A

Project Report on

Vision Assist:

Submitted in partial fulfillment of completion of the course

Advanced Diploma in IT, Networking and Cloud

Submitted by:

AKASH T

MUHAMMED HIDASH

SAHAL MUHAMMED K K

ARCHANA G S

Under Guidance of:

IBM Mentor Mr.ATUL PANDEY

Edunet Mentor Mr.ADWAID R S

Year 2024

Abstract

Vision Assist is an AI-powered mobile application designed to improve the independence, safety, and quality of life for visually impaired individuals. By utilizing advanced technologies such as object detection, facial recognition, and natural language processing, the app offers real-time assistance in navigating daily tasks. Key features of Vision Assist include object identification, spatial description, facial recognition, activity recognition, voice feedback, text reading, and real-time currency recognition. These features enable users to interact with their surroundings more effectively, fostering a greater sense of autonomy and ease in day-to-day activities.

Acknowledgement

I would like to sincerely thank everyone involved in the development of Vision Assist. Special thanks to IBM mentor Mr. Athul Pandey, Edunet mentor Mr. Adwaid R S, and Master Trainer Arpita Roy for their guidance and expertise in AI and ML. I would also like to express my gratitude to my team members, Akash T, Muhammed Hidash, Sahal Muhammed K K, and Archana G S, for their dedicated efforts in developing key features such as object detection, facial recognition, Currency Recognition and frontend. Their hard work and collaboration were crucial in bringing this project to life. Thank you to all who contributed to its success.

Team Composition and Workload Division

Akash. T	:	Currency recognition, Object identification, Backend.
Archana G.S	:	Activity recognition, Voice feedback, Spatial description.
Muhammed Hidash	:	Text reading, Facial recognition, Spatial recognition.
Sahal Muhammed K.K	:	Frontend.

Table of Contents

1. Introduction to Problem

Visually impaired individuals face significant challenges in navigating their surroundings, identifying objects, and recognizing faces, which limits their independence and quality of life. Traditional aids, like canes and guide dogs, offer assistance but have limitations in addressing these needs. Modern technologies, particularly AI and computer vision, present an opportunity to provide more advanced and versatile solutions. Vision Assist leverages these technologies to offer real-time support, helping visually impaired individuals navigate their environment and perform daily tasks with greater autonomy and ease.

2. Literature Review

Previous research and development efforts in assistive technology for the visually impaired have focused on various solutions:

1. **Canes and Guide Dogs:** Traditional aids that provide tactile feedback or guide visually impaired individuals.
2. **Electronic Travel Aids (ETAs):** Devices that use ultrasonic waves to detect obstacles and provide feedback.
3. **Mobile Applications:** Apps like Seeing AI and Be My Eyes that offer object recognition and assistance through crowd-sourced help.
4. **AI and Computer Vision**: Recent advancements in machine learning and computer vision have enabled real-time object detection, facial recognition, and depth estimation.

Despite these advancements, there remains a gap in providing a comprehensive, user-friendly, and accessible solution that integrates multiple functionalities into a single mobile application.

3. Proposed Solution

Vision Assist is an AI-powered mobile application designed to enhance the independence and safety of visually impaired individuals. The app leverages advanced technologies such as object detection, facial recognition, depth estimation, and natural language processing to provide real-time assistance in various aspects of daily life. Key features include:

- **Object Identification:** Recognizes and names objects in the user's environment.
- **Spatial Description:** Describes the spatial arrangement and positioning of objects.
- **Facial Recognition:** Identifies familiar people in the user's vicinity.
- **Activity Recognition:** Describes the activities people are engaged in.
- **Voice Feedback and Interaction:** Provides audible descriptions and guidance.
- **Text Reading:** Reads text from various sources aloud.
- **Real-Time Currency Recognition:** Detects and identifies different denominations of currency.

4. Requirements

4.1 Technology Stack

- **Programming Languages:** Python
- **Frameworks:** Tensor Flow, PyTorch
- **APIs**: OpenAI API
- **Libraries:** OpenCV, Tesseract (for OCR)

4.2 Hardware

- **Smartphone**: With a high-resolution camera, sufficient processing power, and internet connectivity.

4.3 Software

- **Mobile Operating System:** Android or iOS
- **Development Environment:** Android Studio, VScode

4.4 Deployment Environment

- **Mobile Application Store**: Google Play Store for Android and Apple App Store for iOS.
- **Backend Server**: For model updates and potential cloud-based processing.

5. User Requirements

- **Accessibility:** User-friendly interface with large buttons and voice commands
- **Real-Time Feedback:** Immediate audio feedback for detected objects and spatial descriptions.
- **Customization:** Ability to customize feedback preferences and personal settings.
- **Reliability:** Consistent and accurate object detection and description.
- **Privacy:** Secure handling of personal data and images.

6. Design Documentation

- **User Interface (UI):** Designed with high contrast, large fonts, and intuitive navigation. Includes voice command options for ease of use.
- **System Architecture:**
    - **Client-Side:** Mobile application capturing real-time video, processing frames locally or sending to a backend server for processing.
    - **Server-Side:** Optional backend server for processing heavy computations or storing user data.

7. Implementation Details

- **Object Detection:** Implemented using YOLOv3 model fine-tuned on a diverse dataset of objects.
- **Depth Estimation:** Using MiDaS model for monocular depth estimation.
- **Facial Recognition:** Using a pre-trained model like FaceNet, fine-tuned for recognizing familiar faces.
- **Natural Language Processing:** Using OpenAI's GPT-4 API or a LLaMA model to generate descriptive text based on detected objects and their spatial relationships.
- **Integration:** Combining outputs from detection and depth models to generate comprehensive spatial descriptions.

8. Testing

- **Unit Testing**: For individual components like object detection, depth estimation, and text-to-speech.
- **Integration Testing**: Ensuring seamless interaction between different components.
- **User Testing:** Conducting trials with visually impaired users to gather feedback and make necessary improvements.
- **Performance Testing:** Ensuring real-time performance and low latency in providing feedback.

9. Deployment

- **Build and Package:** Prepare the app for deployment on Android and iOS platforms.
- **Submit to App Stores:** Follow guidelines for submitting apps to Google Play Store and Apple App Store.
- **Monitor and Update:** Continuously monitor app performance and user feedback, providing updates and improvements as needed.

10. Future Scope

- **Navigation Assistance:** Integrate turn-by-turn navigation with obstacle avoidance.
- **Indoor Navigation:** Use Bluetooth beacons or Wi-Fi positioning for indoor navigation.
- **Expanded Language Support:** Provide support for multiple languages for voice feedback.
- **Crowdsourced Hazard Reporting:** Allow users to report hazards and share with other users.
- **Enhanced Depth Estimation:** Incorporate dual-camera smartphones or external sensors for more accurate depth estimation.

11. Conclusion

Vision Assist aims to provide a comprehensive, accessible solution to enhance the independence and safety of visually impaired individuals. By integrating advanced AI and computer vision technologies, the app offers real-time object identification, spatial description, facial recognition, and more. With continuous improvements and user feedback, Vision Assist strives to become an essential tool for visually impaired users, significantly improving their quality of life.

Appendix A Project Code

https://github.com/akzgit/Vision_Assist.git

Appendix B Screenshot of Project

Appendix C abbreviation

References