FINAL REPORT

Development of a Mobile Application For Image Processing to Improve Environmental Recognition and Navigation for Blind People

NAME: MOHAMED AZHER ABDULLAH AAZAM

UWL ID: 33118842

MODULE CODE: CP6SL460

PROJECT LINK: Click Here

Declaration

I declare that this is my original work, and that no material previously submitted for a degree or diploma at any other university or school of higher learning is included into my dissertation without acknowledgement. To the best of my knowledge and belief, it contains no previously published or written content by another individual, with the exception of an acknowledgement in the text.

Furthermore, I hereby give the University of West London and American National College the nonexclusive right to reproduce and distribute my dissertation, in whole or in part, in print, electronic, or any other form. I reserve the right to use this content in full or in part in future works.

Mohamed Azher Abdullah Aazam

Acknowledgement

I hereby acknowledge that the work presented herein is entirely my own creation. I extend my sincere gratitude to the institute for providing me with the necessary resources and support throughout the duration of this project. Additionally, I would like to express my heartfelt appreciation to my supervisor, Mr. Mahen Jayalath, for his invaluable guidance, mentorship, and unwavering support during the course of this endeavor. His expertise and encouragement have been instrumental in shaping the outcome of this work, and I am truly grateful for his contributions.

Furthermore, I must stress that, while I obtained the models for this project from Kaggle, all other components of this work, including implementation, analysis, and interpretation, are completely my own.

Abstract

Echo Nav is a groundbreaking mobile accessibility software that utilizes machine learning and image processing to empower visually impaired individuals. This program enables individuals with visual impairments to increase their independence and quality of life.

The system's unique feature is real-time environmental recognition and navigation, enabling users to securely explore their surroundings. This feature uses advanced algorithms like YOLOv5 to accurately recognize and classify objects and barriers.

Echo Nav excels in text-to-speech conversion, empowering users to shop independently by recognizing and providing product information through voice requests.

Although there are worries about data collection and privacy, echo navigation has the potential to help visually impaired individuals. Future developments might enhance bill reading, color identification, object detection, and overall user experience for more inclusion and accessibility.

Keywords: Vision Mate, visually challenged, accessibility app, machine learning, image processing, YOLOv5, environmental detection, text-to-voice conversion, independent shopping, CNN model, clothing color recognition.

Project Link

The proposed system was successfully created and published to Google Drive for easy access and sharing. The project files, which include key documentation, codebases, and resources, have been methodically organized on the drive. Furthermore, the Google Drive connections have been linked into a webpage hosted in my repository. This strategic choice guarantees a planned and coordinated approach to project deliverables, allowing for easy cooperation and access by Supervisor and Interested parties. You may discover the webpage related repository link below.

Link: https://azamazher.github.io/echonav/

Table Of Contents

Table of Tables

Table of Figures

List of abbreviations

Abbreviation	Description
YOLO	You Only Look Once
TTS	Text-To-Speech
STT	Speech-To-Text
NLP	Natural Language Processing
GPU	Graphics Processing Unit
MAP	Mean Average Precision

Table 1 list of abbreviations

Introduction

Individuals with vision impairment encounter significant challenges, affecting their independence and overall quality of life. Developing a mobile accessibility software with machine learning and image processing can empower blind individuals and address these challenges. This research highlights the importance of an application in improving accessibility and autonomy for visually impaired individuals.

Visual impairment is a common condition affecting millions of individuals. The World Health Organization's "World Report on Vision" emphasizes the need of creating innovative approaches for treating visual impairments and improving accessibility for those with visual disabilities (World Heath Organization, 2019). This research sheds light on the problem and highlights the need for assistive technology to improve the lives of visually impaired individuals.

Creating an accessible mobile app can help visually impaired individuals. Badave, Jagtap, Kaovasia, and colleagues tested an Android-based object identification system designed for visually challenged individuals (Ajinkya Badave, Rathin Jagtap, Rizina Kaovasia, Shivani Rahatwad, Saroja Kulkarni, 2020). The application promises to empower blind individuals through technology. The program uses machine learning and image processing to assess and deliver auditory feedback on visual content identified by the smartphone's camera. The smart mobile accessibility application relies heavily on machine learning to function properly.

The application uses complex algorithms and large datasets to recognize and categorize items, language, and environment in real-time. This capability enables blind individuals to access audio descriptions of their surroundings, including information regarding Objects, people, and texts. Image processing enhances the program's functionality by altering and analyzing digital photos to extract useful information. The software can convert smartphone photos to audio. This strategy helps blind people understand their surroundings by distinguishing objects, texts, and patterns.

It also provides auditory advice for navigation, allowing users to easily explore new indoor places. The benefits of the smart mobile accessibility app are numerous. It enables blind people to accomplish previously impossible activities, such as explore new areas and recognize objects. By delivering real-time audio feedback and fostering independence, self-reliance, and social involvement, the software improves their overall well-being.

The proposed smartphone app's test results demonstrate its effectiveness in real-world scenarios.

Research indicates that object detection, language interpretation, and navigation assistance may be performed with high accuracy (Ajinkya Badave, Rathin Jagtap, Rizina Kaovasia, Shivani Rahatwad, Saroja Kulkarni, 2020). User feedback suggests that visually impaired individuals exhibit increased confidence and activity.

Literature Survey

Visual impairment and blindness affect around 253 million individuals globally, with 36 million being blind (Alexy Bhowmick & Shyamanta M. Hazarika , 2017). These illnesses have an impact on independence. Mobility and overall quality of life. Mobile accessibility applications that offer real-time information, object identification, text-to-speech translation, and navigation aids are crucial for addressing these challenges. These apps can empower blind individuals and improve their quality of life.

Bhowmick and Hazarika's study examines existing and future innovations in assistive technology for the visually impaired and blind (Alexy Bhowmick & Shyamanta M. Hazarika , 2017). The 2017 research published in the Journal of Multimodal User Interfaces sheds light on the advancements in technology for visually impaired individuals. This article discusses current advances in assistive devices and gadgets, as well as future trends. The study highlights the need of using technology to increase accessibility and independence for this demographic.

The World Health Organization's "World Report on Vision" provides thorough information on global eye health (World Heath Organization, 2019). The document, published n.d., offers a comprehensive overview of vision-related issues, including prevention, treatment, and access to eye care. This highlights the significance of eye health in public health and emphasizes the need for improved vision care services and policy worldwide.

The National Programmed for Control of Blindness & Visual Impairment (NPCBVI) of the Government of India undertook the "National Blindness & Visual Impairment Survey India 2015-2019" to examine visual impairment and blindness in India over a certain time span (npcbvi.gov.in, 2020). The study provides useful information about the prevalence, etiology, and distribution of blindness in the US. This resource is critical for Indian policymakers and medical experts seeking to establish better eye health treatments and approaches.

Badave, Jagtap, Kaovasia, and colleagues propose an Android-based object detection system for visually challenged individuals (Ajinkya Badave, Rathin Jagtap, Rizina Kaovasia, Shivani Rahatwad, Saroja Kulkarni, 2020). This invention, shown at the 2020 International Conference on Industry 4.0 Technology, aims to enhance the safety and independence of visually impaired individuals. It uses item detection algorithms to help people navigate their surroundings, potentially improving their quality of life.

Matusiak, Skulimowski, and Strurnillo presented two research papers at the 2013 6th International Meeting on Human System Interaction, both on object recognition in mobile phone apps for blind or visually impaired users (P. Strurniłło, K. Matusiak, P. Skulimowski, 2013) (Devi RA, Uthaman I, Shanthanam RR, 2020). These studies explore how technology might help visually impaired individuals perceive and engage with their surroundings using mobile devices. The study explores strategies to increase accessibility and independence for those with visual impairments, offering valuable insights on assistive technologies.

In a paper published in Procedia Computer Science 2020, Sait, Ravishankar, Kumar, et al. designed and developed an assistive device for visually impaired individuals . This initiative intends to help visually impaired individuals with everyday tasks using technology. This study examines the device's qualities and capabilities to enhance accessibility and freedom for this population, perhaps increasing their level of life.

Kuriakose, Shrestha, and Sandnes presented an article in the IETE Tech Review on technologies for assisting blind and visually impaired individuals with navigation (Bineeth Kuriakose, Raju Shrestha and Frode Eika Sandnes, 2022). The research provides an overview of technological solutions to increase mobility and independence for persons with visual impairments. It also discusses the present status of the field and potential improvements in assisted navigation technology.

Manjari, Verma, and Singal's 2020 research in "Internet of Things" offers a comprehensive overview of assistive technology for visually impaired individuals (Kanak Manjari, Madhushi Verma and Gaurav Singal, 2020). The research focuses on improving the lives and accessibility of persons with visual impairments using various technological solutions. This study examines the present state of assistive technologies for vision impairment.

At the 2019 IEEE International Meeting on Consumer Electronics, Chen et al. presented a work on developing an intelligent assistance system for those who are visually impaired and blind individuals (Liang-Bi Chen, Jian-Ping Su, Ming-Che Chen, Wan-Jung Chang, Ching-Hsiang Yang, and Cheng-You Sie, 2019). The study examines how technology may enhance independence and accessibility for this group, as well as possible ways for enhancing their standard of life.

A paper presented at the 2019 International Conference on Computer Vision Workshop by Lin, Wang, Yi, and Lian suggests a deep learning-based wearable assistive device for visually impaired individuals (Yimin Lin, Kai Wang, Wanxin Yi, Shiguo Lian, 2019). Deep learning technology can help visually impaired persons improve their mobility and access to information, leading to a higher quality of life.

Uthaman, Devi and Shanthanam's 2020 research examines an IoT security-based electronic support system for visually impaired individuals (Devi RA, Uthaman I, Shanthanam RR, 2020). This technology combines detection and navigation aid to help visually impaired individuals safely navigate their environment. This research examines how IoT technology might enhance accessibility and safety for this population.

Intelligent accessible mobile software transforms the lives of blind people by going beyond basic image recognition. This innovative application identifies things and reads words, documents, and digital displays, enabling independent access to essential information. It also offers real-time audio feedback to help customers navigate new areas with ease. This innovative device promotes independence, self-reliance, and social inclusion, significantly improving the quality of life for individuals with vision impairments. This technology enhances accessibility and promotes full participation in sighted environments.

Research Gap

Features	Proposed System	Existing Systems		
		Research 1	Research 2	Research 3
Image Processing Features	☑	☑	☑	☑
System for Blind People	☑	☑	☑	☑
Identify Voice Commands	☑	☑	✕	✕
Produce an output as a Voice for assistance	☑	☑	✕	☑
Mobile Application	☑	☑	✕	✕

Table 2 Research Gap

The above table presents a comparison of various current and proposed solutions that are intended to meet the needs of people with visual impairments using technology. Let's review the features and systems described above:

Image Processing

This technology is employed in all planned and operational systems (Study 1, 2, and 3). This characteristic enables systems to interpret pictures and extract relevant info from images. It underpins the capabilities of numerous assistive devices.

Designed for the Blind

All solutions have been designed to meet the unique needs of blind individuals. Prioritizing accessibility ensures that technology addresses the unique needs of those with visual impairments. It includes features like as speech outputs and touchscreen interfaces.

Identify Voice Commands

The ability to recognize voice commands was included in the proposed research and study 1. This feature enables users to interact with the system using voice commands, providing a more intuitive user experience. This feature can greatly enhance usability for individuals who struggle with typical interfaces.

Voice Assistance

The proposed system, Research 1 and 3, can produce voice assistance. This capability is very beneficial for blind people since it allows them to hear information. This approach not only enhances comprehension but also promotes independence by eliminating the need for external assistance.

Mobile Application

Proposed Research and Research 1 are accessible as smartphone apps. Smartphones and tablets are widely used and familiar devices, making them easily accessible. This platform choice enhances the mobility and usability of assistive technology, enabling users to take it with them wherever.

In summary, the proposed system and Research 1 offer several features tailored to the needs of visually impaired individuals. They employ image processing, cater to the blind, recognize voice requests, and send forth speech messages. Mobile applications provide increased accessibility and portability.

Other research methods lack picture processing capabilities, limiting their usefulness for visually impaired people. some Additionally, it lacks the ability to recognize spoken commands and create audio messages, which are crucial for providing assistance to individuals with visual impairment. These approach may not meet the unique needs for my Proposed system.

Research Problems

This research, "Supporting Blind People through the Smartphone Accessibility Application," tackles a serious issue in society: a lack of inclusive technological solutions for blind individuals. This project aims to address the disparities in access to data, independence, and standard of life for blind individuals owing to limitations in present assistive technology. This research project has subcomponents that address unique difficulties for the blind community.

Blind persons have tremendous difficulty navigating their surroundings autonomously. Mobility aids are sometimes limited, especially in outdoor environments with unanticipated impediments.

This study emphasizes the need for improved image processing technologies to enhance awareness of the environment and navigation for those who are blind. Previous research has addressed image processing approaches for these uses, but obstacles remain. Developing algorithms and systems for outdoor situations can be challenging due to their unexpected and varied nature.

Furthermore, conventional image processing technologies are both costly and difficult to use. This restricts them access to a larger community of visually challenged individuals. Developing inexpensive and user-friendly technology that integrate seamlessly into daily life is crucial for benefiting this group.

Finally, the ethical and privacy implications of these developments must be addressed. Complexity in image processing systems increases the risk of unexpected outcomes, including abuse of personal data. Addressing ethical and privacy concerns is crucial for the appropriate development and application of these technologies. This research strives to improve image processing systems by making them more effective, inexpensive, and accessible, while also considering ethics and privacy. More improvement of algorithms for outdoor navigating is required.

Blind individuals may struggle to discover and recognize products when shopping independently. This project aims to create a mobile application that improves market access for visually challenged individuals, particularly those who are blind. The program intends to improve accessibility in retail, allowing blind individuals to shop independently and confidently.

Blind buyers sometimes struggle to identify and locate items on store shelves. To overcome this, the suggested application utilizes YOLOv5, an enhanced Using smartphone cameras, object recognition technology allows for real-time product identification. The app uses text-to-speech (TTS) technology to describe things, making shopping easier by merely pointing the phone to the shelf. TensorFlow provides pre-trained Yolov5 and other modules for the Suggested application. It is free to use, and many models were provided and tested ready to use, Which is built by multiple programmers.

The software has speech-to-text (STT) functionality, enabling users to ask questions about product specifications or store locations. The technology transforms spoken inquiries into text and extracts pertinent information from TensorFlow's pre-trained models. This two-way engagement promotes an inclusive and enjoyable purchasing experience.

The combination of speech recognition, picture processing, TTS, and STT poses substantial technological and usability issues. Thorough research and refining are necessary to provide accurate and fast object identification, natural speech synthesis, and effective STT for comprehending user questions.

Additionally, ethical concerns like as security of data, confidentiality, and user permission must be addressed. Achieving a balance between gathering critical audio and visual data and maintaining user privacy is a top priority.

This project aims to create a YOLOv5-based object identification application with TTS and STT, object detection model is available for download on Kaggle. For more information about pre-trained models, see TensorFlow’s official website. These models contribute to the improvement of market access for blind persons using mobile applications. Addressing technological, practical, and ethical concerns is crucial for producing a successful tool that promotes independence and inclusion of blind individuals in retail settings.

In brief, the research aims to empower blind individuals by solving accessibility gaps in their daily lives. The research aims to enhance the standard of life, independence, and involvement of blind individuals in society by creating a mobile accessible software that covers certain subtopics. This work is a crucial step towards a more equitable and accessible world for all.

Research Objectives

Main Objective

This project aims to enhance accessibility, independence, and quality of life for visually impaired individuals through the development of innovative mobile applications. These applications will leverage modern technologies like object detection and text-to-speech synthesis to help users with common activities.

Specific Objectives

Environmental Recognition and Navigation for Supermarkets:

Suggested mobile app uses advanced object recognition techniques, including the pre-trained YOLOv5 model from Kaggle, to help visually impaired individuals navigate supermarkets and recognize objects in real-time. This promotes greater accessibility and autonomy during shopping.

Include an audio Feedback:

The app should provide aural input to help users understand the surroundings and things in the image. TTS technology can convert written descriptions into audio.

Enhance Depth Perception and Barrier Recognition:
The third objective aims to improve the system's depth perception and obstacle identification accuracy. Optimizing image processing techniques ensures reliable distance measurements for various objects and surfaces. Algorithms should distinguish between different sorts of obstacles to help users recognize possible risks in their path.

Methodology

System Diagram

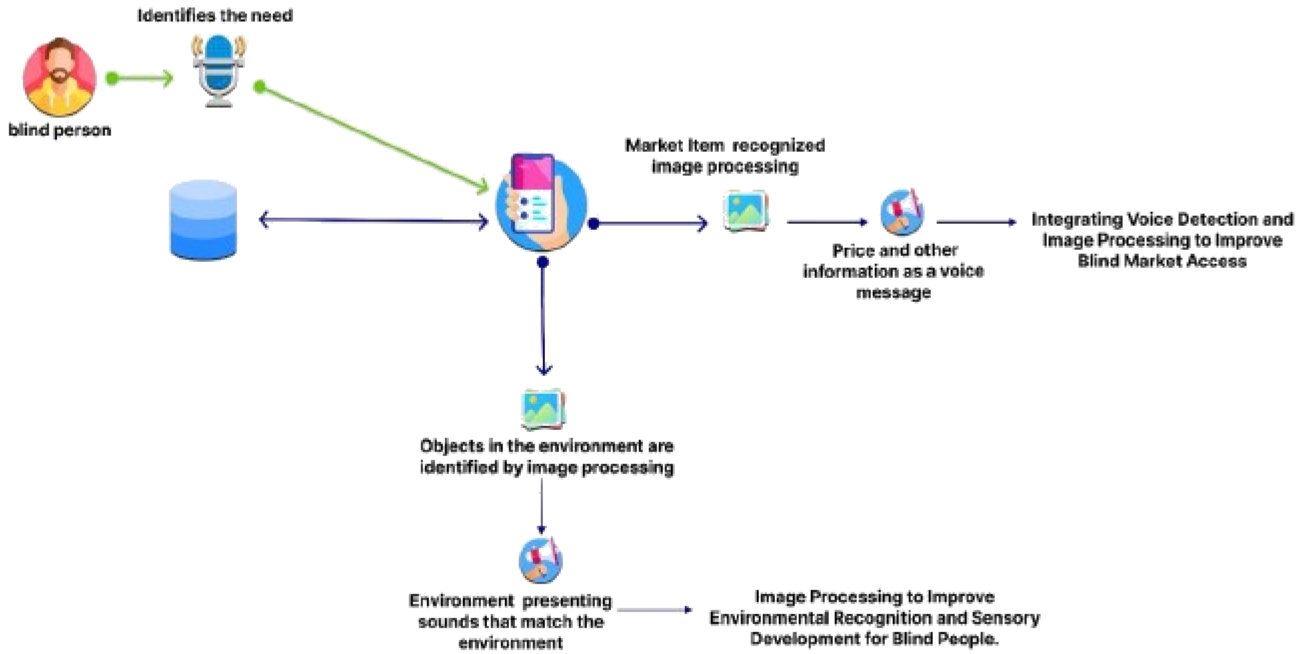

Figure 1 Proposed System Diagram

This article describes a mobile application built with Flutter. The mobile app works with both Android and iOS systems. The mobile accessibility software for blind users uses machine learning and image processing to improve accessibility and independence. The system consists of critical components that work together to analyze visual material collected by the smartphone camera and provide real-time auditory feedback to users.

Identifying Data collection and data processing on Pre-trained Models

Developing a system for supermarket object identification and interior navigation for visually impaired persons requires a wide and representative dataset. This dataset should appropriately represent the many situations that blind individuals traverse, with a focus on supermarket settings. The scope should include indoor and outdoor environments, with a focus on supermarkets.

The camera pictures used for data collecting must be carefully selected. Using fixed cameras and smartphone cameras indoors is a cost-effective approach. These gadgets gather accurate visual data of indoor locations, offering useful information for improved navigation in supermarkets.

Creating unique algorithms for supermarket object detection and interior navigation aids necessitates methodical analysis of a big dataset and the selection of relevant camera images. This dataset is used to train machine learning models and gadgets that help visually impaired people be more independent and secure in supermarkets. Since all models were obtained from Kaggle, model training will not be required for our mobile application.

But not all model can be usable. Because the model which is required for our mibile application should contain Various approaches, such as contrast enhancement and sharpening, are used to improve object visibility and clarity. These innovations improve object visibility, especially in low-light or challenging environments. Data augmentation procedures are crucial for diversifying the dataset. These approaches simulate various perspectives, lighting situations, and backdrops, resulting in unique item appearances. Augmented data helps the model generalize to a wider range of real-world grocery settings. The model that will be utilized for the application should address all of these needs. so Several pre-trained models should be examined, with the best accurate model being utilized for the application.

The model should standardize the format, size, and color representation of object pictures through normalization. Consistency in these qualities enables the machine learning model to analyze data reliably, regardless of source or capturing device. An Accurate object annotation is a critical step in this process. This requires accurately categorizing objects of interest in the dataset. Annotations are used as ground truth labels during model testing, allowing the model to reliably detect grocery goods and their properties.

Yolov5 Model From Kaggle

Using pre-trained Kaggle Models built using Yolov5 for real-time picture recognition does not require us to incorporate our own dataset because such pre-trained models come with their own training datasets by the produced programmer.

If we aim to develop a yolov5 model from scratch, the following process should be taken:

Using your own dataset to adapt YOLOv5 for real-time picture recognition requires many essential processes. Collect a large number of annotated photographs of the things or classes you'd like the model to detect. labeling or VGG Image Annotator (VIA) may help you create bounding boxes around things in pictures to identify their positioning. Save annotations in a suitable format, such as YOLO, which provides the object class and bounding box coordinates based on picture dimensions.

Configure the YOLOv5 model architecture based on your dataset's number of object classes. Typical processes include modifying the model's configuration file to specify the number of classes, adjusting anchor box sizes, and using data augmentation strategies to improve generalization. After completing your dataset and setup, you may begin training. YOLOv5 offers a simple training script with customizable hyperparameters, data pathways, and other features. During training, the model learns to recognize things using labeled examples from your custom dataset. Training time may vary based on dataset complexity and available computational resources.

After training, examine the model's performance using metrics like as mean average precision (mAP) to determine its effectiveness in identifying objects in your specific data. To achieve desired accuracy, fine-tuning and repeated training may be necessary.Custom data images can be used to customize YOLOv5 for real-time image recognition, resulting in a tailored solution for industrial automation, wildlife monitoring, and other applications requiring accurate object detection.

Building Cross-Platform Front ends with Flutter: A Comprehensive Guide

Developing a front-end in Flutter entails developing the user interface (UI) for a mobile or online application. Flutter, an open-source framework, makes this process easier by allowing you to create natively built applications for many platforms using a single codebase. To begin, install Flutter on your development workstation and choose an IDE, such as Android Studio or Visual Studio Code. Next, start a new Flutter project and design the UI by assembling widgets. Flutter offers several layout widgets, navigation tools, and state management capabilities to modify your app's design and usefulness. You will manage user interactions, get data from services, and develop tests to guarantee your app functions properly. After completing your front-end, you can launch it on several platforms and constantly enhance it based on user input and Flutter updates.

Using' speech_to_text' and 'flutter_tts' to enable voice-activated interactions in Flutter apps.

Adding speech_to_text and flutter_tts packages to a Flutter project can improve user engagement and accessibility. The next two paragraphs explain how to utilize these packages for voice command capabilities.

Enabling Voice Input with 'speech_to_text':

To allow speech input in your Flutter application, use the'speech_to_text' package.
Initially add the package as a dependency to your 'pubspec.yaml' file. Next, import it as'stt' (short for Speech To Text). This package allows you to easily collect and convert spoken speech into text within your application. You may setup it, listen for speech input, and manage the recognition results. This allows you to include voice-controlled functionality in your program, such as voice-activated commands or text dictation. To guarantee the program works properly on both Android and iOS devices, handle microphone access permissions accordingly.

```
import 'package:speech_to_text/speech_to_text.dart' as stt;
// Initialize the speech recognition instance
32
final stt.SpeechToText speech = stt.SpeechToText();
// Start listening for voice input
void startListening() async {
  await speech.listen(
    onResult: (stt.SpeechRecognitionResult result) {
      (result.recognizedWords) // Handle the recognition result
    },
  );
}
```

Text-to-Speech Output with 'flutter_tts':

To respond to voice instructions, use the 'flutter_tts' package. Start by including 'flutter_tts' as a dependency and importing it into your Dart project. This software allows you to transform text into speech and adjust pitch, pace, and loudness. You may use it to provide audio answers to users depending on their voice input. By combining'speech_to_text' for input and 'flutter_tts' for output, you may develop voice-controlled interfaces that respond to spoken instructions, making your app more interactive and usable.

```
import 'package:flutter_tts/flutter_tts.dart'; // Initialize the Text-to-Speech (TTS) instance
final FlutterTts flutterTts = FlutterTts();  // Convert text to speech and play it
void speak(String text) async {
  await flutterTts.setSpeechRate(1.0); // Adjust speech rate if needed
  await flutterTts.setVolume(1.0);// Adjust volume if needed
  await flutterTts.speak(text);
}
```

Integrating 'speech_to_text' for voice input and 'flutter_tts' for text-to-speech output creates an interactive and dynamic user experience in Flutter applications. Users may control and engage with the app using voice commands.

Recourses

Yolov5 (Pre-Trained model from Kaggle)

"YOLOv5" represents a substantial advancement in computer vision technology. This advanced object recognition model has gained popularity because to its high speed and precision.
Deep learning in YOLOv5 relies on convolutional neural networks (CNNs).

Figure 2 Yolov5 model

Architecture is used to analyze visuals swiftly and consistently. This allows for real-time object identification in both image and video streams. This technology has several uses, including driverless cars, surveillance systems, and medical imaging, as it can instantly distinguish and categorize items inside frames. The continuation of YOLO series represents continuing efforts to advance computer vision, making it a valuable resource for academics and businesses seeking accurate and effective object detection solutions. Below provided link is what I sued for object detection for my mobile application.

Link to the Downloaded Models:
https://www.kaggle.com/models?framework=tfLite&query=image-object-detection&tfhub-redirect=true

Flutter

Google's Flutter technology simplifies cross-platform app development while providing a native-like user experience. Flutter is based on the Dart programming language, noted for its quickness and simplicity, making it ideal for UI creation. Flutter's main innovation is its widget-based architecture, which treats all elements, including buttons and panels, as widgets.
These widgets may be used to create complex and visually appealing user interfaces.

Figure 3 Flutter 3

Flutter's "hot reload" feature accelerates development time. Hot reload allows developers to rapidly observe the impact of code changes, eliminating the need for time-consuming rebuilds and facilitating experimentation and fine-tuning. This capability is ideal for iterative development and collaborative work.

Flutter has a diverse and growing widget collection, both inside the framework and in the community. The vast ecosystem of pre-built widgets simplifies development and reduces the need for customized solutions. Flutter's widgets are not only basic UI components, but also platform-specific, ensuring your app looks and feels natural across several devices and OS.

Flutter's "write once, run anywhere" concept enables developers to reuse code across several platforms, including iOS, Android, web, and desktop, decreasing development time and effort. Flutter's adaptability makes it a popular choice for startups, companies, and independent developers that want to swiftly create high-quality, cross-platform apps.Flutter has a robust community that provides documentation, plugins, and packages. The lively community supports continuing development and provides developers with varied tools and solutions.

Dart

Google Dart is a contemporary, open-source programming language. Although it has applications in other sectors, it is mainly recognized as the primary language for designing apps using the Flutter framework. Dart aims to improve online and mobile application development by prioritizing speed, simplicity, and efficiency.

Dart's optional type system enhances code quality and debugging by allowing developers to add annotations without strict type checks. This adaptability strikes a compromise between statically typed and dynamically typed languages, offering advantages of both.

Figure 4 Flutter with Dart

Darts are well-known for their exceptional performance. It employs a just-in-time (JIT) compiler for development and an advanced ahead-of-time (AOT) compiler in production. AOT compilation optimizes machine code, resulting in faster execution of Dart programs throughout development and production.

Dart's asynchronous programming paradigm simplifies handling asynchronous operations like as network queries and file I/O. The async/await syntax simplifies asynchronous code development, making it important for responsive app development.

Dart's standard library includes support for common tasks like as collection management, asynchronous programming, and date/time management. The package manager, pub, makes it simple to add external libraries and dependencies to your applications.

Dart is particularly versatile, as it may be used for a multitude of reasons than Flutter. The Dart VM allows for server-side execution, making it suitable for constructing backend services and full-stack web projects. Its code-sharing capabilities between client and server make it ideal for full-stack web development.

Requirements

Functional Requirements

- The system should recognize ambient items and retail products using the YOLOv5 algorithm.
- The system should give audio feedback for output using a TTS engine.

Non Functional Requirements

- Accessibility
- Security
- Performance
- Usability
- Compatibility
- Reliability
- Availability
- Accuracy
- Efficacy
- Scalability

User Requirements

- A user-friendly UI with easy, straightforward navigation.
- Clear and succinct audio output for feedback and output data.
- Compatibility with a variety of devices and platforms improves accessibility.
- Expertise in detecting and interpreting objects and processing images.

Software System

- Operating System (Android)

Hardware Requirements

- Smart mobile devices
- Secure Internet Connection

Commercialization Aspect of the Product

Commercializing the device is crucial for improving the lives of visually impaired individuals through innovative applications. Successful commercialization ensures solutions reach more consumers and have a long-term market presence. Here are some key elements for marketing these products:

User Acquisition and Retention:

Create an advertising and outreach plan to connect with potential consumers.
Partnership with advocacy groups, disability organizations, and leaders from the visually impaired community may be necessary. To engage users, have a strong online presence using websites, social media, and forums.

Provide app information: the website may be a valuable resource for providing information about the proposed app, such as its features, advantages, and target audience. You can provide a clear and brief explanation of the software, as well as screenshots or a video demonstration.

Offer free download: The website can also serve as a venue for consumers to download the application. This is especially important for people who may be unable to locate the software via standard app store routes.

Provide support: The website may also be a valuable resource for people who require assistance with the app. You may include a FAQ section, a knowledge base, or a contact form for people to express their inquiries.

Below Snaps Shows the interface of the Website that is created for application lunch for marketing aspect and structure project outcome.

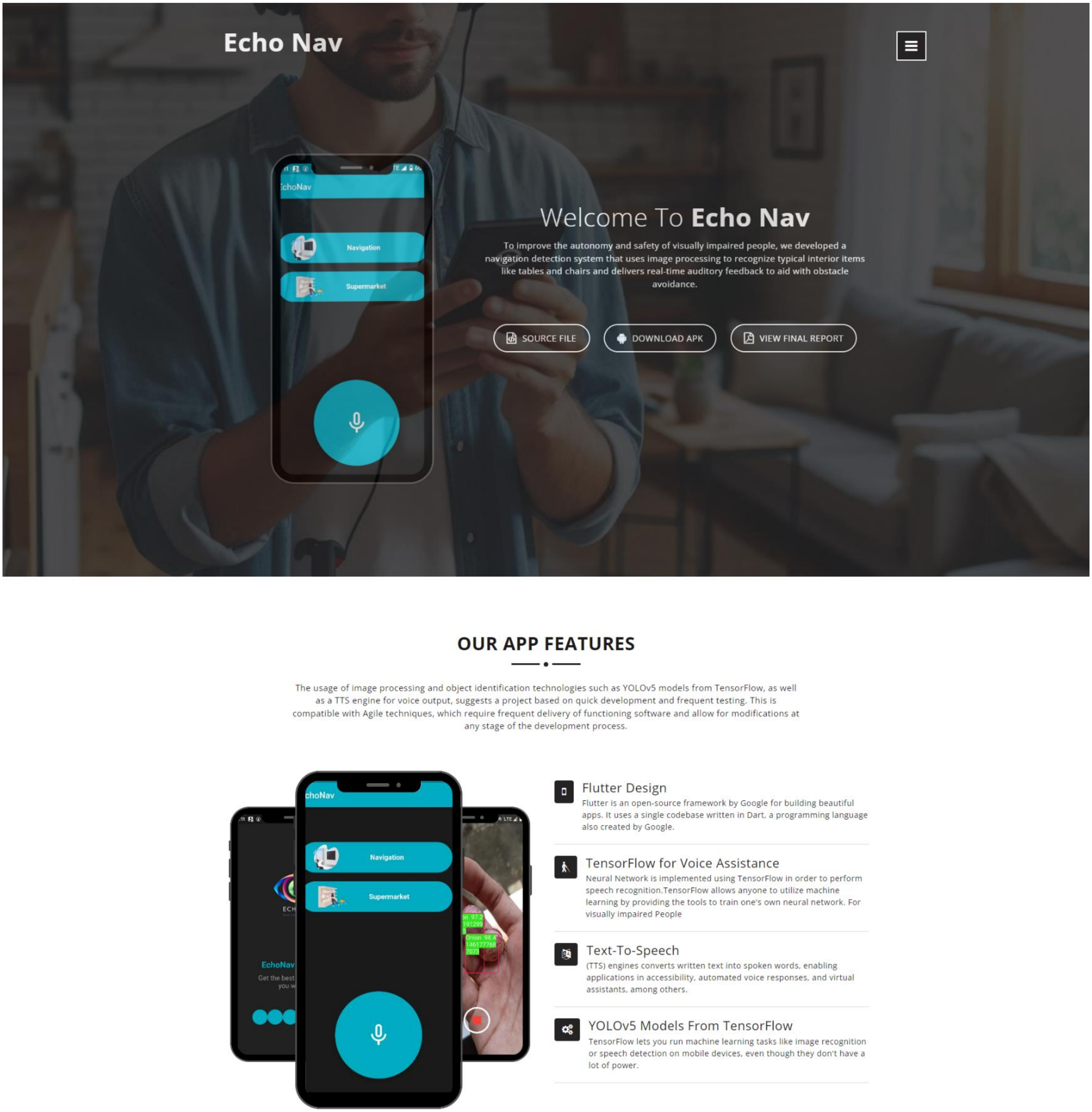

Figure 5 Application lunch website Overview - 1

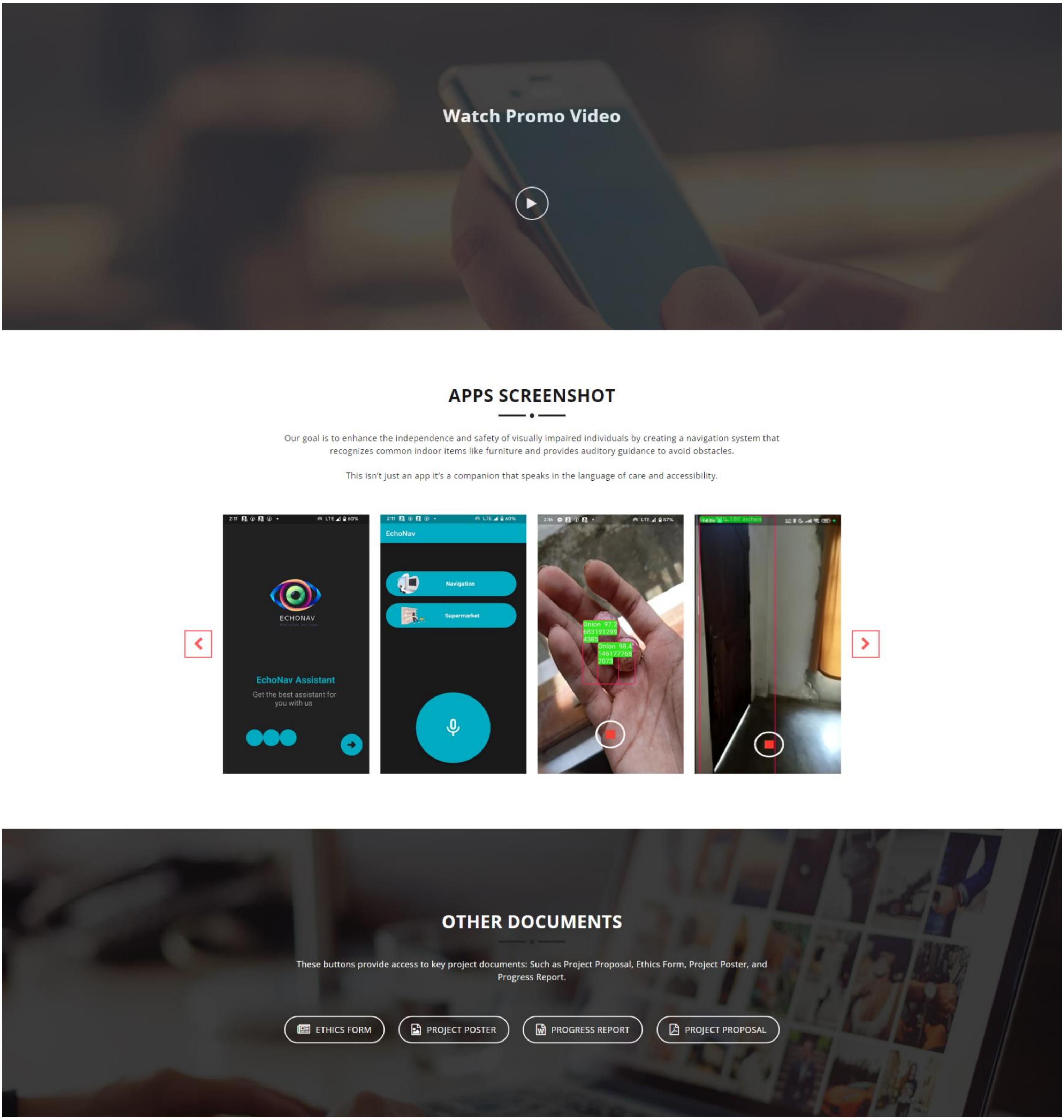

Figure 6 Application lunch website Overview - 2

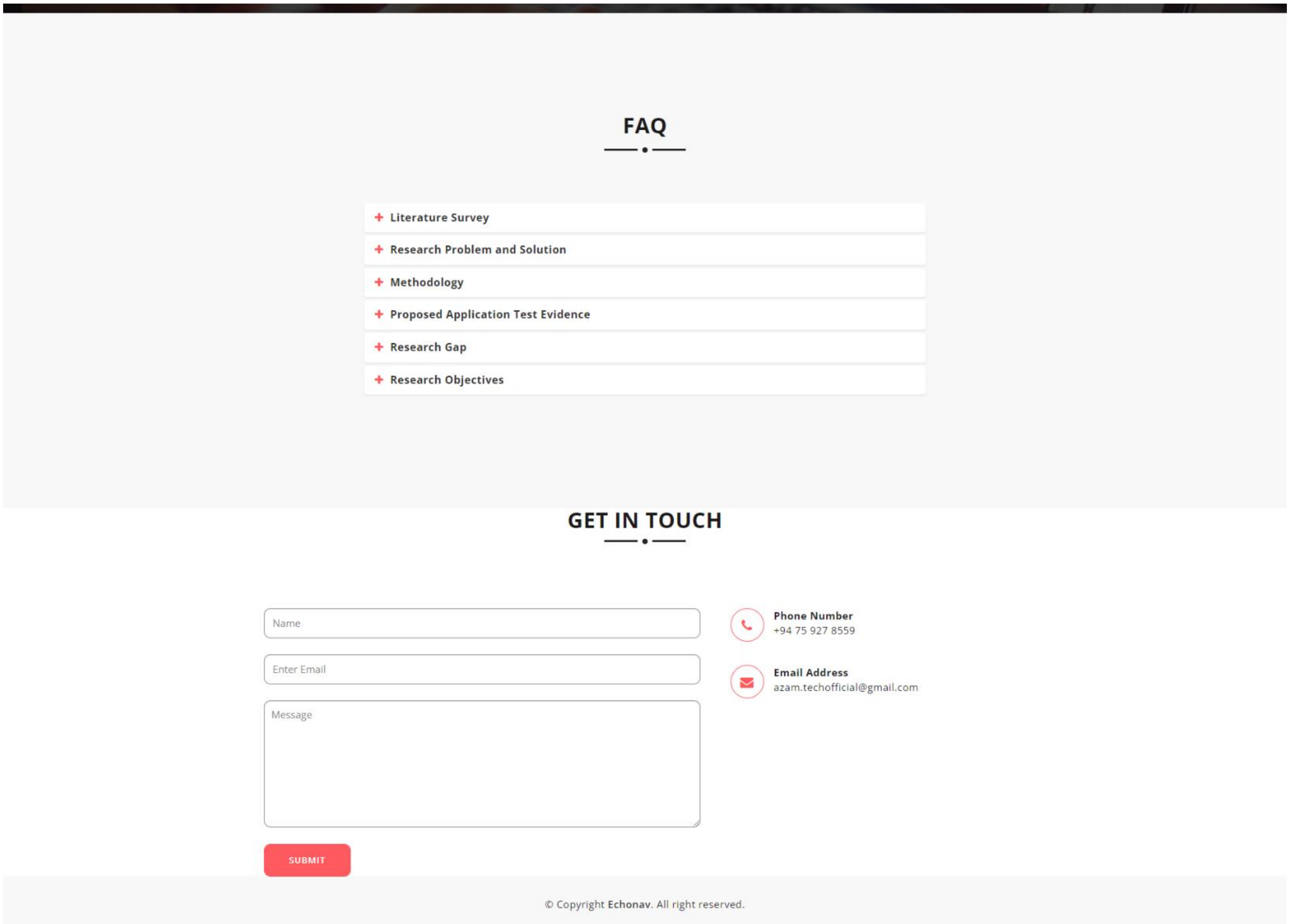

Figure 7 Application lunch website Overview – 3

The website discusses a software that uses image processing and object recognition technologies, such as TensorFlow's YOLOv5 models, for a navigation system designed to aid visually impaired people.

- The program also has a TTS engine for speech output, which enables accessibility and automated answers.
- TensorFlow is used to build the core technology, which enables machine learning activities to be performed on mobile devices.
- The app is built using Google's Flutter framework, an open-source solution for creating attractive apps with a single Dart script.
- The project comprises several papers such as a project proposal, ethics form, project poster, and progress report.

Market research and target demographics:

Understanding the needs, interests, and pain points of the visually impaired population is crucial. Conduct thorough market research to identify the target audience's unique demands. The research should employ questionnaires, interviews, and user testing to get important information.

Developing a Business Model:

Select the optimal business model for each application. Consider selling the software through app stores, offering it as a subscription service, or partnering with organizations that help the visually impaired. Consider price options that create a balance between affordability and sustainability.

Compliance with Accessibility:

Ensure that apps meet accessibility standards and rules, such as the Web Content Accessibility Guidelines (WCAG). This not only enhances the user experience, but also demonstrates dedication to diversity.

Product Evaluation and Improvements:

Collect input from customers on a regular basis to identify areas for improvement. Regularly update apps with user recommendations and developing technology. Prioritize user-centric design and usability.

Data Security and Privacy:

Implement robust data privacy and security protocols, especially in apps that involve text recognition and processing.

Testing Pre-trained models and Implementation

Testing a YOLOv5 model for real-time indoor object detection with classes is a thorough procedure that demands careful attention to detail. YOLOv5, or "You Only Look Once version 5," is a highly accurate and fast object detection method. Serval models were obtained for this project, however only a handful were compatible with the system that was used. Most of the time, these models are evaluated and deployed using Google Collab, which provides a real-time environment. The model developer handles several components of the training process, The developer who constructed the model generated multiple classes. Unfortunately, my suggested method was only able to detect a few crucial phases from one of those class.

Indoor items can be detected, but not all the objects can be detectable. The cause for this is that the Model was created by a third party who did not provide adequate information about his model. So, without a proper explanation, we cannot determine what datasets he utilized to develop this model. Also, because this model is in tflite format, which is mostly compatible with Tensorflow, it can only be viewed once while being created. Once implemented, there is no easy way to access the model and make changes because the developer created these models in Google Collab, to which I do not have access. These are the objects which the pre-trained model can detect so far as for my knowledge.

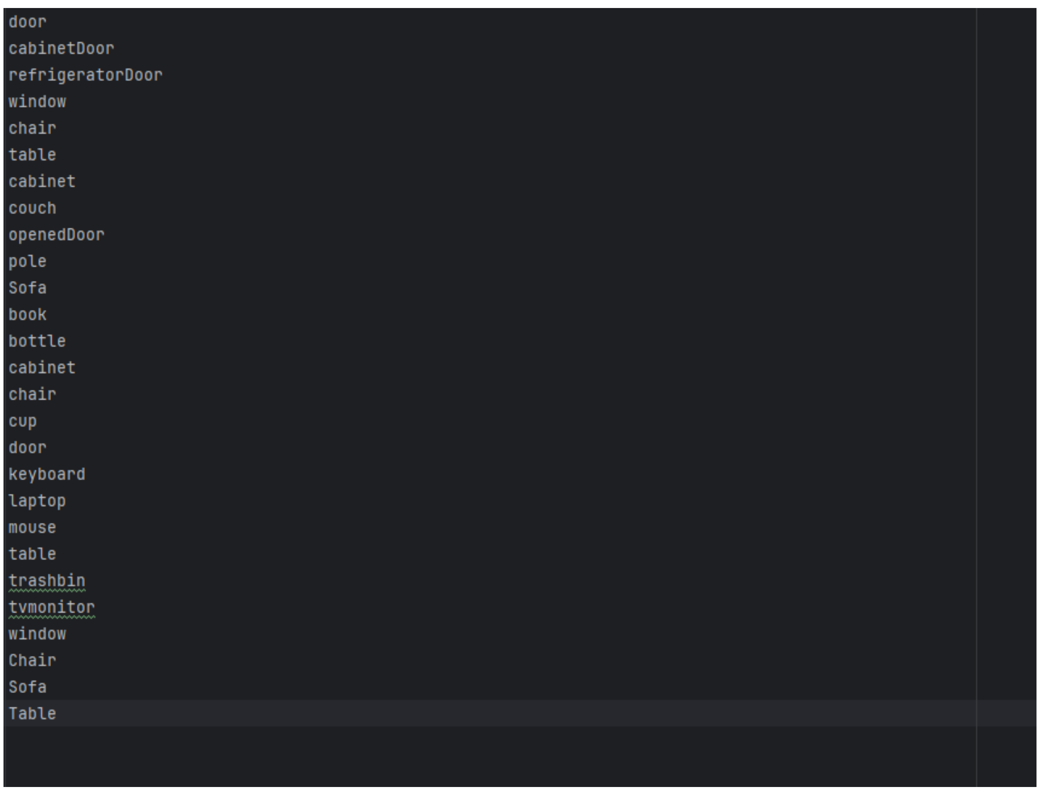

Figure 8 Detectable Objects by the application for navigation

Model Uses Monocular Depth Estimation with Single-Vision Cameras: A Detailed Insight

The model I used for my project uses a method to identify distance from a object in navigation phase. This method is developed and implemented inside the model by the model developer in Kaggle. As for my knowledge from gathering through information the approach he took to make this feature will be following below rules.

Computer vision and optics rely heavily on the use of a single-vision camera to determine distance to objects. This approach applies perspective geometry concepts and may be represented numerically as follows:

Distance to Object (D) = (Object Size in Image (S) * Camera Focal Length (f)) / Apparent Size of Object in Image (A)

This formula uses 'D' as the distance to the item, 'S' as the known physical size, 'f' as the focal length of the camera lens, and 'A' as the apparent size of the object in the acquired image. Each of these parameters is crucial in deciding the distance.

Objects look larger in images as they approach the camera, and smaller as they go away. The focal length of a camera lens impacts how large an item appears in the photograph compared to its actual size. Accurately calculating distance to an item requires knowledge of its size, apparent size in the picture, and camera focal length.

A proper calibration of the camera is essential to assure accuracy. This involves understanding and accounting for both the camera's intrinsic properties (e.g. focus length) and extrinsic factors (e.g. location and orientation in reference to the scene). Calibration ensures the formula appropriately compares the apparent size of an object in a picture to its true size. (Murawski, 2015)

It is vital to understand that this monocular depth estimate approach has limitations. It may struggle with complicated sceneries or irregularly shaped items due to its limited perspective. Accurate depth estimate can be achieved by using many cameras, depth sensors, or sophisticated computer vision algorithms like stereo vision and structure-from-motion. (Rosebrock, 2021)

Distance computation is essential for practical applications like as robotics, augmented reality, object tracking, and autonomous vehicles. Understanding the distance between objects and the camera is crucial for making decisions and interacting with the environment. (Rosebrock, 2021)

Creating a frontend in Flutter for real-time object detection using the YOLOv5 model

```
  double focalLengthFinder(double measuredDistance, double realWidth, double widthInRf) {
    double focalLength = (widthInRf * measuredDistance) / realWidth;
    return focalLength;
}

double distanceFinder(double focalLength, double realObjectWidth, double widthInFrame) {
    double distance = (realObjectWidth * focalLength) / widthInFrame;
    return distance;
}

  Future<void> loadYoloModel() async {
    await vision.loadYoloModel(
        labels: 'assets/navigation/navigation_3.txt',
        modelPath: 'assets/navigation/navigation_3.tflite',
        modelVersion: "yolov5",
        numThreads: 2,
        useGpu: false);
    setState(() {
      isLoaded = true;
    });

  }

  initCamera() async {
    final cameras = await availableCameras();
    vision = FlutterVision();
    controller = CameraController(cameras[0], ResolutionPreset.low);
    controller.initialize().then((value) {
      loadYoloModel().then((value) {

        focal_chair = focalLengthFinder(KNOWN_DISTANCE, CHAIR_WIDTH, chair_width_in_rf);
        focal_table = focalLengthFinder(KNOWN_DISTANCE, TABLE_WIDTH, table_width_in_rf);

        print(focal_chair);
        print(focal_table);

        setState(() {
          isLoaded = true;
          isDetecting = false;
          yoloResults = [];
        });
      });
    });

  }

  Future<void> yoloOnFrame(CameraImage cameraImage) async {
  final result = await vision.yoloOnFrame(
      bytesList: cameraImage.planes.map((plane) => plane.bytes).toList(),
      imageHeight: cameraImage.height,
      imageWidth: cameraImage.width,
      iouThreshold: 0.4,
      confThreshold: 0.4,
      classThreshold: 0.5);
  if (result.isNotEmpty) {
    setState(() {
      yoloResults = result;
    });
```

Figure 9 frontend of the application using flutter

Developing a user-friendly mobile app interface to interact with the model is part of the process. Flutter is a popular open-source UI framework that allows for native app development across mobile, web, and desktop platforms from a single codebase. Here are the steps for creating a Flutter interface for YOLOv5 object detection:

Model Integration with Yolov5

Integrate the YOLOv5 model into your Flutter app to do object detection. Since the model I received from Kaggle already utilizes a TensorFlow Lite model type. The package type is tflite. If you're attempting to construct a model in Google Collab and want to use it in your Flutter app, follow these steps:

Ensure your trained YOLOv5 model is in TensorFlow Lite format (.tflite). Consider using tools such as the PyTorch to TensorFlow Lite converter.

Navigation Model Intergrading with Flutter

```
import ...


class Navigation extends StatefulWidget {...}

class _NavigationState extends State<Navigation> {

  List<dynamic>? _recognitions;

  late CameraController controller;
  late FlutterVision vision;
  late List<Map<String, dynamic>> yoloResults;
  CameraImage? cameraImage;
  bool isLoaded = false;
  bool isDetecting = false;
  int _taps = 0;
  FlutterTts ftts = FlutterTts();
  bool isSpeaking = false;

  double CONFIDENCE_THRESHOLD = 0.4;
  double NMS_THRESHOLD = 0.3;

  //  Distance constants
  double KNOWN_DISTANCE = 62; //INCHES

  double CHAIR_WIDTH = 21; //INCHES
  double TABLE_WIDTH = 33; //INCHES
  double CABINET_DOOR_WIDTH = 15; //INCHES
  double DOOR_WIDTH = 25; //INCHES
  double REFREGERATOR_DOOR_WIDTH = 21; //INCHES
  double WINDOW_WIDTH = 33; //INCHES
  double CABINET_WIDTH = 15; //INCHES
  double COUCH_WIDTH = 33; //INCHES
  double OPEN_DOOR_WIDTH =25; //INCHES
  double POLE_WIDTH = 21; //INCHES
  double SOFA_WIDTH = 50; //INCHES
```

Figure 10 Navigation Model Integrating into flutter - 1

```
double cabinet_width= 200.9855;
double couch_width_in_rf = 378.41183;
double opendoor_width_in_rf = 250.3654;
double pole_width= 240.9855;
double sofa_width_in_rf = 525.41183;

double focal_chair=0.0;
double focal_table=0.0;
double focal_cabinetdoor=0.0;
double focal_door=0.0;
double focal_refregator=0.0;
double focal_window=0.0;
double focal_cabinet=0.0;
double focal_couch=0.0;
double focal_opendoor=0.0;
double focal_pole=0.0;
double focal_sofa=0.0;

double distance=0.0;
String distance_text='';

// Create a queue to store the names of detected objects
final Queue<String> objectQueue = Queue<String>();

@override
void initState() {...}

double focalLengthFinder(double measuredDistance, double realWidth, double widthInRf) {...}

double distanceFinder(double focalLength, double realObjectWidth, double widthInFrame) {...}

Future<void> loadYoloModel() async {
  await vision.loadYoloModel(
      labels: 'assets/navigation/navigation_3.txt',
      modelPath: 'assets/navigation/navigation_3.tflite',
      modelVersion: "yolov5",
      numThreads: 2,
      useGpu: false);
  setState(() {
    isLoaded = true;
  });
```

Figure 11 Navigation Model integrating into flutter - 2

This code uses the device's camera to detect objects using the YOLO (You Only Look Once) concept. Let's break down the main components and features of the code:

Import Statements: The code imports several packages and libraries required for the application's functioning, such as camera management, object detection, text-to-speech (TTS), and others.

Stateful Widget - Navigation: The Navigation class represents a stateful widget that handles the application's principal functionality.

Sate Management: The _Navigation State class controls the state of the Navigation widget. It includes variables for controlling camera features, object identification results, TTS capabilities, and distance calculations.

Initialization: The initState() function performs camera initialization and model loading. It also initializes the text-to-speech capabilities and gives the user first instructions via spoken text.

Camera Initialization: The initCamera() function sets up the camera controller and loads the YOLO model for object detection. It determines focal lengths for various objects using known distances and real-world widths.

Object Detection: The yoloOnFrame() function detects objects in each frame received from the camera stream. It recognizes items like chairs, tables, cabinet doors, windows, doors, and refrigerator doors. When an item is recognized in close proximity, text-to-speech technology generates a voiced warning message.

Distance Calculation: The dimensions and focus lengths of identified objects are used to calculate distance. These distances are used to detect when objects are too close and issue warnings.

Gesture Detection: detects gestures such as drags from the left to the right and from the bottom to the top. Left-to-right drag starts object detection, whereas down-to-up drag stops it.

UI Rendering: The build() function creates the user interface, which displays the camera preview and boxes surrounding identified items. It also has controls for starting and stopping object detection.

Dispose method: When the widget is discarded, the dispose() function is overridden to release resources such as the camera controller and YOLO model.

Overall, this method detects items in real time using the device's camera and issues voiced warnings when certain objects are found in near vicinity. It explains how to integrate the camera, object detection model, and text-to-speech capability to construct a navigation assistance system.

Supermarket Object Detection Model Intergrading with Flutter

```
import ...

class SuperMarket extends StatefulWidget {...}

class _SuperMarketState extends State<SuperMarket> {

  List<dynamic>? _recognitions;

  late CameraController controller;
  late FlutterVision vision;
  late List<Map<String, dynamic>> yoloResults;
  CameraImage? cameraImage;
  bool isLoaded = false;
  bool isDetecting = false;
  int _taps = 0;
  FlutterTts ftts = FlutterTts();
  late Map<String, dynamic> label;

  @override
  void initState() {...}

  Future<void> loadYoloModel() async {...}

  initCamera() async {...}

  Future<void> yoloOnFrame(CameraImage cameraImage) async {...}

Future<void> startDetection() async {
setState(() {
  isDetecting = true;
});
if (controller.value.isStreamingImages) {...}
await controller.startImageStream((image) async {
  if (isDetecting) {...}
});
}

Future<void> stopDetection() async {...}

    List<Widget> displayBoxesAroundRecognizedObjects(Size screen) {...}
```

Figure 12 Supermarket object detection model integrate

This code segment is similar to the previous one, but focuses on recognizing things in a supermarket context. Here's a breakdown of its features:

Import Statements: Similar to the preceding code, this part imports the packages and libraries required for Flutter development, such as camera management, object identification, and text-to-speech capability.

Stateful Widget - Supermarket: The SuperMarket class is a stateful widget that handles the supermarket detecting application's core functionality.

State Management: The SuperMarket widget's state is managed via the _SuperMarketState class. It includes variables for controlling camera parameters, object detection results, and TTS functionality.

Initialization: The initState() function sets up the camera and loads the YOLO model for object recognition. It also initializes the text-to-speech capabilities and gives the user first instructions via spoken text.

Object Detection: The yoloOnFrame() function detects objects in each frame received from the camera stream. It recognizes numerous objects at the shop and uses text-to-speech to read their labels.

Gesture Detector: detects gestures such as drags from the left to the right and from the bottom to the top. Left-to-right drag starts object detection, whereas down-to-up drag stops it.

UI Rendering: The build() function creates the user interface by displaying the camera preview and boxes around detected items. It also has controls for starting and stopping object detection.

Dispose function: When the widget is discarded, the dispose() function is overridden to release resources such as the camera controller and YOLO model.

Overall, this code segment is designed to recognize things in a supermarket setting in real time using the device's camera and provide verbal feedback on the found objects.

Application Overview with testing Evidence

This Android smartphone application is a complex tool meant to improve user experiences by including novel features such as item identification in retail surroundings and navigation aid. The program has a user-friendly interface that was meticulously designed to allow smooth interactions and straightforward navigation. When users start the app, they are presented with a simple design that provides straightforward alternatives for accessing the main features: navigation and supermarket object identification.

The navigation function allows users to obtain real-time guidance by utilizing the device's camera and powerful computer vision algorithms. Users commence the navigation process with a simple left-to-right drag motion, and the system automatically analyzes the surroundings and gives verbal directions based on recognized items and geographical context. The program identifies items like seats, tables, doors, and windows and alerts users to potential barriers or areas of interest along the way. Text-to-speech technology is integrated to provide users with auditory clues regarding recognized items, improving accessibility and assuring a seamless navigating experience.

On the other hand, the supermarket object identification capability allows users to recognize numerous things in a shopping context. When activated, the program uses the device's camera to scan the area for things like seats, tables, and refrigerators, among others. Users may see identified items highlighted inside the camera view, as well as useful text labels explaining the discovered item, thanks to an easy-to-use interface. Furthermore, the program provides spoken feedback, proclaiming the names of discovered items in real time, allowing for seamless interaction and increasing user engagement.

Supermarket Object Detection Test Evidence: https://www.youtube.com/watch?v=7ngiIXdEISA

Navigation Test Evidence: https://www.youtube.com/shorts/klPqLWR8vpw

In the above links Extensive testing on numerous models was carried out to verify the application's reliability and performance across diverse devices and settings. The given application testing data demonstrates the application's resilience and adaptability, with consistent performance across several mobile devices. Whether traversing complicated landscapes or recognizing products in a supermarket, the program produces consistent results, demonstrating its usefulness and usability across a wide range of usage situations.

In essence, this Android mobile application is a cutting-edge solution that uses modern technologies like computer vision, text-to-speech, and intuitive user interfaces to provide better user experiences. The program provides users with important features and functions that revolutionize how they engage with their mobile devices and traverse their environments, from helping navigation in new locations to assisting with grocery purchasing.

Figure 13 Application Interface

Research Findings

The study demonstrates the significant impact that innovative technology solutions may have on the lives of visually impaired individuals. Robust development and testing have shown that smart mobile applications, such as environmental monitoring and navigation, are both technically capable and user-friendly.

They offer tangible benefits such as increased autonomy, confidence, and active participation in everyday activities. The study found that improving item detection accuracy, immediate processing speed, and user feedback led to beneficial outcomes. These findings highlight the need for ongoing changes and innovations to adapt to the rapidly evolving technology landscape. The research highlights the revolutionary potential of these technologies to increase inclusion and independence for blind and visually impaired individuals, paving the way for a more accessible future.

Application Improvement

Using cutting-edge technology and innovative design concepts to fulfill the needs of visually impaired individuals is truly admirable. This initiative aims to improve community accessibility and independence through a unique approach. Let's delve a bit more into the specifics and implications of this endeavor.

The usage of YOLOv5 for object identification is a key aspect of this research. Visually impaired individuals rely heavily on object detection to navigate their environment effectively. This system's accuracy in recognizing objects and providing real-time information is amazing. Using Flutter and Dart for mobile app development is feasible because to their cross-platform compatibility. The inclusion of flutter_tts for aural feedback enhances system usability.

However, it is vital to study the accuracy and speed of object detection.
Users rely heavily on these tools to navigate the purchasing experience smoothly. To maintain relevance and usefulness, the YOLO model and mobile app require continuous updates and improvements.

Implementation on new features:

We can also implement new features such as color detection and bill utilizing. The below method can be used to develop these features into the proposed system.

Color Detection:

As you may be aware I talked about Convolutional Neural Networks (CNNs) in my project proposal to develop this color detection feature too. Unfortunately for the lack of time I didn't implement it. But this is the system design to implement such feature.

Color recognition with Convolutional Neural Networks (CNNs) is a groundbreaking breakthrough. Ability to choose clothing colors independently increases self-confidence and autonomy in visually impaired individuals.

To implement color detection:

Server-side: Use a color detecting method in your server code. Depending on your server-side technology stack, you may use a variety of libraries and techniques to identify colors in photos.

Client side (Flutter): Modify the code so that the color information from the server response is parsed and shown to the user in a relevant fashion. You may wish to display the detected colors or any other information about them in the alert dialog.

Once you've built server-side color detection, you'll need to modify the Flutter code to process the response appropriately and show the color information obtained from the server. Using speech recognition and image processing to create a garment color detection and suggestion system is an effective aid for visually impaired individuals. Its simplicity, accuracy, real-time feedback, customization, and straightforward communication reduce obstacles and promote confidence and independence. This step improves the quality of life for visually impaired individuals by providing a user-friendly tool for making educated wardrobe choices. This phase symbolizes the start of a journey to build assistive technology for the community's specific requirements, resulting in a more inclusive future.

Bill Utilizing

Since the text to Speech Functions are already implemented in The proposes system it is easy to implement this feature too. OpenCV will be used to do this process.

- **OpenCV:** is a necessary tool for preparing and improving utility bill pictures. This utility optimizes photos for OCR by enhancing, reducing noise, and detecting edges.

Our Voice-Controlled Utility Bill Scanning App for Blind People may be brought to life with a strong technology stack and precise development methods throughout this vital implementation phase. This stage focuses on integrating important technologies and creating a cohesive user experience.

To create a feature that recognizes text on a bill and responds with the outcome by voice using text-to-speech (TTS), follow these steps:

Text Recognition: Include a text recognition library or API in your application. This library should be able to extract text from pictures, such as invoices and receipts. Popular choices include Google's Mobile Vision API, Tesseract OCR, and the text recognition tool in Firebase ML Kit.

Image snap: Include a feature in your program that allows users to snap photographs of invoices. This may entail utilizing the device's camera capabilities, similar to the code you gave.

Text Extraction: After capturing a picture of the bill, utilize a text recognition tool or API to extract the text. This text will most likely include item names, pricing, and other bill-related information.

Text Processing: Analyze the captured text to find useful information. Depending on the requirements of your application, you may need to use logic to parse the language and extract certain facts, such as the total amount, things purchased, or any other relevant information.

Outcome Determination: Based on the retrieved information, decide what outcome you want to express to the user. For example, you might notify the user of the total amount on the bill or offer a summary of the things purchased.

Text-to-Speech (TTS): Use a text-to-speech library or API to turn the results into spoken words. This enables your program to respond to the user verbally. Libraries like as Flutter TTS or native TTS APIs are accessible for both iOS and Android devices.

Voice Output: Finally, play the synthetic speech output for the user. This might include playing the TTS output over the device's speaker or headphones, depending on the user's preferences.

By following these steps, you may create a feature that recognizes text on a bill, processes it, and responds with the result by voice utilizing text-to-speech capabilities.

Summery

This research highlights innovative mobile accessibility app designed to enhance the lives of visually impaired persons. These applications, including environmental sensing and navigation, having the potential to improve the way persons with visual impairments interact with their surroundings.

The study highlights mobile applications' ability to foster autonomy, independence, and social inclusion. This community's challenges are addressed with a combination of cutting-edge technology including YOLOv5, CNNs, and OCR tools, as well as user-centric design principles. The findings emphasize the significance of user interaction and cooperation during the development process. Involving visually impaired individuals in user testing and feedback loops demonstrated the usability of the apps and highlighted the need of accessibility in technology design. The study emphasizes the adaptability and scalability of these systems. As technology advances, these apps may be adjusted and enhanced to address a wider variety of challenges, making them future-proof and dynamic tools for enhancing the lives of visually impaired individuals.

These mobile accessibility applications promote inclusion and independence for visually impaired individuals. With continual refinement and development, technology has the ability to revolutionize society and empower individuals.

Conclusion

Research on smart mobile accessibility applications and sub-components has led to increased freedom, participation, and empowerment for visually impaired individuals. This technology integrates image processing, machine learning, and natural language skills to empower blind individuals in their daily lives.

Combining image processing with voice command features in a mobile app is a significant advancement for the independence, safety, and inclusion of the visually impaired.
This groundbreaking system provides real-time object identification, increased navigation aid, and quick information access, boosting users' awareness and confidence in navigating their environment. Technology has the ability to change accessibility and mobility solutions. A focus on user-centered design, privacy safeguards, and ongoing technical breakthroughs is creating a more egalitarian future for the visually impaired population.

Integrating speech recognition and image processing technologies improves market access, promoting inclusion and accessibility. This comprehensive solution, including YOLOv5, Flutter, and Dart, enables consumers to traverse the market autonomously and confidently. User input enhances freedom and confidence in traversing areas, demonstrating a commitment to continual growth and ethical issues.

References

Ajinkya Badave, Rathin Jagtap, Rizina Kaovasia, Shivani Rahatwad, Saroja Kulkarni, 2020. *www.ieeexplore.ieee.org.* [Online]
Available at: https://ieeexplore.ieee.org/document/9102694
[Accessed 24 05 2024].

Alexy Bhowmick & Shyamanta M. Hazarika , 2017. *www.springer.com.* [Online]
Available at: https://link.springer.com/article/10.1007/s12193-016-0235-6
[Accessed 24 05 2024].

Bineeth Kuriakose, Raju Shrestha and Frode Eika Sandnes, 2022. *www.tandfonline.com.* [Online]
Available at: https://www.tandfonline.com/doi/full/10.1080/02564602.2020.1819893
[Accessed 24 05 2024].

Chu-Sing Yang, Ko-Wei Huang, Hsing-Wei Peng, Yen-Min Su, 2019. *www.ieeexplore.ieee.org.* [Online]
Available at: https://ieeexplore.ieee.org/document/8959877
[Accessed 24 05 2024].

Devi RA, Uthaman I, Shanthanam RR, 2020. *An IOT Security based electronic aid for visually impaired detection with navigation assistance system.* [Online]
[Accessed 24 05 2024].

Díaz-Toro, A. A., 2021. *www.hindawi.com.* [Online]
Available at: https://www.hindawi.com/journals/js/2021/6685686/
[Accessed 24 05 2024].

Kanak Manjari, Madhushi Verma and Gaurav Singal, 2020. *www.linkinghub.elsevier.com.* [Online]
Available at: https://linkinghub.elsevier.com/retrieve/pii/S254266052030024X
[Accessed 24 05 2024].

Liang-Bi Chen, Jian-Ping Su, Ming-Che Chen, Wan-Jung Chang, Ching-Hsiang Yang, and Cheng-You Sie, 2019. *www.ieeexplore.ieee.org.* [Online]
Available at: https://ieeexplore.ieee.org/document/8661943
[Accessed 24 05 2024].

Murawski, K., 2015. *www.researchgate.net.* [Online]
Available at:
https://www.researchgate.net/publication/279275407_Method_of_Measuring_the_Distance_to_an_Object_Based_on_One_Shot_Obtained_from_a_Motionless_Camera_with_a_Fixed-Focus_Lens
[Accessed 24 05 2024].

Muthuraman, V., 2019. *www.researchgate.net.* [Online]
Available at:
https://www.researchgate.net/publication/331590226_A_Study_on_Text_Recognition_using_Image_Processing_with_Datamining_Techniques
[Accessed 24 05 2024].

npcbvi.gov.in, 2020. *www.npcbvi.gov.in.* [Online]
Available at: https://npcbvi.gov.in/writeReadData/mainlinkFile/File341.pdf
[Accessed 24 05 2024].

P. Strurniłło, K. Matusiak, P. Skulimowski, 2013. *www.ieeexplore.ieee.org.* [Online]
Available at: https://ieeexplore.ieee.org/document/6577868
[Accessed 24 05 2024].

Prasad, P., 2021. *www.researchgate.ne.* [Online]
Available at:
https://www.researchgate.net/publication/353180608_Real_Time_Indoor_Object_Detection_Aid_for_Blind
[Accessed 24 05 2024].

Rosebrock, A., 2021. *www.pyimagesearch.com.* [Online]
Available at: https://pyimagesearch.com/2015/01/19/find-distance-camera-objectmarker-using-python-opencv/
[Accessed 24 05 2024].

World Heath Organization, 2019. *www.who.int.* [Online]
Available at: https://www.who.int/publications/i/item/world-report-on-vision
[Accessed 24 05 2024].

Yimin Lin, Kai Wang, Wanxin Yi, Shiguo Lian, 2019. *www.ieeexplore.ieee.org.* [Online]
Available at: https://ieeexplore.ieee.org/document/9022410
[Accessed 24 05 2024].

Gantt Chart

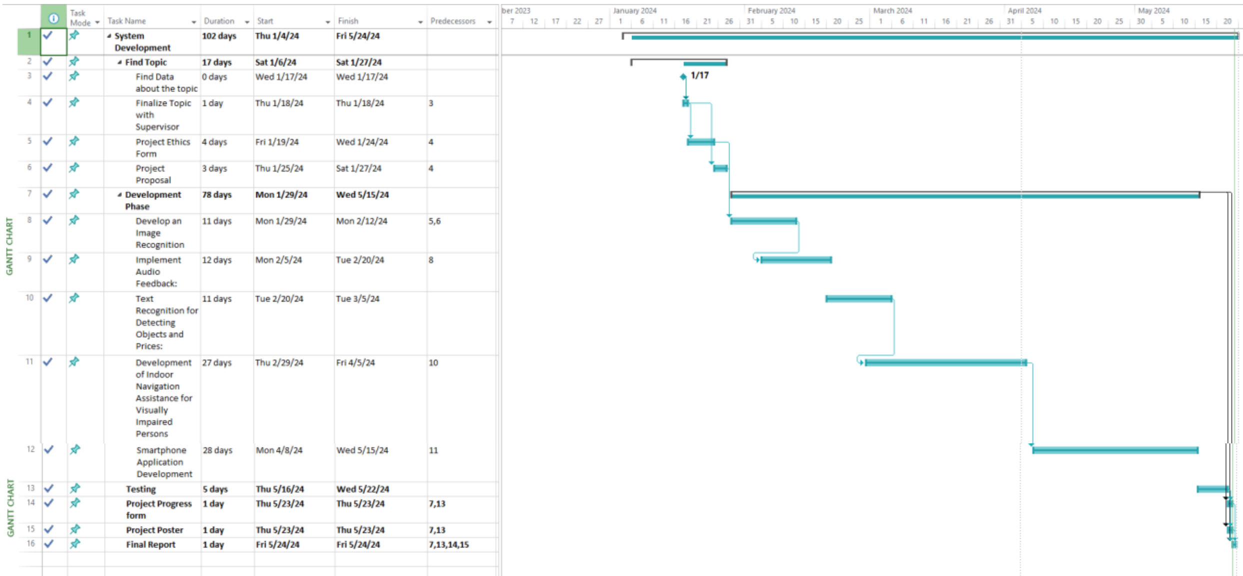

Figure 14 Gantt Chart

As indicated by the Gantt chart above, all jobs were performed effectively, with no omissions or delays. Each activity was completed within the timeframe specified, with scrupulous adherence to the established timetable. There are no outstanding tasks to do, and there has been no variation from the expected timetable. The project has gone successfully, with each milestone completed on schedule and without the need for extra time. This outstanding performance demonstrates the value of rigorous preparation and dedicated execution in guaranteeing the project's success and timely completion.

Work Break down Structure

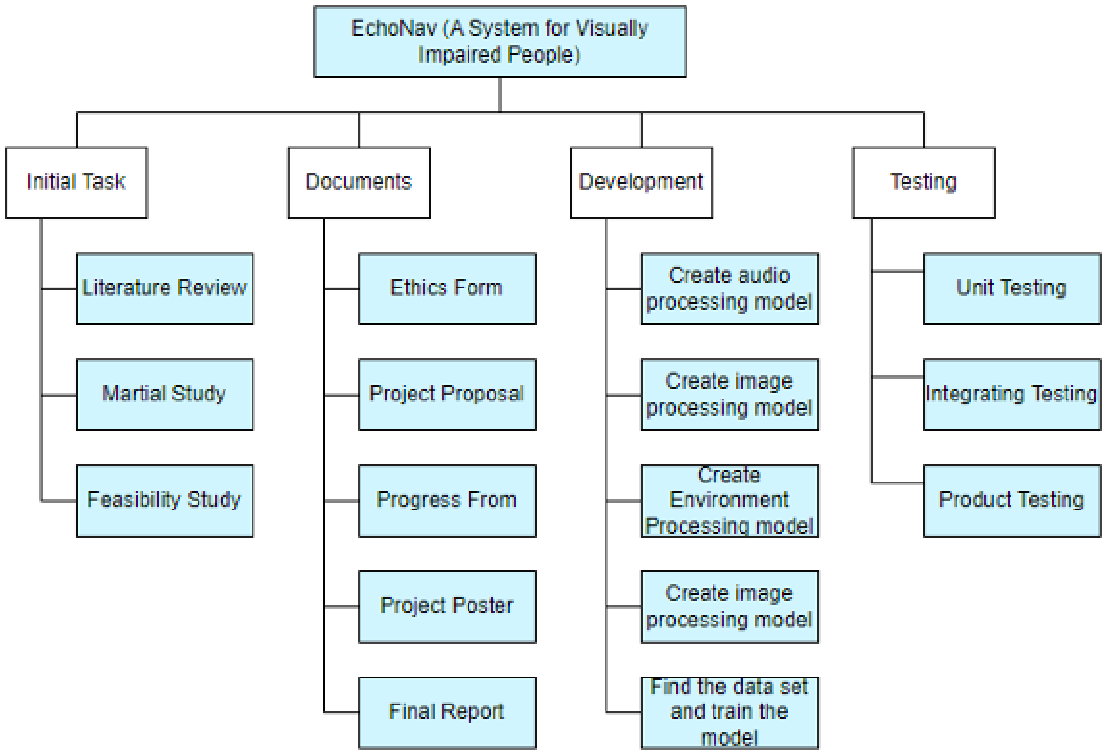

Figure 15 Work Break down Structure

Initial Tasks

- Literature Review [Completed]
- Martial Study [Completed]
- Feasibility Study [Completed]

Documents

- Ethics Form [Completed]
- Project Proposal [Completed]
- Progress Report [Completed]
- Final Report [Completed]
- Project Poster [Completed]

Development

- Create audio processing model [Completed]
- Create image processing model [Completed]
- Create environment processing model [Completed]
- Find the data set and train the model [Completed]

Testing

- Unit Testing [Completed]
- Integrating Testing [Completed]
- Product Testing [Completed]

Website

- Additionally create a application lunch website [Completed]