INDUSTRIAL PROJECT REPORT

Internship project report on

Smart Vision for Visually Impaired

submitted in partial fulfilment of the
Requirements for the award of

Bachelor of Engineering
in
School of Electronics and Communication Engineering

Carried out at

Einetcorp Pvt.Ltd

Submitted By-
A S V Dheeraj
01FE21BEC161

Under the guidance of

Prof. Sheela A B
College Guide
KLE Technological University

Mr. Manoj Bhat
Industry Guide
Einetcorp Pvt.Ltd

SUBMITTED TO:
School of Electronics and Communication Engineering
KLE TECHNOLOGICAL UNIVERSITY
Hubballi

K.L.E SOCIETY'S
KLE Technological University,
HUBBALLI-580031
2024-2025

SCHOOL OF ELECTRONICS AND COMMUNICATION ENGINEERING

CERTIFICATE

This is to certify that the internship project entitled **"Smart Vision for Visually Impaired"** is a bonafide work carried out by **"A S V Dheeraj bearing University Seat No. 01FE21BEC161 in Einetcorp Pvt.Ltd, in partial fulfillment for the award for Bachelor of Engineering in Electronics and Communication in the School of Electronics and Communication Engineering of KLE Technological University, Hubballi for the academic year 2024-2025.**

Prof. Sheela A B	**Dr. Suneeta V. Budihal**	**Dr. Basavaraj S. Anami**
Guide	**Head of School**	**Registrar**

External Viva:

Name of Examiners **Signature with date**

1.

2.

DECLARATION

I hereby declare that the Industrial Project Report entitled **"Smart Vision for Visually Impaired"** is an authentic record of my own work as requirements of Industrial Project during the period from 15/01/2025 to 30/06/2025 for the award of degree of B.E. KLE Technological University, Hubballi under the guidance of **Prof. Sheela A B** and Industry guide **Mr. Manoj Bhat**

A S V Dheeraj
01FE21BEC161

Date :

ACKNOWLEDGMENT

The sense of contentment and elation that accompanies the successful completion of the Industrial project report and the report would be incomplete without mentioning the names of the people who helped in accomplishing this. I absolutely respect the inspiration, support and steering of all the people who have been instrumental in making this project fulfillment. I being the student of KLE Technological University, feel extremely grateful to Einetcorp Private Limited and the college for self assurance best owed in us and for entrusting our mission entitled **Einetcorp Private Limited Internship.**

It gives me immense pleasure to express my deepest sense of gratitude and sincere thanks to my highly respected and esteemed guides **Mr. Manoj Bhat** for their valuable guidance, encouragement and help for completing this work. Their useful suggestions for this whole work and co-operative behavior is sincerely acknowledged.

I also express my gratitude to **Sheela A. B.**, University guide, for her constant support and guidance. I also wish to express my gratitude to **Dr. Suneeta V. Budihal**, HOS of School of Electronics and Communication Engineering.

I would like to express my sincere thanks to **Dr. Ashok Shettar**, Vice Chancellor, KLE Technological University, Hubli for his kind-hearted support and for giving us this opportunity to undertake this project. Also, I like to thank **Mr. P.G.Tewari**, Principal for his whole-hearted support.

-A S V Dheeraj

ABSTRACT

This project focused on building a real-time assistive system for the visually impaired by integrating Optical Character Recognition (OCR) and Text-to-Speech (TTS) capabilities on an embedded AI platform. Using the Blaize Xplorer X600M M.2 accelerator in combination with the Raspberry Pi Compute Module 5 (CM5), the system was designed to capture visual text from the environment and convert it into clear audio feedback. PaddleOCR was employed for efficient text detection and recognition, offering robust performance even in real-world scenarios such as angled views and varying lighting conditions. The recognized text was then passed to a VITTS-based TTS engine to generate natural-sounding voice output in English. Basic object detection features were also integrated to provide spatial awareness about surrounding entities.

To enhance usability, the system was integrated with GPIO-based physical inputs on the Raspberry Pi, enabling intuitive control over different modes of operation. This allowed the assistive system to function without a traditional interface, making it accessible and practical for real-world deployment. The complete pipeline was optimized with frame skipping, threading, and low-latency processing to ensure responsive, real-time performance. This project enabled the team to gain hands-on experience in deploying AI workloads on edge hardware and designing assistive technologies aimed at improving accessibility.

EINETCORP

CIN
U72900KA2021PTC145676

Certificate of Internship

This is to certify that **A S V Dheeraj** bearing USN: 01FE21BEC161 has successfully completed an internship with EiNETCORP Pvt Ltd as a Machine Learning intern from the **15th of Jan 2025 to the 30th of June 2025.**

During his tenure of internship, **Mr. A S V Dheeraj** has worked on Implementing Optical Character Recognition (OCR) on Embedded Systems and Developing an Application Layer Pipeline and similar activities.

Besides exhibiting high comprehension capacity, he has demonstrated his skills with self-motivation to learn new ones. He has maintained an outstanding professional demeanor and showcased excellent moral character throughout the internship period.

We hereby certify that the candidate's overall work is good and satisfactory to the best of our knowledge.

Wishing the candidate all the best in his future endeavors.

For EINETCORP PVT. LTD.

DIRECTOR

Manoj Bhat,
Founder Director
EiNETCORP Pvt Ltd

EINETCORP PVT. LTD. HUBLI.

//CERTIFIED TRUECOPY//

Phone +91 876 223 3319 | **Email** connect.einetcorp@gmail.com | KLE TECH PARK, BVB CAMPUS, Hubballi, 580031

Contents

1 Introduction **10**
- 1.1 Overview of the Internship . . . 10
- 1.2 Company Profile . . . 11
- 1.3 Project Background . . . 11
- 1.4 Problem Statement . . . 12
- 1.5 Project Objectives . . . 12
- 1.6 Organization of the Report . . . 12

2 Literature Survey **14**

3 Methodology **17**
- 3.1 OCR Pipeline . . . 17
  - 3.1.1 Text Detection . . . 17
  - 3.1.2 Text Recognition . . . 18
- 3.2 Text to Speach . . . 18
  - 3.2.1 VITTS Architecture . . . 19
  - 3.2.2 Model Working . . . 19
- 3.3 System Integration . . . 19
  - 3.3.1 GPIO/Trigger Input Handling . . . 19
  - 3.3.2 Optimization : Frame Skipping & MultiThreading . . . 20
  - 3.3.3 Audio Routing . . . 20
- 3.4 Deployment on RPi CM5 . . . 20

4 Results and Discussions **22**

5 Conclusion and Future Scope **25**
- 5.1 Conclusion . . . 25
- 5.2 Future Scope . . . 25

References **26**

List of Figures

3.1 OCR Model Block Diagram 17

4.1 Result of Object Detection task 22
4.2 Result of Optical Character Recognition task 23
4.3 Integrated System Testing 23
4.4 Final Prototype 24

Chapter 1

Introduction

In this chapter, this introduction will encompass the motivation behind the project, a survey of relevant literature, a clear problem statement, and the specific objectives that need to be achieved to complete the project.

Rapid advances in artificial intelligence (AI), especially in the fields of deep learning and computer vision, have revolutionized a number of industries, including assistive technology, robotics, healthcare, and transportation. Among them, the creation of AI-powered tools for the blind and visually impaired has considerable potential as it aims to close the accessibility gap and encourage more self-reliance in day-to-day activities. Real-time observation and interaction in dynamic surroundings are now feasible because to the development of effective edge computing platforms, which allow complex AI algorithms to be directly deployed on low-power embedded devices.

Edge AI solutions lower latency and reliance on cloud infrastructure by combining the computing power of accelerators with the adaptability of embedded devices to analyze data locally. This change in technology is especially advantageous for assistive applications, where privacy and prompt response are essential. In this regard, real-time scene analysis, object identification, and text recognition capabilities may be provided by AI-driven smart vision systems, empowering visually impaired individuals to acquire contextual awareness and make well-informed choices.

1.1 Overview of the Internship

The internship provided a thorough introduction to the construction of edge AI systems, with a focus on both theoretical understanding and real-world application. The training, which took place at Einetcorp Pvt. Ltd. between January 15, 2025, and June 30, 2025, concentrated on creating an AI-based assistive vision system specifically designed for people with visual impairments. Using a Raspberry Pi CM5 platform and the Blaize Xplorer X600M AI inference accelerator, the internship focused on implementing real-time object identification and OCR models.

Starting with the Blaize Picasso SDK setup and configuration, which included installing drivers, configuring PCIe connectivity, and launching example AI apps, the intern participated in several phases of the development lifecycle. Subsequently, deep learning models were converted and optimized using NetDeploy, a Blaize tool for converting standard ONNX models into hardware-compatible forms. GStreamer made it easier to build the pipeline, allowing for real-time image processing, inference, and feedback systems.

Apart from integrating hardware and software, the intern created unique pretreatment

and postprocessing modules, set up GPIO interfaces to switch between application modes, and put in place a text-to-speech (TTS) engine for audio output. Together, these elements created an interactive embedded system that could read text and recognize objects in real time. A functional prototype was deployed at the end of the internship, confirming the system's efficacy in real-world situations and offering practical experience in developing embedded AI applications.

1.2 Company Profile

Einetcorp is a Embedded AI company based in KLE Techpark Incubation Center Hub balli, specializing in Computer Vision that helps visually impaired people to become independent by developing and deploying Deep Learning applications on Edge devices. Founded in 2021, Einetcorp is a privately held company. Our team of industry experts is dedicated to driving innovation, shaping the future, and delivering top-notch AI solutions that helps visually impaired. At Einetcorp, harnessing the power of Artificial Intelligence (AI), Machine Learning (ML), Deep Learning, Computer Vision, and Edge AI to provide cutting-edge solutions that drive enterprise-level transformations. Our comprehensive suite of technologies and expertise allows us to develop AI solutions tailored to the unique needs. The focus is on delivering results and equipping the clients with the tools and capabilities to leverage AI effectively. Einetcorp Pvt Ltd at KLE Techpark Incubation Center emphasis on interdisciplinary collaborations bringing together specialists from many sectors such as engineering, AI, embedded systems, etc. Furthermore, the center provides opportunities for students to get practical experience through internships and research projects. The center prepares students for today's competitive landscape by giving access to cutting-edge devices. Driven by a passion for leveraging deep technologies, continuing to push the boundaries of AI to unlock new possibilities. With a focus on innovation, reliability, Einetcorp is committed to empowering individuals with visual disabilities to harness the full potential of AI and enhance their quality of life. In summary, Einetcorp is an Embedded AI company focused on delivering innovative computer vision solutions for the visually impaired. With expertise in Deep Learning, Computer Vision, and Edge AI, the company is committed to enabling independence and accessibility through cutting-edge AI applications deployed on edge devices.

1.3 Project Background

The intern created unique pretreatment and postprocessing modules, set up GPIO interfaces for application mode switching, and integrated hardware and software. They also put in place a text-to-speech (TTS) engine for audio output. Together, these elements created an interactive embedded system with text reading and object detecting capabilities in real time. At the end of the internship, a functional prototype was deployed, confirming the system's efficacy in real-world situations and offering practical experience in developing embedded AI applications.

The project's main goal was to create and implement a real-time, low-power embedded system that could read printed text from its surroundings and recognize objects. Because of its Graph Streaming Processor (GSP) design, which provides high-performance inference at low latency and power consumption, the Blaize X600M AI accelerator served as the foundation for the system. As the host controller, the Raspberry Pi CM5 oversaw

user interface, peripheral connection, and system logic. When combined, they created a powerful platform that could run sophisticated vision algorithms appropriate for wearable and portable devices.

Pre-trained models like YOLOv8 and PaddleOCR were used to provide key AI functions including multi-class object identification and OCR. These models were then improved and made available through the Picasso SDK. In order to analyze video input, carry out inference, apply postprocessing, and provide real-time audio output using a TTS engine, the system also had a specially designed GStreamer pipeline. To ensure non-visual users could easily switch between program modes (such as object detection and text reading), GPIO buttons were included.

The project handled human-centric design concerns including usability, responsiveness, and interactivity without a visible interface in addition to the technical difficulties of implementing AI on embedded technology. The finished prototype made a significant contribution to the expanding field of inclusive and accessible technology by validating the efficacy of combining computer vision with embedded AI to help visually impaired people.

1.4 Problem Statement

To develop a real-time AI system for accurate visual recognition and audio feedback.

1.5 Project Objectives

The objectives for the problem statement are as follows:

- To research, evaluate, and compare various OCR frameworks that can be used for real-time applications.
- implement a real-time OCR pipeline capable of detecting, classification, and recognizing textual content from diverse sources.
- To integrate the OCR module seamlessly with the TTS system, enabling clear and immediate audio feedback of the recognized text.
- To ensure that the OCR system is accessible and easy to use, by enabling interaction through GPIO-based button controls instead of traditional GUI-based input, thereby accommodating users with visual disabilities.
- To optimize the entire pipeline for low-latency, energy-efficient execution on the Raspberry Pi CM5 with Blaize AI accelerator.

1.6 Organization of the Report

Chapter 1: Introduction This chapter presents the background and motivation for the project, emphasizing the importance of assistive technologies for visually impaired individuals. It also outlines the problem statement and clearly defines the objectives that guided the development of the proposed solution.

Chapter 2: Literature Survey This section provides a comprehensive review of existing works related to Optical Character Recognition (OCR), Text-to-Speech (TTS) systems, and assistive devices. It highlights the key technologies, methodologies, and limitations found in previous research, thereby justifying the need for the proposed system.

Chapter 3: Methodology This chapter explains the step-by-step approach followed to implement the system. It includes the selection and configuration of hardware, the integration of PaddleOCR and Coqui TTS, the design of the language selection mechanism using GPIO, and the development of the complete OCR-to-voice pipeline.

Chapter 4: Results and Discussions This section presents the outcomes obtained from testing the system under various conditions. It includes sample outputs, performance metrics, and discusses the efficiency, responsiveness, and accuracy of the text recognition and speech synthesis components.

Chapter 5: Conclusion and Future Scope The final chapter summarizes the overall work carried out in the project. It reflects on the impact and usefulness of the system for visually impaired users and provides insights into possible future enhancements, such as supporting more languages, adding gesture input, or improving processing speed.

Chapter 2

Literature Survey

Optical Character Recognition (OCR) and Text-to-Speech (TTS) systems must be effectively and closely integrated in order to develop assistive solutions for people with visual impairments. OCR is in charge of identifying and obtaining textual information from visual inputs like scanned papers, billboards, and live video streams. This text is then translated into understandable spoken language using TTS, allowing non-visual people to access printed information. The technological underpinnings of real-time audio-based accessibility solutions are formed by the combination of these systems. In commonplace settings like transit hubs, public signage, instructional materials, computer interfaces, and product packaging, they play a crucial role in facilitating inclusive experiences. The adoption of such technologies has grown significantly due to advances in deep learning, edge computing, and the increasing availability of lightweight models optimized for deployment on devices like the Raspberry Pi.

Convolutional Recurrent Neural Networks (CRNNs) are used for recognition, together with decoders that are based on either Connectionist Temporal Classification (CTC) or attention mechanisms. This enables it to effectively handle sequences of varying length. Out of all the OCR frameworks investigated by the research community and practical implementations, PaddleOCR is a particularly reliable, adaptable, and production-ready pipeline. PaddleOCR is an end-to-end OCR architecture that incorporates modules for text detection, orientation categorization, identification, and even layout analysis. It was created by Baidu and is based on the PaddlePaddle deep learning platform. With plug-and-play components that are easily customizable or interchangeable based on the needs of the application, the framework was created with flexibility in mind. With pre-trained models for more than 80 languages, PaddleOCR's multilingual support is a major benefit that makes it a very appealing option for international and multilingual use cases. Technically speaking, its detection pipeline makes use of Differentiable Binarization (DBNet), which locates text sections exactly using a segmentation-based method.

Additionally, PaddleOCR offers a number of low-power versions, like PP-OCR and PP-OCRv3, that are tailored for inference on devices with limited processing power. These models maintain comparable accuracy even when running on CPUs without the need for GPU acceleration. Du et al. showed in their 2020 study that PP-OCR can process data at real-time speeds on embedded devices while maintaining a high level of recognition performance. Because of its efficiency, PaddleOCR is one of the few frameworks that can provide industrial-grade text recognition in settings with strict power, memory, and latency constraints, which is essential for wearable or portable assistive technology.

Although PaddleOCR provides a full OCR stack, other frameworks offer more spe-

cialized or modular features. Zhou et al. (2017) presented EAST (Efficient and Accurate Scene Text Detector), one such model. By bypassing the intricacy of anchor boxes and proposal networks included in more traditional object detection techniques, EAST is a fully convolutional neural network that can predict the geometries of text bounding boxes in a single forward pass. It is intended for real-time applications and can identify text at any angle. EAST is now a commonly utilized standard for scene text identification, particularly in scholarly research on natural scene comprehension. However, it performs only the detection step and must be paired with a separate text recognition module — such as CRNN — to form a complete OCR system.

Keras-OCR is another popular OCR system that combines a CRNN-based model for recognition with CRAFT (Character Region Awareness for Text Detection) for identifying irregular text layouts. Keras-OCR, which is based on the TensorFlow and Keras frameworks, is praised for its community support, modular codebase, and ease of usage. Because it enables rapid setup and training of bespoke models, it is frequently chosen during the prototyping and testing stages of OCR system design. However, Keras-OCR is not the best option for deployment on edge devices because of its relatively high memory consumption and lack of hardware-aware optimizations. It is more useful for study, teaching, and situations where speed of inference is not a major consideration.

On the other hand, MMOCR is a sophisticated, research-focused OCR framework with a wide range of features that was created by the OpenMMLab team. In addition to recognition models like CRNN, SAR, and NRTR, it supports a wide range of detection models like DBNet, CRAFT, and PAN. Additionally, it offers tools for converting between different OCR datasets, assessing model performance, and displaying findings. Because of its great modularity and adaptability, MMOCR is perfect for scholarly research and extensive analyses of OCR algorithms. Nevertheless, there are substantial equipment requirements and a high learning curve associated with this versatility. MMOCR's computing requirements can be prohibitive for embedded or resource-constrained settings, such Raspberry Pi deployments, unless they are significantly reduced or quantized.

In the field of text-to-speech (TTS), significant advancements have been made in recent years in the use of deep learning to produce expressive and natural-sounding synthetic speech. Proposed by Kim et al. in 2021, the VITS (Variational Inference Text-to-Speech) model is one of the most notable innovations. In order to produce raw audio waveforms straight from text input, VITS is an end-to-end generative model that combines a variational autoencoder, adversarial learning, and normalizing processes. VITS can learn both the linguistic and acoustic aspects of speech in a single model, in contrast to previous models that depend on intermediate mel-spectrograms (such as Tacotron2 + WaveGlow or FastSpeech + Vocoder combinations). This enables generated speech to be more expressive, natural, and prosodic. While it takes a lot of resources to train VITS from scratch, once the model is trained and transferred to a format like ONNX, inference can be done quickly. This makes it possible to install real-time voice synthesis on edge devices without requiring constant internet connectivity.

For applications that need a simple and fast TTS solution, Google Text-to-Speech, or gTTS, provides a practical substitute. With support for numerous languages and dialects, it is a Python module that wraps around Google's cloud-based TTS API. Simple to set up, gTTS requires little local computation and can be used for basic accessibility use cases, language learning tools, and demonstrations. Nevertheless, gTTS is essentially an online solution; in order to work, it needs an active internet connection and provides no additional control over voice quality, speed, or pitch beyond the most basic options.

It is also incompatible with situations where offline or real-time performance is a severe necessity and inappropriate for privacy-sensitive contexts due to its reliance on cloud infrastructure.

In summary, accuracy, computational efficiency, model flexibility, and simplicity of integration must all be balanced when choosing OCR and TTS frameworks for embedded systems, especially those aimed at assistive technology. With its wide language coverage, real-time performance, and low resource usage, PaddleOCR proves to be a useful and effective OCR solution. Similarly, VITS offers a cutting-edge offline speech synthesis pipeline that can produce high-quality audio for real-time applications on gadgets like the Raspberry Pi after it has been taught. When combined, these resources provide a solid basis for creating intelligent, multilingual assistive systems that can improve accessibility and self-sufficiency for people with visual impairments.

Chapter 3

Methodology

A methodology is comprised of methods, techniques, processes, procedures, and rules. Methodologies used in project management are precise, rigid, and typically include a list of actions and tasks at each stage of the project's life cycle. They are defined techniques that outline the specific next steps to take, the reasoning behind each one, and the proper way to carry out each project stage. Different techniques or strategies are discussed in this chapter. The method's justification is explained. The many methods that can be applied throughout the implementation process are discussed in this section.

3.1 OCR Pipeline

Developed by Baidu as a component of the PaddlePaddle deep learning platform, PaddleOCR is an end-to-end open-source OCR solution. It uses cutting-edge deep learning algorithms to effectively detect and identify text in photos. PaddleOCR's excellent accuracy, flexible design, and deployment optimization for embedded systems and edge devices like Raspberry Pi make it very potent. PaddleOCR is the foundation of our research and is used to extract useful text data from real-world camera-captured photos. For visually handicapped users who depend on the system to translate visual input into spoken output, this feature is essential.

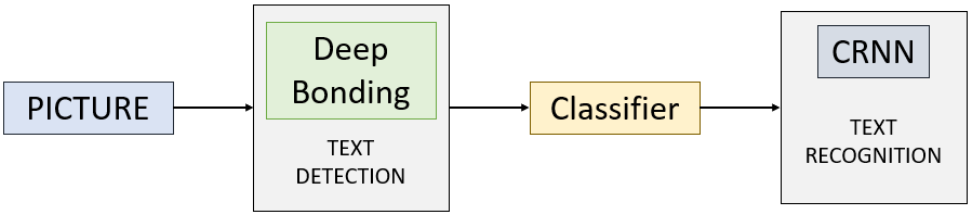

Figure 3.1: OCR Model Block Diagram

3.1.1 Text Detection

The purpose of the text detection stage is to locate text areas inside an image. Although PaddleOCR offers a number of detection methods, we decided to use DBNet (Differen-

tiable Binarization Network) for our system since it strikes a compromise between accuracy and speed.

Working Principle:

- Image Preprocessing: The aspect ratio of the supplied image is preserved when it is scaled. Both padding and pixel normalization are used.
- Backbone Network: A CNN backbone such as ResNet-50 extracts hierarchical image features.
- Segmentation Prediction: A segmentation map is produced, highlighting regions with high text probability.
- Binarization: The segmentation map is sharpened and better text outlines are produced by applying differentiable binarization.
- Box Formation: Bounding boxes are drawn around identified text areas once the contours are taken from the binary map. This process mainly yields highly accurate bounding boxes, even for curved and rotated text. It performs well on both printed and stylized fonts.

3.1.2 Text Recognition

The next step is to identify the information within each text region when it has been identified. Convolutional Recurrent Neural Networks, or CRNNs, are the model used by PaddleOCR.

CRNN Workflow:

- Image Cropping and Normalization: Detected text regions are cropped and resized into a uniform shape.
- CNN Feature Extraction: A CNN extracts spatial features from each cropped image.
- Sequence Modeling with RNN: Character context and sequence are recorded by passing features through a BiLSTM (Bidirectional Long Short-Term Memory) network.
- Character Prediction: A CTC (Connectionist Temporal Classification) decoder receives the final output and, without character-level alignment, outputs the most likely character sequence.

3.2 Text to Speach

Our method use the VITTS (Voice Transformer Text-to-Speech) model to convert the identified text into speech that sounds human. A transformer-based TTS engine, VITTS provides the ideal balance of efficiency, naturalness, and performance.

3.2.1 VITTS Architecture

VITTS uses a modular architecture composed of three main components that work together to convert text into natural-sounding speech. The Text Front-End is responsible for converting raw input text into a sequence of phonemes or linguistic tokens, ensuring accurate pronunciation and consistent linguistic representation. These tokens are then passed to a Transformer Decoder, which maps them to mel-spectrograms using attention mechanisms and positional encoding to capture the temporal and contextual relationships between speech elements. Finally, a HiFi-GAN Vocoder processes the spectrograms and generates the corresponding audio waveforms that can be played back as speech. This separation between spectrogram generation and waveform synthesis allows for greater modularity and flexibility, enabling faster training, more efficient inference, and easier integration of improved components as they become available.

3.2.2 Model Working

In our system, VITTS is used to convert recognized text into speech through a four-stage pipeline designed for real-time performance. The process begins with text normalization, where the input text is cleaned and converted into a phoneme sequence to ensure correct pronunciation. This sequence is then fed into a Transformer-based decoder, which generates high-resolution mel-spectrograms that represent the acoustic features of the speech. These spectrograms are passed to the HiFi-GAN vocoder, which synthesizes a 16-bit audio waveform from the spectrogram data. Finally, the generated waveform is delivered to the user through real-time playback via the system's integrated audio output module. This structured flow ensures that the text-to-speech conversion is smooth, natural-sounding, and operates with minimal latency, making it well-suited for assistive real-time applications.

3.3 System Integration

To ensure seamless operation and user-friendly control, the system was designed to integrate all core functionalities—OCR, object detection, and system control—using physical push buttons connected to the Raspberry Pi's GPIO pins. Each button acts as a trigger to initiate a specific sub-process, allowing mode switching without the need for a touchscreen or keyboard. The system was also optimized for real-time performance by employing multithreading, enabling independent execution of camera input capture, inference, and audio output tasks. Additionally, a structured audio routing mechanism ensures that output is correctly delivered based on the current operational mode.

3.3.1 GPIO/Trigger Input Handling

To enable seamless user interaction without a graphical interface, three physical buttons were interfaced with the Raspberry Pi CM5 using GPIO pins. The pins used were: Pin 3 (GPIO 2) for power control, Pin 5 (GPIO 3) for object detection, and Pin 7 (GPIO 4) for OCR activation. A common ground connection was established using Pin 9. Each button is mapped to a specific callback function in the control script, which listens for falling edge signals and triggers the corresponding action. Pressing the power button initializes or shuts down the system, while the other two buttons toggle their respective

processes—object detection or OCR—on or off. The use of pull-up resistors ensures reliable button input detection. This hardware-based trigger system provides an intuitive and accessible interface for users to switch between modes or terminate the application without relying on a touchscreen or external input device.

3.3.2 Optimization : Frame Skipping & MultiThreading

To ensure real-time responsiveness and reduce processing latency, a frame skipping mechanism was implemented as part of the pipeline. Instead of feeding every captured frame to the OCR or object detection models, only 1 out of every 10 frames was selected for inference. This significantly reduced the computational load while still maintaining sufficient visual context for recognition tasks. Since both OCR and object detection are resource-intensive, this approach prevented system bottlenecks and allowed the pipeline to run smoothly on the embedded platform. Skipping redundant frames helped conserve processing time, memory, and power—crucial factors in edge AI deployments.

In addition to frame reduction, multi-threading played a critical role in achieving real-time system behavior. Independent threads were created for handling camera input, running OCR inference, and managing TTS output. This allowed different components of the pipeline to execute concurrently, without blocking each other. For example, while one thread was busy performing inference on a frame, another could handle audio playback or prepare the next image. This parallelism improved system responsiveness and made the experience smoother for the end user. Overall, the combination of frame skipping and multi-threaded execution was essential to meeting the real-time constraints of the project while preserving accuracy and usability.

3.3.3 Audio Routing

In the proposed system, audio output is generated using a VITTS-based Text-to-Speech (TTS) engine, which provides clear, human-like speech feedback for the visually impaired. Since both OCR and TTS run on separate threads, a queue-based communication mechanism was used to pass recognized text from the OCR thread to the TTS thread. This decoupled design ensured that text recognition and speech synthesis could operate asynchronously, enhancing overall system responsiveness. To avoid redundant or repetitive audio playback, a similarity check was implemented: if the current OCR output text shares more than 80% similarity with the previously spoken sentence, it is filtered out and not sent to the TTS engine. This reduced unnecessary speech output and improved the quality of user interaction. The final, filtered text is sent to the VITTS model, which synthesizes the corresponding voice output and plays it through the system speaker in real time.

3.4 Deployment on RPi CM5

For seamless usability and a plug-and-play experience, the entire system was configured to launch automatically at boot on the Raspberry Pi Compute Module 5 (CM5). A custom Bash script was written to initiate the main Python control program, and a corresponding systemd service was created to register this script as part of the boot process. As a result, the moment the Raspberry Pi is powered on, the system initializes the GPIO interface, plays a welcome audio message, and begins listening for user input via the

physical buttons. This approach eliminates the need for any manual startup commands or graphical interface, allowing the assistive system to be operational immediately after powering up. Such deployment makes the device user-friendly, especially for non-technical users, and ensures it is always ready to perform OCR or object detection tasks upon request.

Chapter 4

Results and Discussions

Using the Blaize Xplorer X600M AI accelerator, this part evaluates the integration and performance of the object detection, OCR, and TTS pipelines built on the Raspberry Pi CM5. The accuracy, latency, responsiveness, and real-time application of each module were evaluated both separately and as a component of the entire system.

YOLOv8 models that were improved with NetDeploy were used to assess the object detection module. The system demonstrated high confidence scores in identifying common things in well-lit environments, including people, cars, and bags. Low-latency inference was made possible by the Blaize accelerator, which allowed for real-time processing speeds. Under moderate motion, detection performance held steady, while including a Kalman SORT tracker enhanced object continuity and decreased output jitter.

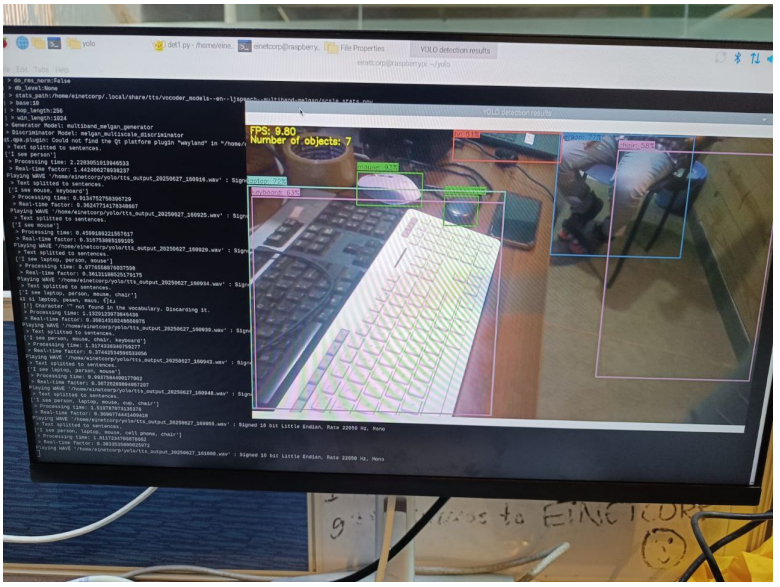

Figure 4.1: Result of Object Detection task

The TTS module produced natural-sounding speech with little delay when it was built using Glow-TTS with HiFi-GAN as the vocoder. Between 1.2 and 1.5 seconds passed between text input (from OCR or detection) and audio output. The object names and text strings were pronounced clearly, and the voice output was comprehensible and well-paced.

PaddleOCR models, which were made available via Blaize's GStreamer pipeline, were used to implement OCR. Even with noisy backdrops or uneven clarity, the pipeline was able to detect and classify printed text from photos and live video feeds with high accuracy. It was appropriate for real-time use in assistance applications because its detection and identification delay was between 1 and 1.5 seconds for normal inputs.

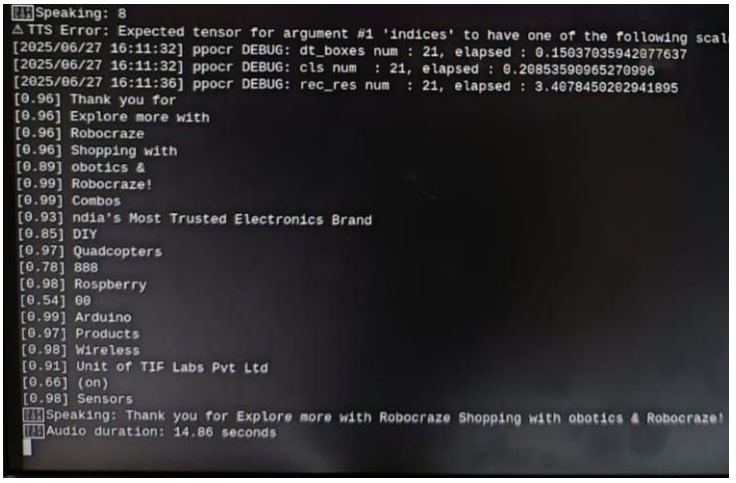

Figure 4.2: Result of Optical Character Recognition task

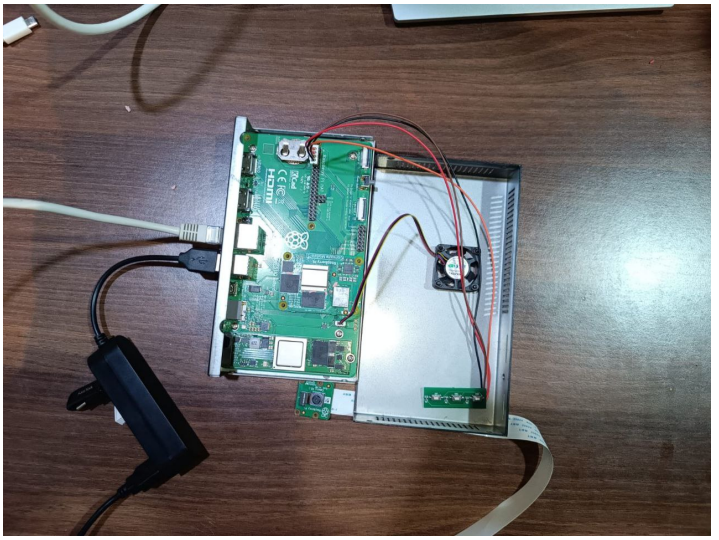

Figure 4.3: Integrated System Testing

A number of real-world use cases, including detecting household objects and vocalizing printed labels, were used to evaluate the entire pipeline, which combines object identification, OCR, and TTS. With seamless transitions between speech output and display input, the integration went well. Even with constant use, the system's responsiveness held steady, and GPIO-based controls let users switch between modes (such as detection and OCR) and activate voice feedback as required. The system's versatility and efficacy

Figure 4.4: Final Prototype

as a visual aid for visually impaired users were demonstrated by its performance in both indoor and outdoor environments.

To assess the software pipeline's performance in real time, a specialized testing environment was created. The Blaize Xplorer X600M accelerator, a USB camera for input, an external audio output device, and a Raspberry Pi CM5 development board were all part of the hardware setup. Performance evaluation, stress testing, and intensive debugging were made possible by this setup. During the testing phase, latency, model performance, and GPIO interface may all be adjusted.

The completed prototype was intended to be a small, integrated solution that combined all of the essential modules—OTS, OCR, and object detection—into a single deployable device. To replicate real-world use, the prototype was powered by a portable source and housed in a lightweight case. It was tested in live settings, at different distances, and under various lighting conditions. Its utility as a real-time assistive vision device that can recognize text and objects and provide instant audio feedback was proven by its consistent performance.

Chapter 5

Conclusion and Future Scope

5.1 Conclusion

The Smart Vision for Visually Impaired project effectively illustrates how cutting-edge AI technology can be used to help people with visual impairments better comprehend their environment. The system can precisely identify and recognize text in real time by utilizing PaddleOCR to construct an effective OCR pipeline. Additionally, the identified material is transformed into natural-sounding speech by including an optimized TTS system with Glow-TTS models, allowing for smooth aural feedback.

By detecting environmental items, the object detection component—powered by a lightweight YOLOv11 model tailored for edge devices—further enhances the system and gives users vital spatial awareness. The Raspberry Pi CM5 effectively deploys the bundled functionalities, guaranteeing that the system functions offline, making it both cost-effective and practical for real-world use, especially in rural or low-connectivity areas.

5.2 Future Scope

The 8-channel analog input module can be further developed to serve higher-level applications such as industrial automation, automotive sensor networks, and environmental monitoring systems. By enhancing hardware precision and scalability, it can support complex sensor arrays for real-time data acquisition in smart factories or autonomous vehicles. Integration with robust protection circuits and improved power management will make it suitable for harsh and safety-critical environments. Additionally, miniaturization and modular design will enable seamless embedding into advanced control systems, enabling accurate multi-sensor monitoring and improving decision-making in IoT and Industry 4.0 applications.

Bibliography

1. Y. Du, X. Pan, H. Xu, et al., "PP-OCR: A Practical Ultra Lightweight OCR System," arXiv preprint arXiv:2009.09941, 2020.

2.X. Zhou, C. Yao, H. Wen, et al., "EAST: An Efficient and Accurate Scene Text Detector," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017, pp. 5551–5560.

3. F. Morales, "Keras-OCR," GitHub Repository, 2020..

4. MMOCR Contributors, "MMOCR: OpenMMLab Text Detection, Recognition and Understanding Toolbox," OpenMMLab, 2021.

5. J. Kim, S. Kim, J. Kong, and J. Yoon, "Conditional Variational Autoencoder with Adversarial Learning for End-to-End Text-to-Speech," in Proceedings of the 38th International Conference on Machine Learning (ICML), 2021. .

6. gTTS Developers, "gTTS: Google Text-to-Speech Python Library," PyPI, 2023..

rep1

ORIGINALITY REPORT

12%	9%	7%	2%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

www.researchgate.net Internet Source	2%
www.huggingface.co Internet Source	2%
www.colab.ws Internet Source	2%
www.raspberrypi.com/documentation Internet Source	1%
github.com Internet Source	1%
bcmi.sjtu.edu.cn Internet Source	1%
vlib.itrc.as.ir Internet Source	1%