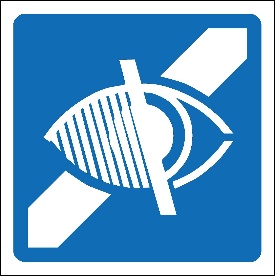


Cairo University

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

**Supervised by**

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](#_Toc136834819)

[الملخص iv](#_Toc136834820)

[ACKNOWLEDGMENT v](#_Toc136834821)

[Table Of Contents vi](#_Toc136834822)

[List of Figures viii](#_Toc136834823)

[List of Tables ix](#_Toc136834824)

[List of Abbreviation x](#_Toc136834825)

[List of Symbols xi](#_Toc136834826)

[Contacts xiii](#_Toc136834827)

[Chapter 1: Introduction 1](#_Toc136834828)

[1.1 Motivation and Justification 1](#_Toc136834829)

[1.2 The Essential Question 1](#_Toc136834830)

[1.3 Project Objectives and Problem Definition 2](#_Toc136834831)

[1.4 Project Outcomes 2](#_Toc136834832)

[1.5 Document Organization 2](#_Toc136834833)

[Chapter 2: Market Visibility Study 3](#_Toc136834834)

[2.1 Targeted Customers 3](#_Toc136834835)

[2.2 Market Survey 3](#_Toc136834836)

[2.2.1 Competitive Project 1 3](#_Toc136834837)

[2.2.2 Competitive Project 2 3](#_Toc136834838)

[2.3 Business Case and Financial Analysis 3](#_Toc136834839)

[Chapter 3: Literature Survey 5](#_Toc136834840)

[3.1 Background on Topic 1 5](#_Toc136834841)

[3.2 Background on Topic 2 5](#_Toc136834842)

[3.3 Comparative Study of Previous Work 5](#_Toc136834843)

[3.4 Implemented Approach 5](#_Toc136834844)

[Chapter 4: System Design and Architecture 7](#_Toc136834845)

[4.1 Overview and Assumptions 7](#_Toc136834846)

[4.2 System Architecture 7](#_Toc136834847)

[4.2.1 Block Diagram 7](#_Toc136834848)

[4.3 Module 1 8](#_Toc136834849)

[4.3.1 Functional Description 8](#_Toc136834850)

[4.3.2 Modular Decomposition 9](#_Toc136834851)

[4.3.3 Design Constraints 19](#_Toc136834852)

[4.3.4 Other Description of Module 1 20](#_Toc136834853)

[4.4 Module 2 20](#_Toc136834854)

[4.5 Face Detection 23](#_Toc136834855)

[4.5.1 Functional Description 23](#_Toc136834856)

[4.5.2 Modular Decomposition 24](#_Toc136834857)

[4.5.2.1 Feature Extraction 24](#_Toc136834858)

[4.5.2.2 Feature Selection and Classifier Construction 24](#_Toc136834859)

[4.5.2.3 Detection 25](#_Toc136834860)

[4.5.3 Design Constraints 26](#_Toc136834861)

[4.6 Emotion Detection 26](#_Toc136834862)

[4.7 Retial Product Identifier 29](#_Toc136834863)

[4.7.1 Functional Description 29](#_Toc136834864)

[4.7.2 Modular Decomposition 30](#_Toc136834865)

[4.7.3 Design Constraints 30](#_Toc136834866)

[4.8 Apparel Recommender 30](#_Toc136834867)

[Chapter 5: System Testing and Verification 31](#_Toc136834868)

[5.1 Testing Setup 31](#_Toc136834869)

[5.2 Testing Plan and Strategy 31](#_Toc136834870)

[5.2.1 Module Testing 31](#_Toc136834871)

[5.2.2 Integration Testing 31](#_Toc136834872)

[5.3 Testing Schedule 31](#_Toc136834873)

[5.4 Comparative Results to Previous Work 31](#_Toc136834874)

[Chapter 6: Conclusions and Future Work 32](#_Toc136834875)

[6.1 Faced Challenges 32](#_Toc136834876)

[6.2 Gained Experience 32](#_Toc136834877)

[6.3 Conclusions 32](#_Toc136834878)

[6.4 Future Work 32](#_Toc136834879)

[References 33](#_Toc136834880)

# List of Figures

[Figure ‎4‑1 Face Detection Flow Chart 23](file:///D:\University\Senior%202\GP\Github\Graduation-Project\Final%20Report\GP2\GP%20Book.docx#_Toc136834883)

[Figure ‎4‑2 Examples of Harr-like features 24](file:///D:\University\Senior%202\GP\Github\Graduation-Project\Final%20Report\GP2\GP%20Book.docx#_Toc136834884)

[Figure ‎4‑3 Example of cascaded classifier 25](file:///D:\University\Senior%202\GP\Github\Graduation-Project\Final%20Report\GP2\GP%20Book.docx#_Toc136834885)

[Figure ‎4‑4 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#_Toc136834886)

[Figure ‎4‑5 Emotion Detection Flow Chart 27](file:///D:\University\Senior%202\GP\Github\Graduation-Project\Final%20Report\GP2\GP%20Book.docx#_Toc136834887)

[Figure ‎4‑6 Example of HOG feature 28](file:///D:\University\Senior%202\GP\Github\Graduation-Project\Final%20Report\GP2\GP%20Book.docx#_Toc136834888)

[Figure ‎4‑7 Example of ensemble of regression trees 28](file:///D:\University\Senior%202\GP\Github\Graduation-Project\Final%20Report\GP2\GP%20Book.docx#_Toc136834889)

[Figure ‎4‑8 Flow Chart of Retail Product Identifier 29](file:///D:\University\Senior%202\GP\Github\Graduation-Project\Final%20Report\GP2\GP%20Book.docx#_Toc136834890)

# 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 |
|  |  |

# 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 system is intended for VIB people. They can greatly benefit from the system by using the mobile application to make their day-to-day operations easier.

## Market Survey

In this section, list the competitive products to your work. Similar commercial tools/platforms should be mentioned and discussed. Write a subsection for everyone of them and explain its pros and cons in that subsection

### Competitive Project 1

Explain and discuss each competitive project

### Competitive Project 2

Explain and discuss each competitive project

## Business Case and Financial Analysis

In this section you describe the success of establishing a company to sell your product (or service)

Two Aspects must be addressed

Business Case: Based on Market survey above you should anticipate how many products you will sell over the next 5 years and how will you set your price to counter the competition.

Financial Analysis: Based on the business case we must anticipate

1. The Capex (Capital Expenditure): These are one-time spending that you pay for development and buying things for the company
2. The Opex (Operational Expenses): These are recurring payments for salaries and marketing and … etc.

Then you create what we call a cash flow table (on an excel sheet). In this sheet you put down your monthly capex and opex on a set of rows and your reveneus (money you get back from selling product/services) on another set of rows.

The difference between both sums is your profit before tax.

It is likely that this difference is negative at beginning until your sales increase and counter the expenses.

From this cash flow analysis you find the date of the break even point wbich is the date at which all the money you get back equals the money you spent. From that date onward you will be making true profit ☺.

# Literature Survey

This chapter consists of two parts. In part one, give any necessary engineering and non-engineering backgrounds that you see important for the complete understanding of your project. These backgrounds include, but are not limited to, facts, theory, formulas, algorithms and techniques. In other words, any pivotal knowledge to your project should be given, discussed, and properly defined. In part two give a short literature review of the latest publications related to your project within past three years if applicable. Specially in this chapter, avoid lengthy unrelated discussion. More important, copy-and-paste should never be used. You have to write everything with your style and wording.

In this space, before the first section, write an introductory paragraph to describe the topics and organization of the chapter

## Background on Topic 1

Give this section a title related to the topic you cover and then write the related information as explained above.

## Background on Topic 2

Similar to the previous section, give this section a title related to the topic you cover and then write the related information as explained above.

Just choose the most two important topics however you make sure to cover all necessary facts, theory, formulas, algorithms and techniques.

## Comparative Study of Previous Work

In this section give a comparative, classified short literature review of the latest publications the latest publications related to your project within past three years if applicable

## 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

## Module 1

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.

4.3.1 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.

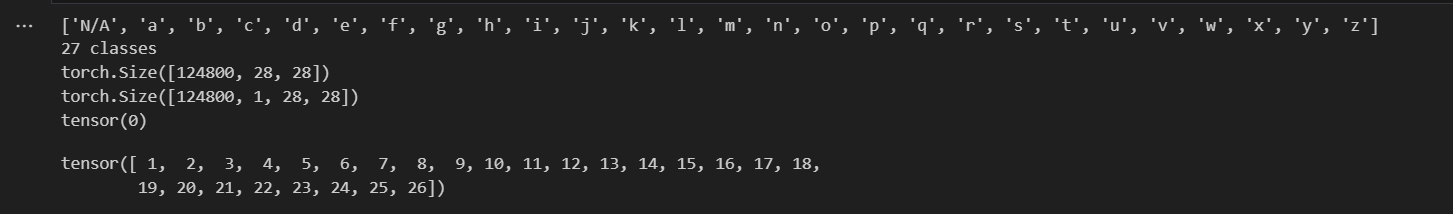


Figure ‎4‑1 the N/A that is found in the emnist dataset making them 26 classes

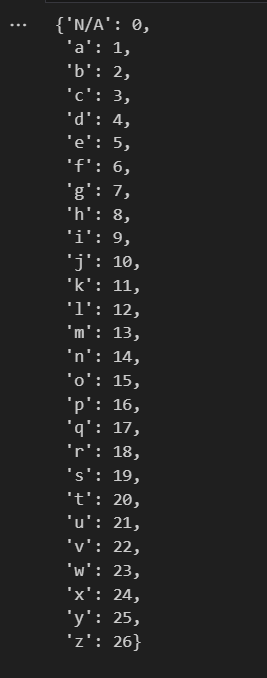


Figure ‎4‑2 Each index represents one of the alphabets

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.

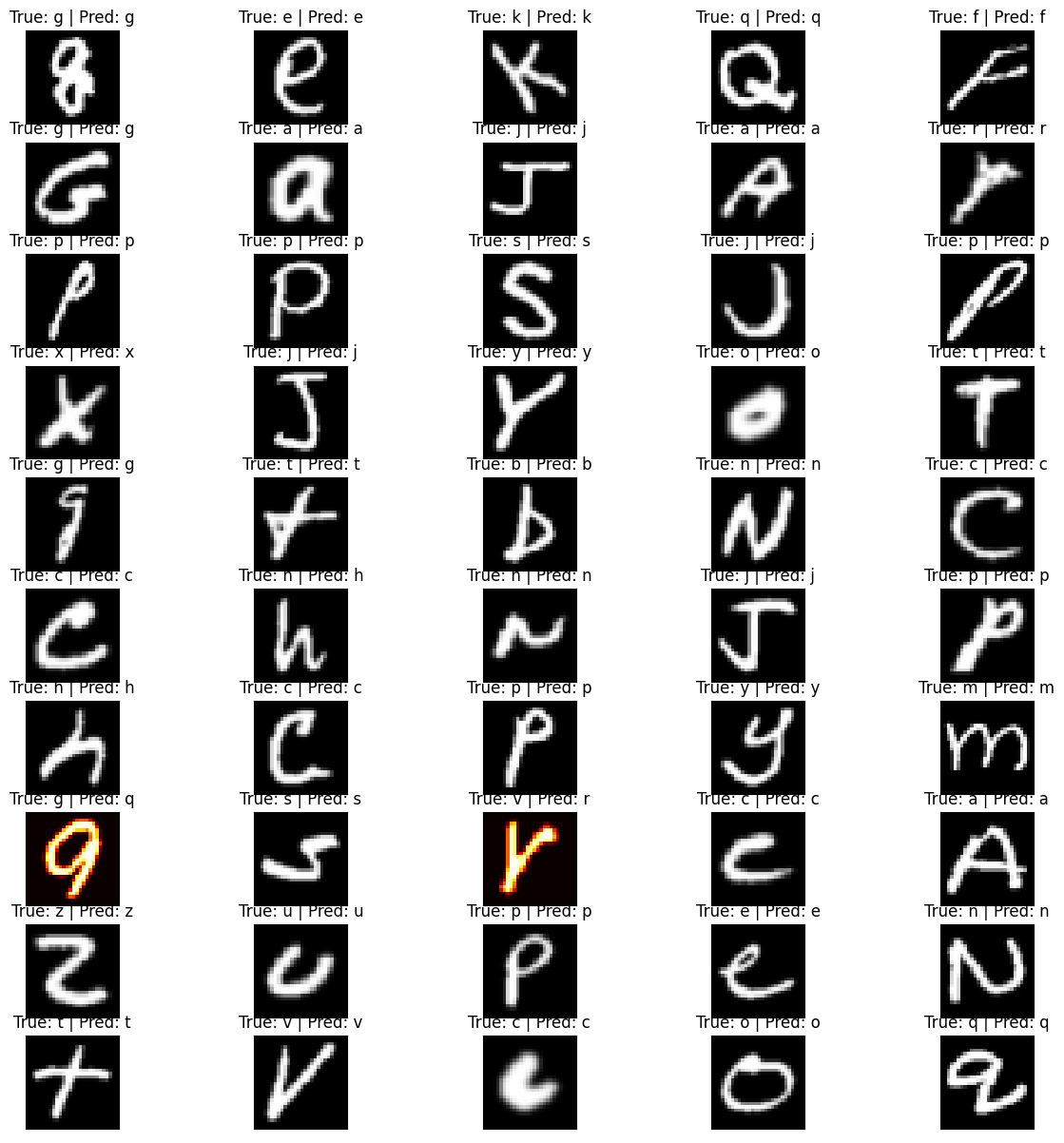


Figure ‎4‑3 The result of the trained module it can recognize most of the letters but mistake few that are similar like in this figure we have v mistaken for r and g mistaken for q hence the accuracy of this model is 93.2%

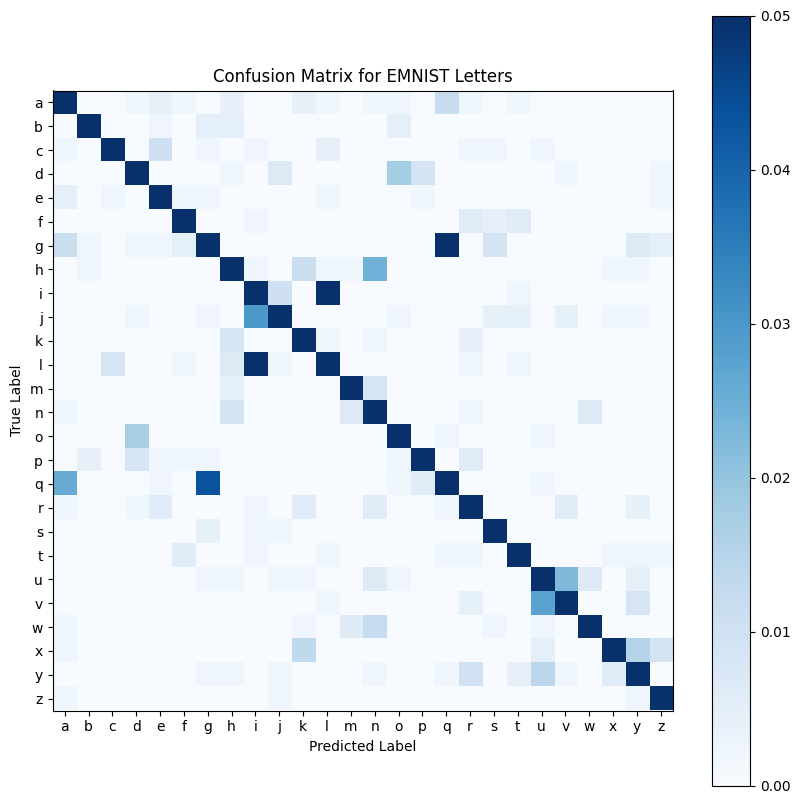


Figure ‎4‑4 The resulting plot is a 26x26 grid representing the performance of the character recognition model on the EMNIST Letters dataset. The rows represent the true labels, and the columns represent the predicted labels.

### **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.

Here's a breakdown of the code segment:

python

def extract\_text\_from\_document\_image(image\_path, model):

preprocessed\_image = preprocess\_document\_image(image\_path)

lines = segment\_lines(preprocessed\_image)

text = ''

temp = None

for line in lines:

characters = segment\_characters(line)

for char\_img in characters:

char = recognize\_character(char\_img, model)

if temp is None:

temp = char

text += char

text += '\n'

return text,temp

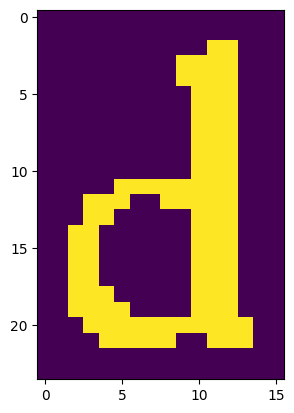


Figure ‎4‑5 When applying the below function we get the letter separated and padded so it can be easily detected by our model and recognized as the letter D

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. Reading the Dataset: The model starts by reading the dataset containing images of various currency denominations. The dataset should include a diverse range of images with different lighting conditions, orientations, and scales to ensure accurate recognition under real-world conditions.

2. 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.

3. 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.

4. 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.

5. Output: The model provides the identified currency denomination as output, which can be used to inform the visually impaired user about the currency in hand. This output can be delivered through an accessible interface, such as an audio prompt or a tactile feedback device.

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**

**4.4.3. Design Constraints**

Explain the constraints that affect the design of the module

**4.4.4. Other Description of Module 2**

Give any other necessary discussion of the module to ensure that it is clearly described.

## 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:

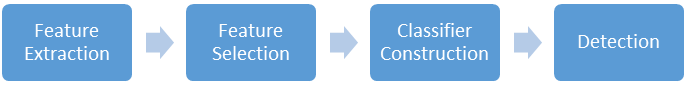
1. Feature extraction: The algorithm uses Haar-like features, which are simple rectangular patterns that capture the contrast between adjacent regions in an image. These features are computed efficiently using an integral image representation.

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 ‎4‑6 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

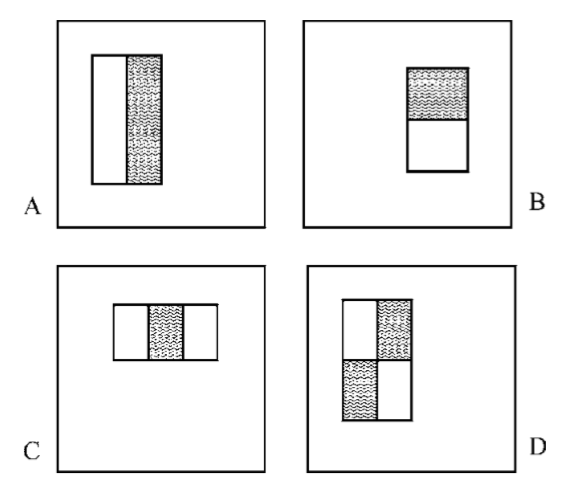
The first step that needs to be done is creating the Haar-like feature that the algorithm will select from for face detection. Haar-like features are rectangular patterns that capture the contrast between adjacent regions in an image. The Haar-like features used are: two-rectangle, three-rectangle, and 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 ‎4‑7 Examples of Harr-like features

#### Feature Selection and Classifier Construction

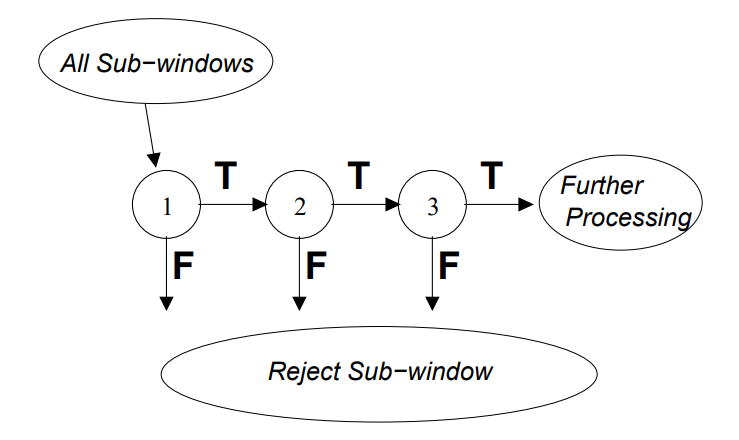
Since each one of those features is a weak classifier, Adaboost is used to select the most relevant Haar-like features and combine them into a strong classifier that can accurately detect faces in the image. The final output of the Adaboost is a list of weighted weak classifiers that can act as one strong 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 ‎4‑8 Example of cascaded classifier

#### Detection

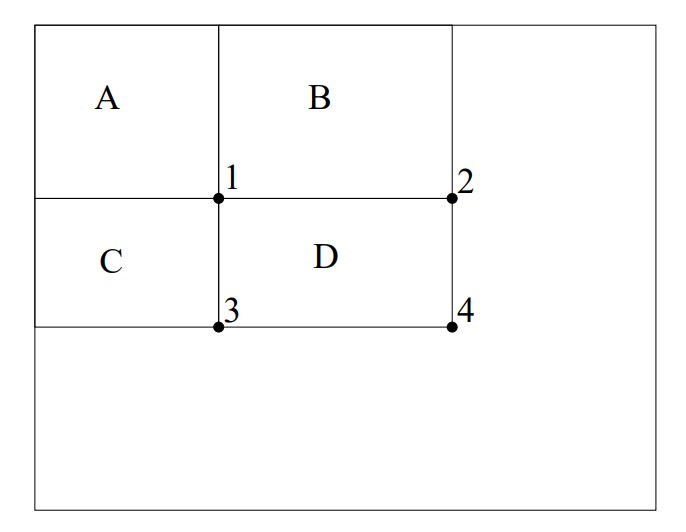
After creating the cascaded classifier, the module is ready for face detection. The input image is first preprocessed. The preprocessing includes normalization and calculating the integral image. An integral image is a data structure that allows for fast and efficient computation of sum of pixel values 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 ‎4‑9 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.

**4.4.1. 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 ‎4‑10 Emotion Detection Flow Chart

**4.4.2. 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.

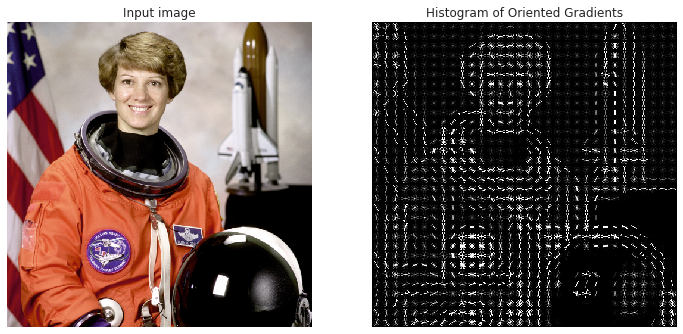


Figure ‎4‑11 Example of HOG feature

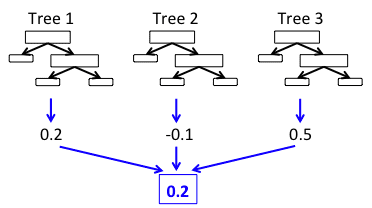
 As for the facial landmark extractor, the model uses a machine learning model that has been trained on a large dataset of annotated images to learn how to locate the landmarks. The model consists of an ensemble of regression trees that split the image into smaller regions and predict the offset of each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

Figure ‎4‑12 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.

**4.4.3. 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 ‎4‑13 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

# 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