32](#_Toc136835514)

[6.4 Future Work 32](#_Toc136835515)

[References 33](#_Toc136835516)

# List of Figures

[Figure ‎4‑1 the N/A that is found in the emnist dataset making them 26 classes 9](#_Toc136835517)

[Figure ‎4‑2 Each index represents one of the alphabets 10](file:///D:\University\Senior%202\GP\Github\Graduation-Project\Final%20Report\GP2\GP%20Book.docx#_Toc136835518)

[Figure ‎4‑3 The result of the trained module it can recognize most of the letters but mistake few that are similar like in this figure we have v mistaken for r and g mistaken for q hence the accuracy of this model is 93.2% 14](#_Toc136835519)

[Figure ‎4‑4 The resulting plot is a 26x26 grid representing the performance of the character recognition model on the EMNIST Letters dataset. The rows represent the true labels, and the columns represent the predicted labels. 15](#_Toc136835520)

[Figure ‎4‑5 When applying the below function we get the letter separated and padded so it can be easily detected by our model and recognized as the letter D 18](file:///D:\University\Senior%202\GP\Github\Graduation-Project\Final%20Report\GP2\GP%20Book.docx#_Toc136835521)

[Figure ‎4‑6 Face Detection Flow Chart 23](file:///D:\University\Senior%202\GP\Github\Graduation-Project\Final%20Report\GP2\GP%20Book.docx#_Toc136835522)

[Figure ‎4‑7 Examples of Harr-like features 24](file:///D:\University\Senior%202\GP\Github\Graduation-Project\Final%20Report\GP2\GP%20Book.docx#_Toc136835523)

[Figure ‎4‑8 Example of cascaded classifier 25](file:///D:\University\Senior%202\GP\Github\Graduation-Project\Final%20Report\GP2\GP%20Book.docx#_Toc136835524)

[Figure ‎4‑9 Example of an integral image. The sum of pixels in rectangle D equals to 4 - 3 - 2 + 1 25](file:///D:\University\Senior%202\GP\Github\Graduation-Project\Final%20Report\GP2\GP%20Book.docx#_Toc136835525)

[Figure ‎4‑10 Emotion Detection Flow Chart 27](file:///D:\University\Senior%202\GP\Github\Graduation-Project\Final%20Report\GP2\GP%20Book.docx#_Toc136835526)

[Figure ‎4‑11 Example of HOG feature 28](file:///D:\University\Senior%202\GP\Github\Graduation-Project\Final%20Report\GP2\GP%20Book.docx#_Toc136835527)

[Figure ‎4‑12 Example of ensemble of regression trees 28](file:///D:\University\Senior%202\GP\Github\Graduation-Project\Final%20Report\GP2\GP%20Book.docx#_Toc136835528)

[Figure ‎4‑13 Flow Chart of Retail Product Identifier 29](file:///D:\University\Senior%202\GP\Github\Graduation-Project\Final%20Report\GP2\GP%20Book.docx#_Toc136835529)

# List of Tables

**No table of figures entries found.**

# List of Abbreviation

|  |  |
| --- | --- |
| Abbreviation | Definition |
| AI | Artificial intelligence |
| HOG | Histogram of Oriented Gradient |
| RPI | Retail Product Identifier |
| VIB | Visually impaired and blind |
| WHO | World Health Organization |
| YOLO | You Only Look Once |
| UI | User Interface |
| UX | User Experience |
| CV | Computer Vision |
| GPU | Graphical Processing Unit |

# List of Symbols

# Contacts

**Team Members**

|  |  |  |
| --- | --- | --- |
| **Name** | **Email** | **Phone Number** |
| Ahmed Mohamed Ismail | [ahmedmoh123@hotmail.com](mailto:ahmedmoh123@hotmail.com) | +2 01028300083 |
| Moaz Mohamed El Sherbini | moaz5657@gmail.com | +2 01018711749 |
| Mostafa Ashraf Ahmed | moustafa.achraf@hotmail.com | +2 01003993985 |
| Nader Youhanna Adib | naderyouhanna@gmail.com | +2 01285003523 |

**Supervisor**

|  |  |  |
| --- | --- | --- |
| **Name** | **Email** | **Number** |
| Dr. Mona Farouk | mona\_farouk@eng.cu.edu.eg | +2 01005042029 |

This page is left intentionally empty

# Introduction

According to the WHO, around 2 billion people are visually impaired or blind. This is not a minority. Nevertheless, very little has been done to help them throughout their day. Mobile phones offer accessibility features for them, but these features are not enough for day-to-day activities.

The proposed system offers a mobile application that uses AI to help VIB people complete their daily tasks. It captures images from the user’s camera as input and gives the user feedback through a text-to-speech module.

## Motivation and Justification

VIB users are often put at a disadvantage regarding their visually able peers. Technological advancements have always been concerned with providing better and easier to use solutions. These efforts have been largely directed toward the use of sensors, which can in many are not available to every user.

Moreover, many of the applications that can be found in the market are not particularly easy to use. They often require some degree of tactile interaction, which VIB users will most probably not be able to provide. Some of these applications are designed to be used by sighted people alongside VIB users, which can come as impractical.

The before mentioned reasons led us to consider using AI and Machine Learning techniques to create a mobile application that can serve as an assistant to VIB people. We will be addressing these previous problems by rendering the contact between the application and the VIB user purely vocal as much as the desired features allow for it. In other words, the user will communicate with the chatbot through speech.

## The Essential Question

The essential question is how to use AI and Machine Learning techniques to create a mobile application that can serve as an assistant to visually impaired and blind (VIB) people. This is relevant to the Vision and Mission of the Faculty of Engineering at Cairo University as it aligns with their goal of using technological advancements to provide better and easier-to-use solutions for everyone.The proposed system aims to address previous problems by rendering the contact between the application and the VIB user purely vocal, allowing for easier communication and interaction.

## Project Objectives and Problem Definition

The problem being addressed is the disadvantage faced by visually impaired and blind (VIB) users in comparison to their visually able peers. The objective of the project is to use AI and Machine Learning techniques to create a mobile application that can serve as an assistant to VIB people. The application aims to address previous problems by rendering the contact between the application and the VIB user purely vocal, allowing for easier communication and interaction.

## Project Outcomes

The outcome of the project would be a mobile application that uses AI and Machine Learning techniques to serve as an assistant to visually impaired and blind (VIB) people.

## Document Organization

In this section, you have to give the organization of the report and a quick description of the following chapters.

# Market Visibility Study

The project market is an innovative virtual assistant designed specifically for visually impaired and blind individuals. This cutting-edge technology aims to improve the quality of life for those with visual impairments by providing them with a tool that can assist them in their daily lives. With its advanced features and user-friendly interface, the project market is set to revolutionize the way visually impaired and blind people interact with the world around them.

## Targeted Customers

The target customers for our mobile application are VIB individuals who are looking for a more accessible and intuitive way to interact with their smartphones. Our application aims to address the disadvantage that VIB users often face in comparison to their visually able peers. The application's purely vocal interaction allows for a more natural and convenient way for VIB users to communicate with their phones. Our target customers are those who seek a mobile application that is easy to use, reliable, and tailored to their specific needs, allowing them to access the same features and functionalities as their sighted counterparts.

## Market Survey

In this section, we will list the competetive products to our application. We will explore similar commercial tools and platform and discuss them. A subsection will be dedicated to each one of them.

### Smart Glasses of Envision

Envision smart glasses are designed to help VIB individuals read. They were built on the entreprise edition of Google Glasses and rely heavily on Artificial Intelligence. The glasses aim to articulate everyday visual information into speech. The features of the glasses include scanning text, scene description, light detection, cash recognition, colors detection, finding people, finding objects, teach a face and exploring. Our product is different from Envision’s glasses in the measure that it is an application, which means no hardware purchase is required by our customers. Moreover, no hardware maintenance is required. This cuts production costs but has a negative impact on performance, since we have no access to extra sensors.

### Google Lookout

Google’s Lookout is an application that can help people identify food labels as well as find objects in a room. It can also scan documents, money, and products.

Lookout uses computer vision and machine learning technology to assist people with low vision or blindness get things done faster and more easily. Lookout is available for free for Android devices on the Play Store. Using their phone’s camera, Lookout makes it easier to get more information about the world around people and do daily tasks more efficiently like sorting mail, putting away groceries, and more. After identifying objects in the scene, the application provides audio feedback about what it detects in the environment. The user can also customize the application's settings to receive specific types of feedback and adjust the volume and speed of the audio output. he application is also integrated with TalkBack, Google's screen-reading software, which enhances its accessibility for VIB individuals.

However, one potential limitation of Google Lookout is that its performance varies depending on lighting conditions, camera quality, and user proficiency. It also requires a stable internet connection. Moreover, the application is only available on Android devices, which may limit its accessibility to users who prefer other operating systems.

### Be My Eyes

Be My Eyes is another application that is intended to help VIB individuals to navigate the world. It connects visually impaired users with sighted volunteers. The volunteers then help the user get around via a live chat function. It also aims to be integrated in the future with OpenAI’s ChatGPT-4 and an AI powered volunteer that willprovide instantaneous identification, interpretation and conversational visual assistance for a wide variety of tasks.

To use the application, a visually impaired user can request assistance through a video call, and a sighted volunteer will answer the call and provide assistance in real-time by describing the visual surroundings or helping with a task. The volunteers are trained to provide assistance in a variety of areas, such as reading labels, identifying colors, or navigating unfamiliar environments.

The application also offers a specialized feature called "Specialized Help," which connects users with representatives from various partner organizations, such as Microsoft, Google, and the American Diabetes Association, who can provide assistance with specific issues related to their products or services.

One of the strengths of Be My Eyes is its ease of use and accessibility, as the application is designed to be simple and intuitive. It also provides a valuable service to VIB individuals by leveraging the power of technology and human connection.

However, one potential limitation of Be My Eyes is its reliance on volunteers, which can result in inconsistent availability and varying levels of expertise.

### Microsoft Soundscape

Microsoft Soundscape is an application built by the Enable Group in Microsoft Research. The Soundscape app is breaking barriers and opening up new possibilities for visually impaired people with voice-based navigation. Anyone can take this app on the go and enjoy the independence that comes with being able to explore the world on their terms. Using a stereo headset such as Air Pods, users can traverse new and old environments guided by a map delivered in 3D sound.

This product uses a combination of GPS, compass, and audio feedback to provide spatial information to the user. The application creates a detailed 3D audio map of the user's surroundings, with audio cues indicating the direction and distance of nearby landmarks, points of interest, and intersections. The user can also set audio beacons and markers to help them navigate to specific locations. This way, it provides an immersive experience designed to help VIB individuals.

However, Microsoft Soundscape is not as acuurate in crowded or complex environments where multiple sounds and audio may overlap. It also requires a stable internet connection and a smartphone with a GPS and compass. Furthermore, it is only available on iOS devices.

### Facing Emotions

Facing Emotions is an application developed by Huawei. It identifies the 7 basic human emotions of irritation, contempt, sorrow, fear, anger, surprise, and happiness. The app turns those emotions into unique sounds to help the visually impaired learn how the person on the other side of the conversation is feeling. The application is designed to help users improve their emotional intelligence and communication skills by providing real-time feedback on their facial expressions. However, the application is only available for Huawei smartphones, which may limit its accessibility to users who prefer other brands or operating systems.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Features Apps | Lookout | Envision | Soundscape | Facing Emotions |
| Chatbot |  |  |  |  |
| Scene Descriptor | Done |  | Done |  |
| Text reader | Done | Done |  |  |
| Emotion detector |  | Done |  | Done |
| Currency detector | Done |  |  |  |
| Products detector | Done |  |  |  |

## Business Case and Financial Analysis

The development of a mobile application VIB individuals presents a significant business opportunity in a growing market. According to the World Health Organization, there are approximately 285 million visually impaired individuals worldwide, with 39 million of them being blind. The demand for assistive technology and solutions is expected to increase as the population ages and the prevalence of visual impairments rises. The development of an accessible and intuitive mobile application can meet this demand and provide a valuable service to visually impaired individuals.

The financial analysis of the project will depend on various factors such as the development costs, marketing expenses, and revenue streams. The development costs will include the expenses related to the software development, designing, and testing of the application. The marketing expenses will include the cost of advertising, promotion, and distribution of the application. The revenue streams can be generated through various sources such as in-app purchases, subscriptions, or advertisements. However, our project is non-lucrative and we do not aim to generate revenue through it. Instead, our focus is on addressing the needs and challenges of VIB individuals by providing them with an accessible and intuitive way to interact with their smartphones. This approach aligns with the principles of social responsibility and inclusivity, and can help to improve the quality of life of VIB individuals.

## SWOT Analysis

|  |  |
| --- | --- |
| **Strengths** | **Weaknesses** |
| * AI chatbot * 24/7 availability * Privacy * Internet access not needed(maps can be saved offline) * Understand other people better by knowing their facial emoticons | * AI functions are not 100% accurate hence can identify wrong currencies or emotions * Lack of empathetic human care * Expertise need |
| **Opportunities** | **Threats** |
| * Improving humans life * Convenience * Spreading awareness | * Risk of over-depending on the technology * Data security |

# Literature Survey

Computer Vision (CV) is a field of AI that is interested in enabling machines to interpret, process, and understand visual data. Since our project depends significantly on CV, the first part of this section wil be dedicated to explaining some key concepts of AI that are pivotal for understanding this section and the project in general.

Some of the key techniques used in computer vision include machine learning, deep learning, image processing, pattern recognition, and computer graphics.

Image processing techniques include image filtering, which is used to enhance or suppress some features of an image, blur an image or highlight some specific features.

In a CV pipeline, most often comes next feature extraction, where the algorithm identifies and extracts important features from an image. These are usually corners, edges and textures.

Deep Learning, which is a subste of machine learning, is also one of the prominent techniques used in computer vision. Convolutional Neural Networks, or CNN’s for short, are a type of neural networks which uses the convolution operation.

## Comparative Study of Previous Work

In this subsection, we will conduct a comparative study of previous work that has been done in the field. This will help us to understand the existing research and identify gaps that we can address in our own study.

To begin with, we will review the literature on the topic and identify the key studies that have been conducted in this area.

### Virtual Assistant for blind people

This project proposes to use AI, ML, Image, and Text recognition to assist people who are blind or visually impaired. The concept is realized using and Android mobile App that includes features such as voice assistant, image recognition, currency recognition, e-book and chatbot. It is a visual-based project consisting of few main components such as a camera, raspberry Pi, Sensors, Microphones and vibrators mountain together.

### A Smartphone-Based Mobility Assistant Using Depth Imaging for Visually Impaired and Blind

In this research, they made use of a mobile phone with a depth camera function for obstacle avoidance and object recognition. It includes a mobile app that is controlled using voice and gesture controls to assist in navigation. The proposed system gathers depth values from 23 coordinate points that are analyzed to determine whether an obstacle is present in the head area, torso area, or ground area, or is a full body obstacle. In order to provide a reliable warning system, the research detects outdoor objects within a distance of 1.6 m. Subsequently, the object detection function includes a unique interactable feature that enables interaction with the user and the device in finding indoor objects by providing an audio and vibration feedback, and users were able to locate their desired objects more than 80% of the time.

### An insight into smartphone-based assistive solutions for visually impaired and blind people: issues, challenges, and opportunities

The paper reviewed research avenues in smartphone-based assistive technologies for blind people, highlighted the need for technological advancements, accessibility-inclusive interface paradigm, and collaboration between medical specialists, computer professionals, usability experts and domain users to realize the potential of ICT-based interventions for blind people. It analyzes a comprehensive review of the issues and challenges for visually impaired and blind people with the aim to highlight the benefits and limitations of the existing techniques and technologies.

### Blind- Sight: Object Detection with Voice Feedback

Image classification techniques are used to identify the features of the image and categorize them into their appropriate class. The text description of the recognized object will be sent to the Google Text-to-Speech API using the gTTS package. Voice feedback on the 1st frame of each second will be scheduled as an output to help the visually impaired hear what they cannot see. The following Modules are implemented: Image Capture, Feature Extraction, Object Classification and Speech synthesis

### Emotion Detection Algorithm Using Frontal Face Image

This paper proposes an emotion detection algorithm using a frontal facial image. There are three stages: image processing, facial features extraction and emotion detection. In image processing stage, the face region and facial component is extracted by using fuzzy color filter, virtual face model, and histogram analysis method. The features for emotion detection are extracted from facial component in facial feature extraction stage. In emotion detection stage, the fuzzy classifier is adopted to recognize emotion from extracted features. The modules used are image processing, facial features extraction and emotion detection.

### Clothing matching for visually impaired persons

As we all know matching clothes is one of the important steps when deciding what to wear but since visually impaired people face difficulties when it comes to color, this paper focuses on recognizing clothing patterns in four categories (plaid, striped, patternless, and irregular) and identifies 11 clothing colors. A camera mounted upon a pair of sunglasses is used to capture clothing images. The clothing patterns and colors are described to blind users verbally. This system can be controlled by speech input through microphone.

## Implemented Approach

Conclude this chapter by this section stating the approach chosen from those reviewed, **but more important your justification why you chose this approach** along with any modifications added to the approach.

Notice, you may be implementing several techniques however you must illustrate the general framework for your approach.

# System Design and Architecture

This chapter represents the main body of your project. It should describe the project in full details. This chapter should answer the questions: “what has been done?” and “how it has been done?”. As such, the steps you went through to realize the project should be highlighted and properly discussed. Your scientific approaches and methodologies should be clarified. The discussion should adopt a logical flow starting from the whole block diagram, to coarse modules, and finally to fine modules. While writing this chapter, try to give as much details as possible, such that an interested reader could easily replicate your work and improve it.

In this space, before the first section, write an introductory paragraph on how you design and build your project

## Overview and Assumptions

In this section, introduce how you design you system and develop its underlying architecture. Any employed assumptions should be clearly enumerated and justified.

## System Architecture

The architecture of your system should be given in this section. This architecture should be first represented as a block diagram (subsection 5.2.1), which clarifies different project modules and the connections between them. You may add more subsections to properly explain your design. If possible, flowcharts are better included to ensure that the big picture and the interaction between different modules are very clear to the reader. Thereafter, each module should have a separate subsequent section to clearly describe and discuss it.

### Block Diagram

Draw the block diagram of your architecture and generally discuss its modules. After reading this subsection, interested audience should have understood the big picture of your system design and architecture. The interaction between modules should also be conveyed in this subsection

## Text Reader

For visually impaired individuals, accessing textual information in daily life can be challenging. The text recognition module we've developed addresses this issue by leveraging advanced deep learning teckhniques to accurately extract text from images. By integrating this module into assistive technologies such as screen readers and smartphone applications, we can significantly improve the quality of life for the visually impaired. Our module's adaptability enables it to recognize text in various languages, fonts, sizes, and orientations, making it a versatile tool for enhancing accessibility in diverse contexts. Overall, the text recognition module offers a powerful solution for breaking down barriers faced by the visually impaired and fostering a more inclusive society.

### Functional Description

The text recognition module is designed to recognize and extract text from images using a 3-layered Convolutional Neural Network (CNN) architecture trained on the EMNIST dataset. The primary function of this module is to facilitate the conversion of image-based text into a machine-readable format, enabling various applications such as optical character recognition, document digitization, and assistive technologies for the visually impaired.

The EMNIST dataset, an extension of the popular MNIST dataset, comprises handwritten characters from multiple languages, providing a rich and diverse source of data for training the text recognition module. This ensures that the module is capable of recognizing a wide range of characters and text styles.

The 3-layered CNN architecture consists of the following layers:

1. Convolutional Layer: This layer is responsible for detecting local features in the input image, such as edges and corners, by applying a series of convolutional filters. It uses a ReLU activation function to introduce non-linearity and improve learning efficiency.

2. Pooling Layer: This layer reduces the spatial dimensions of the data by applying a max-pooling operation, which selects the maximum value from each local region. This process helps to reduce the computational complexity and improve the model's ability to recognize features regardless of their position in the image.

3. Fully Connected Layer: This layer acts as a classifier that takes the output of the previous layers and maps it to the appropriate character class. The softmax activation function is used in this layer to produce probability scores for each character class.

The text recognition module utilizes this 3-layered CNN architecture to process the input image and generate predictions for each character present in the image. The module then converts these predictions into a machine-readable format, such as a string or an array of characters, thereby completing the text recognition process.

In conclusion, the text recognition module provides an efficient and robust solution for extracting text from images using a 3-layered CNN architecture trained on the EMNIST dataset. Its ability to recognize a wide range of characters and text styles makes it a valuable tool for various applications in both industry and assistive technology sectors.

### Modular Decomposition

This section provides an in-depth discussion of the three main stages of the text recognition module: preprocessing the data, creating and training the CNN model, and creating bounding boxes for letter detection in real documents.

#### Stage 1: Preprocessing and Data Augmentation

The first stage of the text recognition module involves several steps to preprocess the EMNIST dataset and prepare it for training the 3-layered CNN architecture. The following steps are carried out in this stage:

1. Removing N/A: The EMNIST dataset may contain instances where character labels are missing or marked as not applicable (N/A). These instances can negatively impact the model's training as they do not provide meaningful information for learning. To ensure a clean and accurate dataset, all instances with N/A labels are removed during the preprocessing stage.

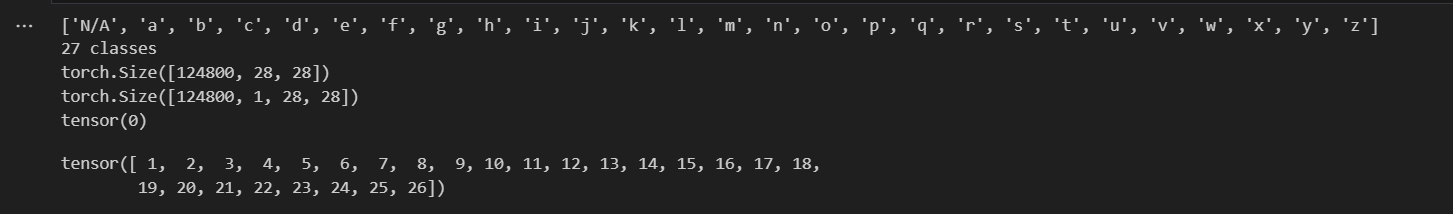


Figure ‑ the N/A that is found in the emnist dataset making them 26 classes

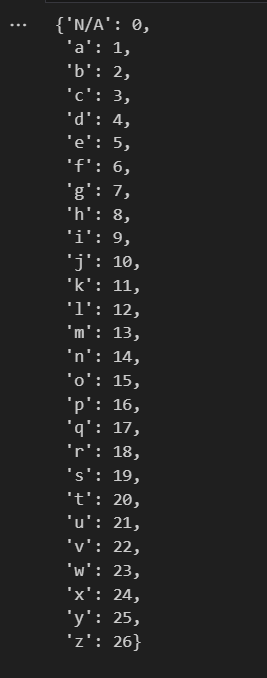


Figure ‑ Each index represents one of the alphabets

2. Normalization: The pixel values in the images are normalized to a range of 0 to 1 by dividing each pixel value by 255. This process helps to improve the numerical stability of the algorithm and can lead to faster convergence during model training.

3. Data Augmentation: In order to make the model more robust and invariant to various transformations, data augmentation techniques are applied to the EMNIST dataset.

These techniques include:

- Rotation: Images are rotated by a random angle within a specified range (e.g., -15 to 15 degrees). This process ensures that the model can recognize characters even if they appear at different orientations in the input images.

- Lighting: The brightness and contrast of the images are adjusted by applying random scaling factors. This step helps the model become more resilient to different lighting conditions, improving its ability to recognize text in real-world scenarios.

4. Reshaping and Grayscale Conversion: The input images in the EMNIST dataset are converted to grayscale, as color information is not essential for character recognition. Additionally, the images are reshaped to match the input dimensions expected by the CNN architecture.

5. Dataset Splitting: The preprocessed EMNIST dataset is divided into three subsets: training, validation, and testing. The training set is used to train the model, while the validation set is employed for hyperparameter tuning and model selection. The test set is reserved for evaluating the final model's performance on unseen data. Typically, a stratified sampling approach is used to ensure that each subset has a balanced distribution of the different character classes.

By following these detailed preprocessing and data augmentation steps, the text recognition module ensures that the EMNIST dataset is adequately prepared for training the 3-layered CNN. This comprehensive approach helps improve the model's robustness, generalization capabilities, and overall performance in recognizing text from images.

#### Stage 2: Creating and Training the CNN Model

The second stage of the text recognition module involves creating and training a CNN model for character recognition. The provided code defines a custom neural network called emnistnet that uses the PyTorch framework. Let's break down the code and discuss each layer and the activation functions used.

**emnistnet CNN Architecture**

The emnistnet architecture consists of three convolutional layers, three batch normalization layers, three fully connected layers, and two dropout layers.

1. Convolutional Layers:

- conv1: The first convolutional layer has 1 input channel (grayscale image), 16 output channels (feature maps), and a 3x3 kernel with padding of 1. This layer is responsible for detecting low-level features in the input images, such as edges and corners.

- conv2: The second convolutional layer has 16 input channels, 32 output channels, and a 3x3 kernel with padding of 1. This layer detects higher-level features, such as textures and patterns, by combining the low-level features from the previous layer.

- conv3: The third convolutional layer has 32 input channels, 64 output channels, and a 3x3 kernel with padding of 1. This layer captures even more complex features, thereby improving the model'sability to recognize characters.

2. Batch Normalization Layers:

- bnorm1, bnorm2, and bnorm3: These layers are applied after each convolutional layer. Batch normalization helps improve model training by normalizing the activations of each layer and reducing internal covariate shift. This leads to faster convergence and improved generalization performance.

3. Fully Connected Layers:

- fc1: The first fully connected (linear) layer takes an input of size 3 \* 3 \* 64 and outputs 128 units. This layer is responsible for combining the high-level features learned by the convolutional layers to form a more abstract representation.

- fc2: The second fully connected layer takes an input of size 128 and outputs 64 units. This layer further refines the abstract representation.

- fc3: The third fully connected layer takes an input of size 64 and outputs 26 units, which corresponds to the number of classes (letters) in the EMNIST dataset. This layer is responsible for classifying input images into one of the 26 letter classes.

4. Dropout Layers:

- dropout1 and dropout2: These layers are applied after the first and second fully connected layers, respectively. Dropout is a regularization technique that helps prevent overfitting by randomly dropping out (i.e., setting to zero) a fraction of the units during training. In this case, the dropout rate is set to 0.5.

**Activation Functions and Training**

In the forward method of the emnistnet class, the ReLU activation function is replaced with the Leaky ReLU activation function. Leaky ReLU is an improved version of the ReLU function that allows for a small, non-zero gradient when the input is negative. This helps alleviate the "dying ReLU" problem, where a large number of ReLU neurons become inactive and stop learning during training.

The training process is carried out using the function2trainmodel function. This function trains the model for 30 epochs using the Adam optimizer with a learning rate of 0.0005. The learning rate scheduler ReduceLROnPlateau is used to reduce the learning rate when the test loss plateaus, which helps improve convergence and avoid overshooting the optimal weights.

During each epoch, the model's weights are updated using the backpropagation algorithm to minimize the categorical cross-entropy loss, which measures the difference between the true labels and the predicted probabilities. The training error, test error, train loss, and test loss are recorded for each epoch, allowing for performance evaluation and model selection.

In summary, the provided code defines a custom CNN architecture called emnistnet for character recognition using the PyTorch framework. The model consists of convolutional layers, batch normalization layers, fully connected layers, and dropout layers, with Leaky ReLU activation functions. The model is trained using the Adam optimizer, a learning rate scheduler, and the categorical cross-entropy loss. The training process is carried out for 30 epochs, and the performance is evaluated using the train and test errors and losses.

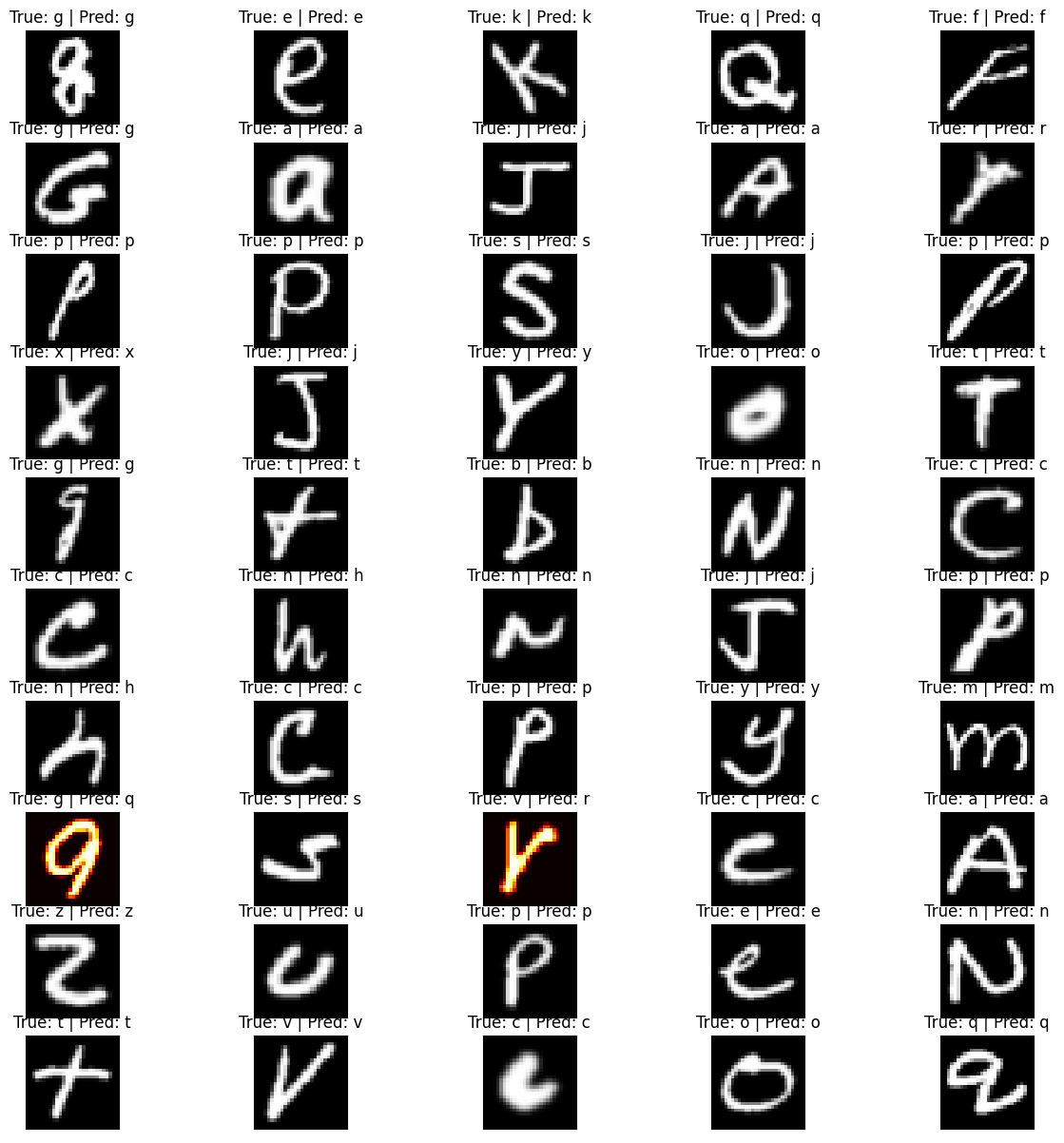


Figure ‑ The result of the trained module it can recognize most of the letters but mistake few that are similar like in this figure we have v mistaken for r and g mistaken for q hence the accuracy of this model is 93.2%

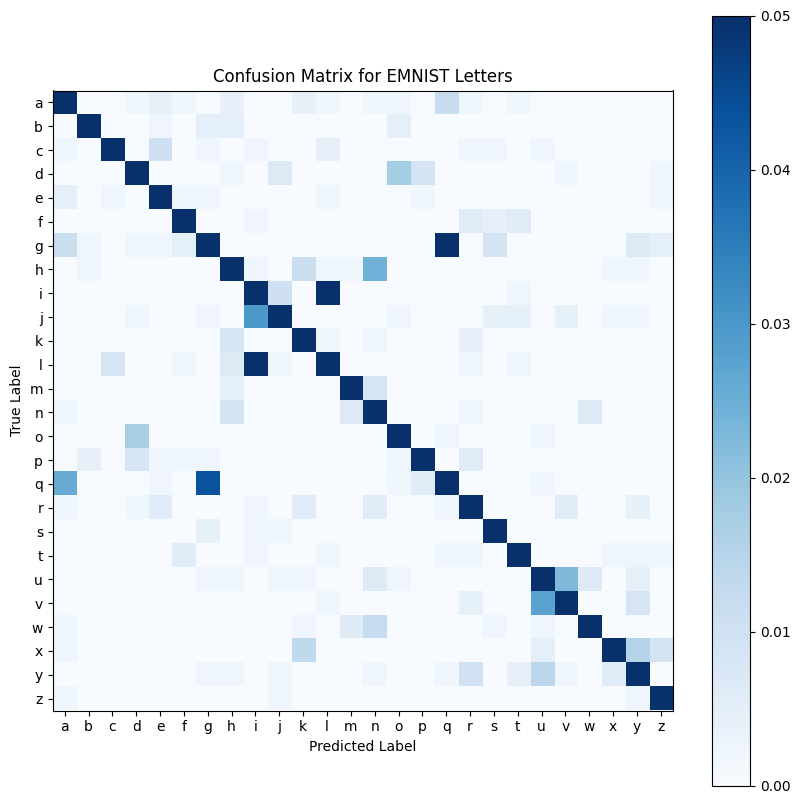


Figure ‑ The resulting plot is a 26x26 grid representing the performance of the character recognition model on the EMNIST Letters dataset. The rows represent the true labels, and the columns represent the predicted labels.

#### **Stage 3: Post-processing and Text Output**

The last stage of the text recognition module involves post-processing the images and predicting the characters using the trained CNN model. The provided code includes several functions that work together to achieve this goal:

1. preprocess\_document\_image: This function takes an image file path as input and preprocesses the image by converting it to grayscale and applying binary inversion and thresholding using Otsu's method. The result is a binary image where the text is white and the background is black.

2. add\_padding: This function takes a binary image and an optional padding value (default is 2) and adds padding around the image. This is useful for ensuring that the characters do not touch the borders of the image, which can help improve character recognition.

3. segment\_lines and segment\_characters: These functions segment the preprocessed image into lines and characters, respectively. They use morphological operations (dilation) and contour detection to find the bounding boxes of the lines and characters in the image. The resulting bounding boxes are sorted by their position to maintain the order of the text.

4. skeletonize: This function takes a binary image and skeletonizes the text, which can help improve character recognition by simplifying the text's structure and reducing the impact of noise.

5. resize\_character: This function takes a character image, resizes it to a specified size (default is 28x28), and adds padding around the image. This is necessary because the CNN model expects input images to have a size of 28x28.

6. recognize\_character: This function takes a character image and a trained CNN model as input and predicts the character using the model. The input character image is preprocessed, resized, and converted to a PyTorch tensor before being passed to the model. The model's output is a probability distribution over the 26 letter classes, and the predicted character is the class with the highest probability. The function also displays the character image using Matplotlib.

7. extract\_text\_from\_document\_image: This is the main function that combines all the steps mentioned above. It takes an image file path and a trained CNN model asinput and outputs the recognized text from the document image. The function first preprocesses the input image and then segments it into lines and characters. For each character image, it calls the recognize\_character function to predict the character using the trained model. The recognized characters are concatenated to form the output text, with line breaks added between the lines of text. The function returns the output text and the last recognized character as a tuple.

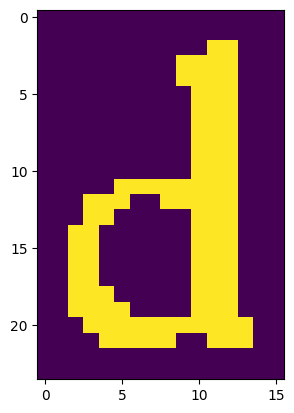


Figure ‑ When applying the below function we get the letter separated and padded so it can be easily detected by our model and recognized as the letter D

The **extract\_text\_from\_document\_image** function is called with the path to the input document image and the trained CNN model. It first preprocesses the image using preprocess\_document\_image and then segments the lines using segment\_lines. For each line, it segments the characters using segment\_characters and then recognizes each character using recognize\_character. The recognized characters are concatenated to form the output text, with line breaks added between the lines of text. The function returns the output text and the last recognized character as a tuple.

In summary, this stage of the text recognition module involves post-processing the images, segmenting the document into lines and characters, and predicting the characters using the trained CNN model. The final output is the recognized text from the document image.

### Design Constraints

While the described approach for text recognition using a CNN model on the EMNIST Letters dataset is effective, there are some constraints and limitations:

1. Limited character set: The model is trained on the EMNIST Letters dataset, which only includes uppercase English letters (26 classes). It lacks support for lowercase letters, digits, punctuation, and special characters. To recognize a more comprehensive set of characters, the model would need to be trained on a more diverse dataset.

2. Sensitive to image preprocessing: The accuracy of the text recognition depends heavily on the quality of the image preprocessing and segmentation steps. If the input image is not adequately preprocessed, or if the segmentation of lines and characters is not accurate, the overall performance of the text recognition system may be degraded.

3. Inability to handle distorted or noisy text: The model may struggle with recognizing text that is distorted, noisy, or written in unusual fonts, as the EMNIST dataset mostly contains clean and normalized character images. To improve the model's robustness to these variations, additional training data with diverse fonts, distortions, and noise levels could be included.

4. Lack of context: The approach recognizes individual characters independently without considering the context of the surrounding characters. This may lead to recognition errors that could be avoided by incorporating a language model that considers the statistical properties of the language, such as n-grams or more advanced models like RNNs and Transformer-based models.

5. No support for different languages: This approach is designed for English text recognition. To support other languages, the model would need to be trained on a dataset that includes characters and writing systems from those languages.

6. Computational complexity: The CNN model and the image processing steps can be computationally intensive, especially for large images or real-time applications. Optimizations and more efficient models may be required for deployment on resource-constrained devices or in real-time scenarios.

7. Lack of rotation and scale invariance: The model may not perform well when dealing with text that is rotated or scaled differently from the training data. To handle such variations, additional data augmentation or alternative approaches like spatial transformer networks could be employed.

### Other Description of Module 1

The text recognition module uses a CNN model trained on the EMNIST Letters dataset to recognize uppercase English letters in document images. It involves three stages: data preprocessing and augmentation, model training, and post-processing. The module preprocesses the dataset and augments it for better performance. The CNN learns to map input character images to letter classes. Finally, the input document images are preprocessed, segmented into lines and characters, and fed into the trained model to predict the text. While effective, the module has limitations such as sensitivity to preprocessing, inability to handle distorted/noisy text, lack of context awareness, and constraints to uppercase English letters. Improvements can be made by incorporating additional training data, advanced models, and language models.

## Module 2

The currency recognition model, which utilizes kNN with histogram, texture, and ORB features, plays a significant role in improving the quality of life for visually impaired individuals. Handling and identifying currency denominations can be a challenging task for people with visual impairments, as they rely heavily on touch and other non-visual cues to differentiate between banknotes and coins.

By providing a reliable and accessible solution, the currency recognition model empowers visually impaired users with greater independence and confidence in their daily financial transactions. This can have a profound impact on their overall well-being and integration into society. Here are some key benefits of this model for visually impaired individuals:

1. Enhanced Autonomy: The model enables visually impaired users to manage their finances independently, without relying on others for assistance. This fosters a sense of self-reliance and contributes to their autonomy in performing daily tasks.

2. Reduced Risk of Fraud: The model can help protect visually impaired individuals from potential fraud or exploitation by accurately identifying currency denominations. This ensures that they receive the correct change during transactions and helps to prevent any financial losses due to dishonest practices.

3. Increased Confidence: Being able to handle financial transactions effectively boosts the confidence of visually impaired people. This increased self-assurance can positively impact other aspects of their lives, such as social interactions and career opportunities.

4. Ease of Use: The currency recognition model can be integrated into user-friendly applications or devices, making it easily accessible for visually impaired users. With simple interfaces and audio feedback, these tools can be tailored to suit the needs and preferences of visually impaired individuals.

5. Inclusion and Accessibility: By providing an effective currency recognition solution, the model promotes financial inclusion and accessibility for visually impaired people. This can lead to a more inclusive society where people with disabilities have equal opportunities to participate in economic activities.

In conclusion, the currency recognition model using kNN with histogram, texture, and ORB features is of great importance to visually impaired individuals. It enhances their autonomy, reduces the risk of fraud, increases confidence, and promotes inclusion and accessibility. By addressing a crucial aspect of daily life, this innovative solution contributes to a better quality of life for people with visual impairments.

**4.4.1. Functional Description**

The currency recognition model for visually impaired people is designed to identify different currency denominations accurately and efficiently. It follows a series of steps, from preprocessing the dataset to extracting features and finally, applying the kNN algorithm for classification. Here's a step-by-step functional description of the model:

1. Preprocessing: Before feature extraction, the model applies preprocessing techniques to the input images. This step may include resizing, denoising, and normalization, among other methods, to enhance the images and ensure consistency across the dataset. Preprocessing helps improve the overall performance of the model by reducing noise and variations that may adversely affect feature extraction and classification.

2. Feature Extraction: After preprocessing, the model extracts three types of features from the images: histogram features, texture features, and ORB features. Histogram features capture the color distribution, texture features describe patterns and structural information, and ORB features provide rotation, scale, and illumination invariant information. By combining these features, the model creates a comprehensive feature vector representing each currency image.

3. kNN Classification: Once the feature vectors are obtained, they are used as input for the k-Nearest Neighbors (kNN) algorithm. The kNN classifier works by comparing the feature vector of an input image to the feature vectors of known currency images in the training dataset. It identifies the 'k' nearest neighbors (where 'k' is a user-defined parameter) and assigns the input image to the majority class among these neighbors. This process allows the model to accurately recognize different currency denominations based on their features.

In summary, the currency recognition model for visually impaired people follows a systematic process of reading the dataset, preprocessing images, extracting histogram, texture, and ORB features, and applying the kNN algorithm for classification. This approach enables the model to accurately and efficiently recognize various currency denominations, providing vital assistance to visually impaired users in their daily financial transactions.

**4.4.2. Modular Decomposition**

#### First Stage: Data Preprocessing

In the currency recognition model, the first stage focuses on data preprocessing. This stage plays a critical role in preparing the input dataset and ensuring that the images are in a suitable format for subsequent feature extraction. The provided code contains several image manipulation techniques intended to augment the dataset and enhance the model's robustness. The following is a detailed overview of the code:

1. Rotation and Scaling: The functions rotate(img, angle) and scale(img, scale) are responsible for applying rotation and scaling transformations to the input images, respectively.

2. Noise Addition: The noisy(noise\_typ, image) function introduces various types of noise to the input image based on the specified noise\_typ. Supported noise types include Gaussian noise, salt and pepper noise, Poisson noise, and speckle noise.

3. Blurring: The blur(img, blur\_type) function applies different blurring techniques to the input image according to the blur\_type parameter. The available blur types comprise average blur, Gaussian blur, median blur, and bilateral blur.

4. Lighting Adjustment: The lighting(img, lighting type) function modifies the lighting conditions of the input image based on the lighting type parameter. The lighting adjustments encompass brightness, contrast, gamma correction, histogram equalization.

In summary, the data preprocessing stage involves applying a range of transformations, noise, blur, and lighting adjustments to the input images. These techniques contribute to creating an augmented dataset with diverse angles, lighting conditions, scales, and noise levels. This stage serves as a crucial foundation for the subsequent feature extraction and classification stages in the currency recognition model.

#### Second Stage: Feature Extraction

The second stage of the currency recognition model for visually impaired people involves extracting features from the preprocessed images. This stage is critical as it generates a comprehensive feature vector that effectively represents each currency image and enables the model to classify different denominations accurately. The feature extraction process comprises three steps:

1. Histogram Features: In this step, the model extracts color information from the currency images by creating histograms. A histogram is a graphical representation of the distribution of color values in an image. It provides insight into the overall color composition and intensity patterns, which are essential attributes for currency recognition. The histograms are computed for each color channel (e.g., Red, Green, and Blue) and then concatenated to form a single feature vector.

2. Texture Features: To capture the structural and textural details of the banknotes, the model extracts texture features using methods such as the Gray-Level Co-occurrence Matrix (GLCM) or Local Binary Patterns (LBP). These methods analyze the spatial relationships between neighboring pixels and encode the presence of various patterns, such as edges, corners, and textures. Texture features provide crucial information about the unique patterns present in different currency denominations, enhancing the model's classification capabilities.

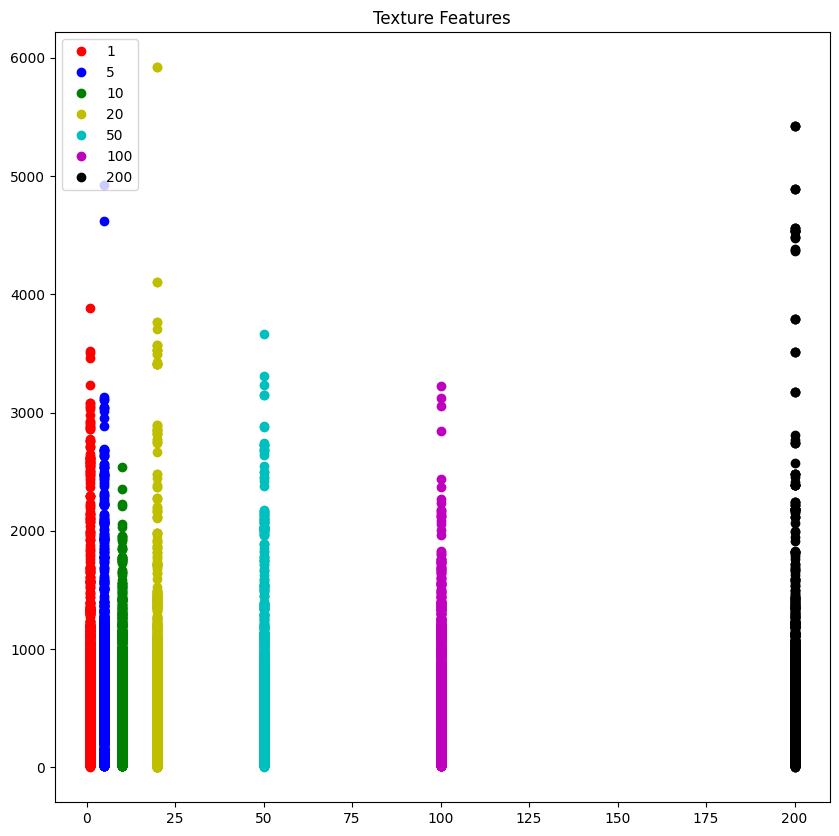


Figure ‑ GLCM of each currency which shows difference in texture for each cuurency

3. ORB Features: The Oriented FAST and Rotated BRIEF (ORB) algorithm is employed to extract keypoint and descriptor information from the currency images. The ORB method is a fast and efficient feature detector and descriptor extractor that is invariant to rotation, scale, and illumination changes. It identifies distinctive points (keypoints) in the images and computes a binary descriptor for each keypoint, capturing the local patterns around them. These keypoints and their corresponding descriptors serve as an essential input for the classification stage.

After extracting the histogram, texture, and ORB features, the model combines them to create a comprehensive feature vector for each currency image. This feature vector is then utilized as input for the k-Nearest Neighbors (kNN) algorithm during the classification stage. The second stage of feature extraction is vital as it enables the model to capture various visual attributes of the banknotes, ultimately leading to accurate currency recognition and improved assistance for visually impaired users.

#### Third Stage: Dimensionality Reduction using PCA

After the feature extraction stage, the currency recognition model proceeds to the third stage, which involves dimensionality reduction using Principal Component Analysis (PCA). The high-dimensional feature vector obtained from the previous stage may contain redundant information and increase computational complexity. PCA helps to address these issues by transforming the original feature space into a lower-dimensional space while retaining most of the information.

1. Standardize the Feature Data: Before applying PCA, it is important to standardize the feature data to ensure that each feature contributes equally to the analysis. Standardization scales the feature values to have a mean of 0 and a standard deviation of 1.

2. Determine the Optimal Number of Principal Components: To find the best number of principal components, calculate the explained variance ratios for each component and plot the cumulative explained variance against the number of components. From the plot, you can visually inspect the point at which the cumulative explained variance reaches your desired threshold (e.g., 95%). The `n\_components` variable will contain the best number of components based on the threshold.

3. Apply PCA with Optimal Number of Components: Perform PCA using the determined number of components (`n\_components`). This step transforms the high-dimensional feature vector into a lower-dimensional representation, where the principal components capture the most important variations in the data.

4. Update the Feature Vector: Replace the original high-dimensional feature vector with the lower-dimensional representation obtained from PCA. This compact representation retains the most relevant information while reducing computational complexity and mitigating the risk of overfitting.

In this third stage, PCA effectively reduces the dimensionality of the feature data, leading to a more efficient and robust currency recognition model. The lower-dimensional representation serves as input for the k-Nearest Neighbors (kNN) algorithm during the classification stage, enabling accurate currency recognition and enhanced assistance for visually impaired users.

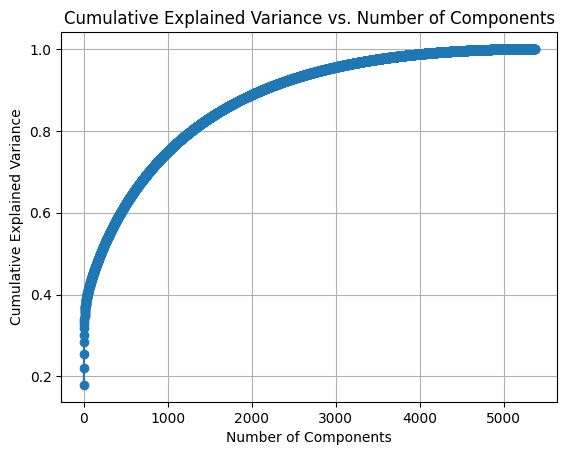


Figure ‑ dataset components before PCA transformation

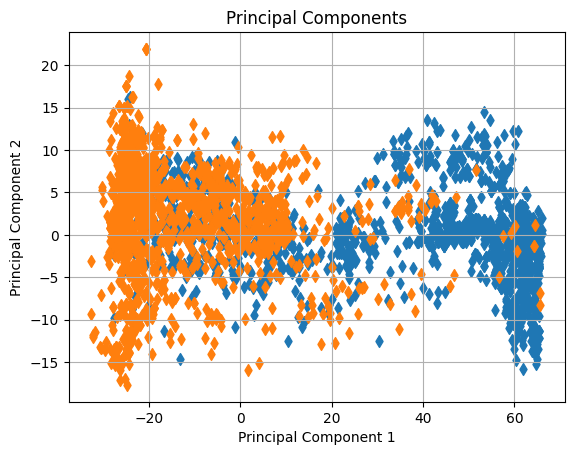


Figure ‑ Shows the difference between 5 EGP and 10 EGP components value after PCA

#### Fourth Stage of the: kNN Classification

The fourth and final stage of the currency recognition model for visually impaired people involves classifying the preprocessed images using the k-Nearest Neighbors (kNN) algorithm. This stage is crucial as it leverages the comprehensive feature vectors obtained from the feature extraction stage to accurately recognize different currency denominations. The kNN classification process consists of the following steps:

1. Training the kNN Classifier: Using the feature vectors extracted from the training dataset, the kNN classifier is trained to recognize various currency denominations. Unlike other machine learning algorithms, kNN does not require an explicit training phase; instead, it "memorizes" the training data, which serves as a reference for making predictions on new, unseen images.

2. Choosing the Value of 'k': The 'k' in kNN refers to the number of nearest neighbors the algorithm considers when making a classification decision. This value must be carefully selected to balance the trade-off between underfitting and overfitting. A smaller 'k' value may result in a model that is sensitive to noise and outliers, while a larger 'k' value may lead to a model that is too generalized and less accurate. Cross-validation techniques, such as k-fold cross-validation, can be employed to find the optimal value of 'k'.

3. Distance Metric: The kNN algorithm relies on a distance metric to compute the similarity between the feature vector of an input image and those of the training dataset. Common distance metrics include Euclidean distance, Manhattan distance, and Minkowski distance. The choice of the distance metric depends on the nature of the feature vectors and can significantly impact the classification performance.

4. Classification: To classify a new, unseen currency image, the kNN algorithm calculates the distance between its feature vector and those of the training dataset. It then identifies the 'k' nearest neighbors, i.e., the 'k' training samples with the smallest distances. The input image is assigned to the majority class among these neighbors, resulting in the recognized currency denomination.

In summary, the third stage of the currency recognition model for visually impaired people involves classifying preprocessed images using the k-Nearest Neighbors algorithm. This stage effectively leverages the comprehensive feature vectors obtained from the feature extraction stage to accurately recognize various currency denominations. By employing kNN classification, the model provides valuable assistance to visually impaired users in their everyday financial transactions and promotes financial independence.

**4.4.3. Design Constraints**

When implementing the currency recognition model using k-Nearest Neighbors (kNN) with ORB, histogram, and GLCM features, there are specific constraints that may affect the design, performance, and efficiency of the model. Addressing these constraints is essential for creating an effective currency recognition solution for visually impaired users. Some of the key constraints associated with this combination of features and kNN are:

1. Feature Integration: Combining the ORB, histogram, and GLCM features into a single feature vector for kNN classification requires careful consideration. The integration method should maintain the distinct characteristics of each feature type while ensuring the resulting feature vector is suitable for kNN distance calculations. Designers should explore various feature concatenation or fusion techniques to achieve an optimal integration that preserves the information content of each feature type.

2. Feature Dimensionality: Combining multiple feature types (ORB, histogram, and GLCM) can result in high-dimensional feature vectors, which may lead to increased computational complexity and memory requirements for the kNN algorithm.

3. Feature Scaling: Since ORB, histogram, and GLCM features have different value ranges and distributions, it is essential to scale or normalize the features before applying the kNN algorithm. Proper feature scaling ensures that all features contribute equally to the classification process and prevents features with larger value ranges from dominating the distance calculations.

4. Computational Complexity: As mentioned earlier, kNN can be computationally intensive, particularly when dealing with high-dimensional feature vectors and large datasets. Combining ORB, histogram, and GLCM features may exacerbate this issue.

5. Memory Requirements: The memory requirements of kNN may be increased when incorporating multiple feature types, as the algorithm stores the entire training dataset for classification. Designers should explore strategies like data compression, feature selection, or instance selection to reduce memory usage and accommodate devices with limited memory, such as smartphones or wearable devices.

6. Robustness: The model's robustness to real-world challenges like lighting conditions, image quality, scale, rotation, and occlusions depends on the effectiveness of the chosen features (ORB, histogram, GLCM) and the kNN classifier. Ensuring the model's robustness may require fine-tuning the feature extraction parameters, selecting an appropriate distance metric, and optimizing the value of 'k' for the kNN algorithm.

By considering and addressing these constraints during the implementation of the currency recognition model using kNN with ORB, histogram, and GLCM features, designers can develop an accurate, efficient, and robust solution that caters to the needs of visually impaired users and promotes their financial independence.

## Face Detection

The face detection algorithm used is based on the Viola-Jones algorithm. The reason this algorithm was used is that it is a very fast system, running at 14 frames per second, even though its accuracy is not the greatest.

### Functional Description

The algorithm consists of mainly four steps:

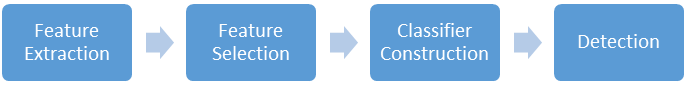
1. Feature extraction: The algorithm uses Haar-like features, which are simple rectangular patterns that capture the contrast between adjacent regions in an image. These features are computed efficiently using an integral image representation.

2. Feature selection: The algorithm uses a machine learning technique called AdaBoost to select a small subset of features that are most relevant for face detection. This reduces the computational cost and improves the accuracy of the algorithm.

3. Classifier Construction: The algorithm builds a cascade of classifiers, each of which is composed of a linear combination of features selected by AdaBoost. The cascade is designed to reject non-face regions quickly, while passing face regions to the next stage.

4. Detection: The algorithm scans the input image at multiple scales and locations, applying the cascade of classifiers to each sub-window. If a sub-window passes all stages of the cascade, it is marked as a face region.

Each step will be discussed in detail in the next part.



Training Phase

Figure ‑ Face Detection Flow Chart

### Modular Decomposition

As mentioned in the above part, the algorithm is based on four main steps. In each of the following paragraphs, we will discuss each step in detail.

#### Feature Extraction

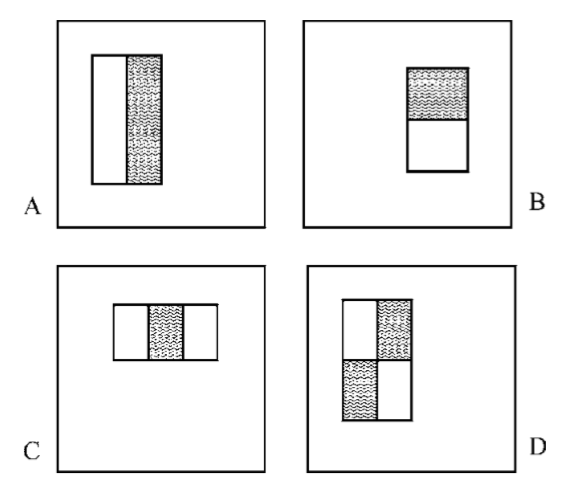
The first step that needs to be done is creating the Haar-like feature that the algorithm will select from for face detection. Haar-like features are rectangular patterns that capture the contrast between adjacent regions in an image. The Haar-like features used are: two-rectangle, three-rectangle, and four-rectangle features. The two-rectangle features are two adjacent rectangles that either have the same height and different widths or have the same width but different heights. The three-rectangle features are similar to the two-rectangle features, but the difference is that there are three adjacent rectangles instead of two. The four-rectangle feature consists of four rectangles that are diagonal to each other. The features are created on a 24x24 image. This will create 162,336 features. Those features are saved and used for training.

Figure ‑ Examples of Harr-like features

#### Feature Selection and Classifier Construction

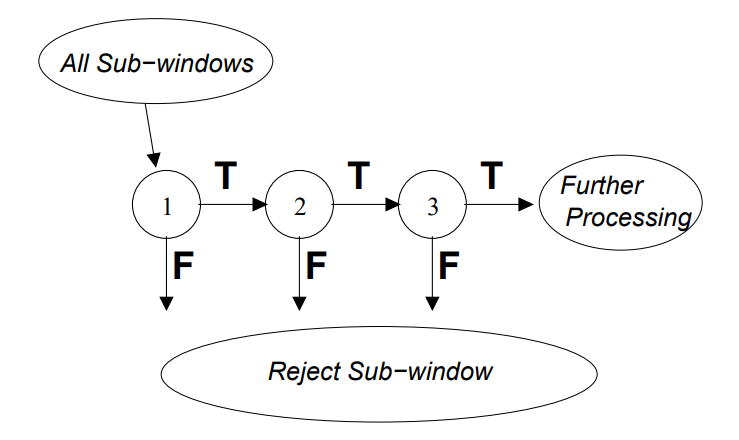
Since each one of those features is a weak classifier, Adaboost is used to select the most relevant Haar-like features and combine them into a strong classifier that can accurately detect faces in the image. The final output of the Adaboost is a list of weighted weak classifiers that can act as one strong classifier. This classifier is composed of twenty-five stages. A region in the image is considered a face if it passes all twenty-five stages. If one of the regions does not pass any of the stages, the region is immediately discarded and another region is considered. The reason for using stages is to reject non-face regions rapidly, thus reducing computation.

Figure ‑ Example of cascaded classifier

#### Detection

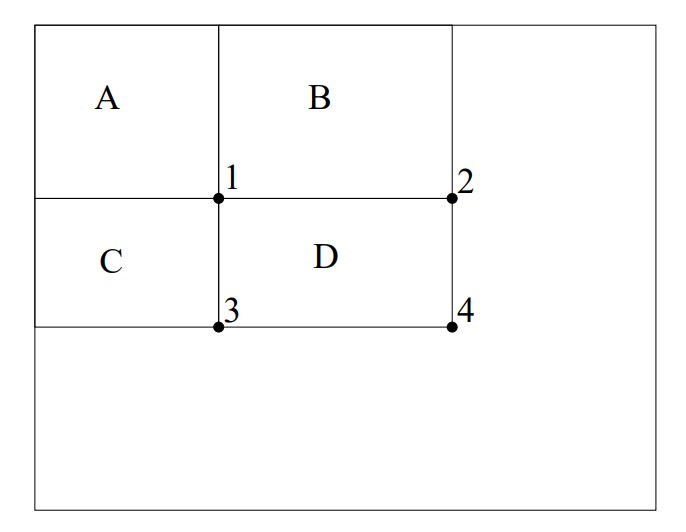
After creating the cascaded classifier, the module is ready for face detection. The input image is first preprocessed. The preprocessing includes normalization and calculating the integral image. An integral image is a data structure that allows for fast and efficient computation of sum of pixel values in a rectangular region of an image. It is also known as a summed-area table or a cumulative distribution table. The integral image is computed by adding up the pixel values along the rows and columns of the original image, such that each element of the integral image is equal to the sum of all the pixels above and to the left of it in the original image.

Figure ‑ Example of an integral image. The sum of pixels in rectangle D equals to 4 - 3 - 2 + 1

A sliding window is used to traverse the input image. Each sub-window is passed to the cascaded classifier, and if it passes all the stages, the region is considered a face. An image pyramid is formed by scaling down the input image by a scale factor. This is done to detect faces with different sizes.

### Design Constraints

Even though the algorithm is a very rapid face detection algorithm, it has its drawbacks. The main drawback of the algorithm is that faces need to be upright and well illuminated in order to be detected.

Another constraint of the algorithm is that it needs a very large and labelled dataset of face and non-face images. This can be difficult because the non-face images should not contain any faces, so they need to be checked manually.

Finally, the training phase of the algorithm takes a considerable amount of time. This is caused by two things: the very large dataset and the fact that Adaboost cannot be run in parallel. One could argue that this is not a design constraint. However, if a modification is needed and retraining is required, this will be a very time-consuming process.

## Emotion Detection

For individuals who are VIB, nonverbal cues such as facial expressions can be challenging to interpret, making it difficult to understand the emotions of others. Emotion detection technology has the potential to enhance the communication and social interactions of VIB individuals by providing them with a tool to better recognize and respond to the emotional states of others. By using advanced algorithms and machine learning techniques, emotion detection technology can analyze facial expressions to identify emotional states. This can help VIB individuals to better understand and respond to the emotions of others, leading to improved communication and stronger social connections.

### **Functional Description**

The emotion detection module first starts by detecting any faces in the input image. If no faces are detected, the module is immediately terminated. In the event that a face is found, the module continues.

Once a face is detected, the module starts extracting facial landmarks from the face. Facial landmarks are specific points on a face that are used to identify and track various features and expressions. These landmarks are the (x, y) coordinates of key points on the face, such as the corners of the eyes, the tip of the nose, and the corners of the mouth. Here, the facial landmarks are used to analyze the facial expression.

After the extraction of facial landmarks from the face in the image, those landmarks are used to predict the emotion of the face. The prediction is done using a random forest classifier. The random forest classifier is trained on a labeled dataset of human faces expressing various emotions.

Figure ‑ Emotion Detection Flow Chart

### Modular Decomposition

The face detection part of the module uses the HOG feature descriptor. The HOG algorithm works by dividing an image into small regions called cells, and computing a histogram of gradient directions for each cell. The histograms are then normalized and concatenated to form a feature vector that represents the shape and appearance of the image. Then, these descriptors are used with a linear classifier to detect faces.

The classifier is trained on a large dataset of face images and is capable of detecting human faces in a variety of lighting conditions and orientations. The model uses a sliding window approach to scan an input image at multiple scales and locations, looking for regions that contain facial features.

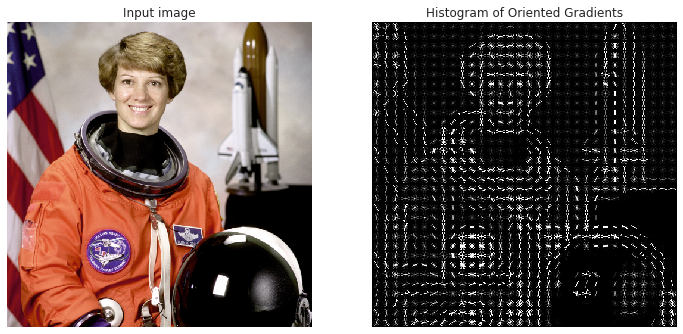


Figure ‑ Example of HOG feature

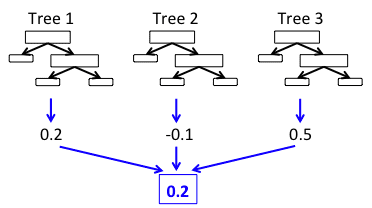
 As for the facial landmark extractor, the model uses a machine learning model that has been trained on a large dataset of annotated images to learn how to locate the landmarks. The model consists of an ensemble of regression trees that split the image into smaller regions and predict the offset of each landmark from the center of the region. The function combines the predictions of all the trees to obtain the final landmark coordinates.

Figure ‑ Example of ensemble of regression trees

Last but not least, the emotion detection uses a random forest classifier to classify the emotion of the face based on the facial landmarks passed on from the previous stage.

### Design Constraints

One of the main drawbacks of this design is that it cannot detect whether someone is suppressing their emotions. Moreover, detection of fake emotions is not possible.

## Retial Product Identifier

Retail Product Identifier (RPI) is designed to aid VIB people in shopping. RPI uses the camera of a smartphone to scan the product and provide audio feedback about its name and brand. RPI aims to empower VIB people with more information and convenience when shopping for their needs and preferences.

### Functional Description

The module consists of two main parts:

1. The first part is product detection. It detects the type of retail product. And returns a string that contains the type of products found. Then, it passes those product images to the next stage.
2. The second part is logo detection. Just like humans, it knows the brand of the retail product from the logo. It also returns a string that contains the logo detected for each item.

Figure ‑ Flow Chart of Retail Product Identifier

### Modular Decomposition

The first part of the module that detects the product type is based on YOLO algorithm. YOLO is a state-of-the-art object detection algorithm that can detect and classify multiple objects in an image with high speed and accuracy. Unlike traditional object detection methods that use a sliding window approach or region proposal networks, YOLO divides the input image into a grid of cells and predicts bounding boxes and class probabilities for each cell. YOLO also uses a single neural network to perform the detection task, which makes it faster and more efficient than other methods that use multiple networks or stages.

The second stage is also based on YOLO architecture. As it proved to be superior in terms of speed and accuracy. Due to the lack of a sufficient dataset, data augmentation has been used on the dataset to compensate for that.

### Design Constraints

Since YOLO is based on a neural network, it needs a very large dataset. This presented difficulties in the training process.

## Apparel Recommender

## Face Recognition using Eigenfaces

The Eigenfaces algorithm is a technique for facial recognition that is based on the machine learning concept of Principal Component Analysis (PCA). It is one of the earliest techniques that have been developed for facial recognition, having been developed in the 1980s by Sirovich and Kirby. Since then, it has become one of the most popular techniques for facial recognition.

### Functional Description

This module is designed to enable users to recognize the faces of their friends. It accepts an image of a person's face as input and provides the user with either the name of the person if it is recognized, or a message indicating that the person was not detected. Due to the way this algorithm works, which will be detailed in the following subsections, adding an new unkown face in the database for future recognitions would require training to be done on the whole dataset again, which is impractical. This is why we have taken the freedom of defining the inputs and outputs of the module as described above.

### Modular Description

The basic idea behind the algorithm is to represent facial images as linear combinations of a small number of characteristic feature vectors, called “Eigenfaces”. These Eigenfaces represent the proncipal components, or Eigenvectors, of the distribution of facial images in a training set. We calculate them by performing Principal Component Analysis (PCA).

PCA is a mathematical feature extraction and selection technique that is often used in machine learning algorithms. The technique analyzes a set of training data and identifies the most important features, or “principal components”. These principal components describe the variability in the dataset. In our case, these principal components represent the features or pixels that help the most in distinguishing between faces.

Now we will explain the working of the algorithm. First, the facial images are normalized by subtracting them from the average or mean face, which is also calculated from the training set. The images are then preprocessed by using smoothing filters. This helps in noise reduction and in enhancing the most important features.

A picture containing screenshot, x-ray film

Description automatically generatedNext, the normalized images are used to create the covariance matrix: it represents the statistical relationship between the different pixel values or features in the image. The covariance matrix is a square matrix that summarizes important relationships in the data. It captures the important data variations and provides insight as to how different features change together.

After calulcatin the covariance matrix, its Eigenvectors and Eigenvalues are calculated. These are calulcated using linear algebra. The eigenvectors represent the directions in which the data varies the most, while the eigenvalues indicate the amount of variance along each eigenvector. Each Eigenvector corresponds to one Eigenvalue. Therefore, the Eigenvectors that have the highest Eigenvalues represent the directions in which data varies the most, which corresponds to the features that are best for differentiating betweent the different classes, or in our case faces. Based on some threshold, whether it be on the number of Eigenvectors or the minimum value of the variance, some Eigenvectors are selected while others are discarded.

All of what has been described is the steps taken in the training phase. To recognize a new face, the algorithm projects the face onto the Eigenfaces and calculates the distance between the projected face and the Eigenfaces in the dataset. The face is then classified with the closest match. Any classifier at this point can be used. In our case, we used the K-Neares Neighbours (KNN) with K = 3.

Graphical user interface, application

Description automatically generatedAs for the dataset, we used our own faces. Each of us captured some photos with different angles, expressions and different lighting conditions. We used these photos to train the algorithm and got impressive results: the algorithm recognizes us in most of our other testing photos. Measuring accuracies here would not be exactly meaningful because of the small size of the dataset. However, we have tried with another dataset, the Olivetti dataset, and got a 97% accuracy. We used 400 images of 40 people where each image is 64 x 64 pixels. We used 320 images for training and 80 images for testing.

Text

Description automatically generated

One point of weakness of this algorithm is that the images must be close ups of faces. The algorithm therefore did not perform well for datasets where faces where not the main element in the image (15% accuracy for the LFW dataset).

This is expected of the Eigenfaces algorithm, because it does not try to extract certain features from the images that would make the difference between one face and the other. Rather, it makes the assumption that the faces distribution over the whole image space (which is all possible combinations of pixels all possible width x height images) is not random. Based on this assumption, the algorithm aims to calculate the Eigenvectors (called Eigenfaces) that best desribe the distribution of face images over the images space.

## Frontend

For the frontend part of our application, we used the Android development framework Flutter, developed by Google. This tool allows the building of high-performance, cross-platform mobile apps for Android and iOS. It uses the Dart programming language, which was also developed by Google, to build mobile apps with reactive styles. Our choice of the framework was determined by many factors including the ease of use of Flutter. One of its main features is hot reload, which allows developers to see changes in their code almost instantly without having to build the whole app again. This greatly speeds up the development process and makes it easier to iterate on designs and features.

### Functional Description

This module aims to be the interface with the user, and so it should be easy to use and intuitive. The navigation inside the application should be entirely voice-based so that the application is easy to use by VIB individuals. It should also be fast and provide for real-time change between the different application’s modules.

A voice-based navigation means that the user can go from one module to the other by nothing other than voice commands. On the other hand, the application also needs to communicate information to the user, such as the output of our algorithms like the description of scenes or the classification of clothes and money denominations. This is why and equally important aspect must be present in the application: audio feedback.

Despite this, the UI is beautiful and allows for people whose visual impairment is not total blindness to enjoy a simple yet efficient experience.

Each module in the project corresponds to a Widget in the Flutter application, which maintains modularity, clarity, and ease of use. The users can customzie the application’s settings to adjust the rate of speech of the audio feedback.

The application also aims to be user-friendly even for users who have limited technology experience.

There are of course some limitations to the application. The application requires a stable and continuous access to internet. As unreliable a this may seem, a compromise must be made with the application’s performance. Due to the very nature of AI and CV, the algorithms must run on a highly efficient platform, such as a server with GPUs. To achieve this, constant connection to the internet must be made.

### Modular Description

**4.10.2.1 StartUp Page**

The StartUp page is the first thing that a user may see when they open the application. It displays our logo and takes about a second before transitioning to the next page: the HomePage.

**A picture containing text, screenshot, computer, operating system

Description automatically generated**

**4.10.2.2 The HomePage**

The Home Page is the central hub where users can navigate to all other modules. Upon loading, the home page will greet the user with a friendly message, providing a brief explanation of how to navigate the app. The greet message is played as sound so that VIB individuals may not need assitsance even in the first time they install the application. The greeting asks the user to speak any module name to navigate to it, only by voice command. On the background is displayed too a semi-transparent image of our logo.

Users do not have to command the module name exactly to be directed to it. Instead, they can simply express their intent and the chatbot module Alan, which will be discussed further in this report, will understand what they want. For example, commands like ‘Scene Descriptor’, ‘Describe’, ‘Describe what’s in front of me’, ‘What is in front of me ?’, ‘What can you see ?’ and many more will direct the user to the Scene Descriptor module.

At this stage, the user can also directly speak with Alan. They can chat with Alan and ask for guidance in about navigation or about other things. Examples to questions that users can ask Alan are: ‘What’s the weather like today ?’, ‘Can you read this for me ?’ and ‘How much is this ?’.

**A close-up of a logo

Description automatically generated with medium confidence**

**4.10.2.3 Menu Drawer**

A screenshot of a phone

Description automatically generated with low confidence The user can open the menu drawer from the top left corner in order to navigate the application. However, as stated before, this is certainly not the only way to navigate the application. The main way is voice-based: the user directs navigation by voice commands only. However, for visually imapired individuals who are not completely blind and who may want to explore the different modules that we offer, or for VIB individuals who may have a sighted assistant, the menu drawer offers an etremely simple yet efficient overview of the different modules. It implements a beautiful and simple UI that lets any user get a coherent idea about what the application offers. Each module is represented by an expressive icon next to its name. For example, the scene descriptor has a camera, the face recognizer a face, the emotion recognizer a heart, the clothes descriptor a shirt and pants, and so on. Clicking on any one of them redirects the user A screenshot of a phone

Description automatically generated with medium confidenceto the appropriate module.

**4.10.2.4 Scene Descriptor**

When the user accesses the scene descriptor module, either through voice command or by clicking on the menu icon, a camera interface appears. Clicking on the camera button initiates a continuous loop that captures a photo, sends it to the server for analysis, and returns information about detected objects along with their distances from the device. This information is then spoken out loud to the user.

It is important to note that the loop will not start a new iteration until the previous spoken sentence is completed, regardless of its length. This process repeats indefinitely, allowing the user to continuously receive updated information about their surroundings.

A computer on a desk

Description automatically generated with medium confidence

**4.10.2.5 Other Modules**

All the other modules in the application have the same form: a camera opens up and captures an image, which is sent to the server for analysis. Results are spoken out to the user. This includes: Face Recognizer, Emotion Recognizer, Text Reader, Clothes Descriptor, and Currency Recognizer.

**4.10.2.6 How It Works**

The How It Works Module contains a brief paragraph that explains the basic functioning of the application and which provides enough directions for any user to begin using the application. Since our target audience is VIB individuals, this message is not only displayed on the screen but also spoken out loud.

# System Testing and Verification

In this chapter, you have to explain all the steps you carried out to ensure that project outcomes are realized correctly. Your testing setup, strategy and environment should therefore be described. Your efforts for unit testing as well as integrated system testing should be given. Finally, the results from different testing scenarios should be highlighted and discussed.

In this space, before the first section, write an introductory paragraph on how you test and verify the correct operation of your system

## Testing Setup

Explain the setup you are using in testing your project

## Testing Plan and Strategy

Explain the methodology you follow while testing your project in details

### Module Testing

Explain the steps you carried out to test different modules within the project. Give and discuss the results obtained from the testing of these modules

### Integration Testing

Explain the steps you carried out to test the integrated system of your project. Give and discuss the results obtained from this whole project testing

## Testing Schedule

Mention your testing schedule

## Comparative Results to Previous Work

Give a summary of comparative results to previous work in Tabulated and or Graphical form along with a short commentary.

# Conclusions and Future Work

This chapter should summarize the whole project, it features and limitation. Moreover, you should give directions for future work

In this space, before the first section, write an introductory paragraph for the chapter

## Faced Challenges

Mention all the problems/challenges that you faced while working with the project and how you overcome them

## Gained Experience

Mentioned the experience/skills that you gained from working with the project

## Conclusions

Write your conclusions regarding the project. Mention its features and limitations

## Future Work

Give possible extensions, enhancements and future work of you project, such that subsequent students could build on your work and develop larger systems/platforms.

# References

The references should be ordered according to their appearance in the text. Ensure that all references are cited throughout your report text. The following are examples of how to write different types of references “[1] Book, [2] Journal/magazine articles, [3] conference paper, [4] website, [5] thesis”. Replace the fields with those of your used references. Question marks “??” should be replaced by the corresponding number

1. Author1, Author 2,…, “Book title,” name of publishing firm, edition, year
2. Author1, Author2,…., “Title of journal article,” name of the journal, vol. ??, no. ??, pp. ??, year of publication
3. Author1, Author2,…, “Title of conference paper,” in proceedings of conference name, city, country, date, year, pp. ??
4. Author or Corporation name, “Title,” year, link for the website, last accessed: date of last access
5. Author, “Thesis title,” M.Sc./Ph.D. thesis, Department, University, year

**Appendix A: Development Platforms**

**and Tools**

This appendix explains used tools, platforms, and hardware kits. Any ready-made module should be mentioned and discussed in this appendix. The appendix is divided into two main sections; one for the hardware and the other is for software. Within each section, you could add as much subsections as needed, according to the number of tools and platforms that you use in your project.

In this space, before the first section, write an introductory paragraph to the appendix

**A.1. Hardware Platforms**

A description of any used hardware platforms/kit should be written in this section. Each platform/kit is better described in a separate subsection. (A1.1..)

**A.2. Software Tools**

A description of any used software tool/package should be written in this section. Each tool/package is better described in a separate subsection (A2.1,..)

**Appendix B: Use Cases**

Include all your use cases

**Appendix C: User Guide**

Prepare a user guide for your project. Ensure that the guide is clear, detailed and easy for an ordinary customer to use your project. Employ figures and charts as needed to facilitate the use of your guide

**Appendix D: Code Documentation**

Your code or parts of the code you feel necessary could be included here (optional) however for one copy of this report an attached CD with all of the code is a must.

Remember you will deliver three copies of this report.

**Appendix D: Feasibility Study**

Give a detailed feasibility study of your project