# SCAN-N-LEARN : SMART ASSISTIVE TOOL FOR DYSLEXIC KIDS

### A PROJECT REPORT

***Submitted by***

**AMRITHA KRISHNAKUMAR JOKI(910622108003)**

**CHARU NIVEDIDA A.N (910622108006)**

**SANTHNI M.D(910622108048)**

***in partial fulfillment for the award of the degree of***

## BACHELOR OF ENGINEERING

***in***

**ARTIFICIAL INTELLIGENCE AND DATA SCIENCE**



## K.L.N. COLLEGE OF ENGINEERING,POTTAPALAYAM

**(An Autonomous Institution, Affiliated to Anna University, Chennai)**

## NOVEMBER 2025

## K.L.N. COLLEGE OF ENGINEERING,POTTAPALAYAM

**(An Autonomous Institution, Affiliated to Anna University, Chennai)**

## BONAFIDE CERTIFICATE

Certified that this project report **“ SCAN-N-LEARN : SMART ASSISTIVE TOOL FOR DYSLEXIC KIDS”** is the bonafide work of “**Ms. AMRITHA KRISHNAKUMAR JOKI(910622108003) , Ms.CHARU NIVEDIDA A.N (910622108006) ,Ms. SANTHNI M.D(910622108048)”** who carried out the Project Work jointly under our supervision

**SIGNATURE SIGNATURE**

**Dr. SURESH RAJA S, Dr. BALAMURUGAN S,**

**M.C.A,M.PHIL.(COMP.SCI). B.E.(EEE),M.TECH(C&I).,**

**M.E.(CSE),PH.D., PH.D.,**

**HEAD OF THE DEPARTMENT ASSISTANT PROFESSOR(SG)**

**ARTIFICIAL INTELLIGENCE AND ARTIFICIAL INTELLIGENCE**

**DATA SCIENCE AND DATA SCIENCE**

**K.N.L.COLLEGE OF ENGINEERING K.L.N.COLLEGE OF ENGINEERING**

(An ISO 9001-2008 Certified Institution) (An ISO 9001-2008 Certified Institution)

**POTTAPALAYAM, SIVAGANGAI, POTTAPALAYAM,SIVAGANGAI,**

**TAMIL NADU, INDIA. TAMIL NADU,INDIA.**

Submitted for the project work viva-voce examination held on

**INTERNAL EXAMINER EXTERNAL EXAMINER**

### ACKNOWLEDGEMENT

Any work would be unfulfilled without a word of thanks. We hereby take pleasure in acknowledging the persons who guided me throughout our work.

First and foremost, thanks are to the omnipotent for providing us with his abundant blessings all throughout. We all extend our heartfelt thanks to **Er.K.N.K.KARTHIK, B.E.,** President of our college and **Dr.A.V.RAMPRASAD, M.E., Ph.D.,** Principal for provisioning us with the all required.

We esteem our self to articulate our sincere thanks to **Dr. S. SURESH RAJA, M.C.A.,M.PHIL.(COMP.SCI).,M.E(CSE).,PHD** Head of the Artificial Intelligence and Data Science for leading us towards the zenith of success.

We express our grateful thanks to our **Project Guide Dr. S. BALA MURUGAN B.E(EEE),M.TECH(C&I),PH.D.,** and **Project Coordinator Dr. S. BALA MURUGAN B.E(EEE),M.TECH(C&I),PH.D.,**for their invaluable guidance and motivation. Their assistance and advices had been very helpful throughout our project. I would like to thank all teaching and non-teaching staffs of our department who have been the sources of encouragement and ideas. I thank them for lending their support whenever needed.

**ABSTRACT**

Scan n Learn: A Smart Assistive Tool for Dyslexic Kids is a multilingual reading companion designed to help children with dyslexia access and understand printed text more effectively. Dyslexia is a neurodevelopmental disorder that affects reading accuracy, spelling, and decoding, often leading to reduced learning confidence and slower academic progress. To overcome these barriers, the proposed system provides an auditory reading experience by converting printed text into natural speech through a structured OCR-to-TTS (Optical Character Recognition to Text-to-Speech) pipeline.

The system employs a Raspberry Pi 4 Model B integrated with a camera module to capture printed material in real time. Captured images undergo preprocessing, including grayscale conversion and contrast enhancement, to improve OCR accuracy. A script-aware OCR engine using Tesseract models recognizes multiple languages such as English, Hindi, Telugu, and Malayalam. The extracted text is then processed by a fast and efficient Text-to-Speech engine, producing clear, phonetically accurate audio output. Additional text normalization ensures correction of misread characters before playback.

By transforming visual text into intelligible speech, Scan n Learn reduces cognitive effort and enhances comprehension for dyslexic learners. The system promotes independent reading, supports inclusive education, and offers scalability for future integration of additional languages and preprocessing features, making it a practical and innovative solution in assistive learning technology.

**TABLE OF CONTENTS**

| **CHAPTER NO** | **TITLE** | **PAGE NO** |
| --- | --- | --- |
|  | **ABSTRACT** | iv |
|  | **TABLE OF CONTENTS** | i |
|  | **LIST OF TABLES** | v |
|  | **LIST OF FIGURES** | vi |
|  | **LIST OF ABBREVIATIONS AND NOMENCLATURE** | vii |
| **1** | **INTRODUCTION** | 1 |
|  | **1.1 SIGNIFICANCE OF EARLY INTERVENTION IN DYSLEXIA** | 1 |
|  | **1.2 CHALLENGES FACED BY DYSLEXIC LEARNERS** | 2 |
|  | 1.2.1 Reading difficulty | 2 |
|  | 1.2.2 Limited accessibility | 2 |
|  | 1.2.3 Psychological impact | 2 |
|  | 1.2.4 Language diversity | 2 |
|  | 1.2.5 Dependence on human assistance | 2 |
|  | **1.3 OBJECTIVES OF THE SMART ASSISTIVE TOOL FOR DYSLEXIC KIDS** | 3 |
|  | 1.3.1 Development of an image-based text extraction system | 3 |
|  | 1.3.2 Implementation of multilingual text recognition | 3 |
|  | 1.3.3 Integration of high-quality text-to-speech synthesis | 3 |
|  | 1.3.4 Design of a portable and standalone device | 3 |
|  | 1.3.5 Development of a user-friendly interface | 4 |
|  | 1.3.6 Evaluation of system performance and accuracy | 4 |
|  | **1.4 COMPARATIVE ANALYSIS OF CURRENT READING AIDS** | 4 |
|  | 1.4.1 JAWS screen readers | 4 |
|  | 1.4.2 C-Pen reader | 4 |
|  | 1.4.3 Limitations of C-Pen reader & JAWS application | 5 |
|  | **1.5 PROPOSED COGNITIVE READING MODEL** | 5 |
|  | 1.5.1 Raspberry Pi–based assistive hardware module | 5 |
|  | 1.5.2 Multilingual vision-to-speech software engine | 5 |
|  | 1.5.3 Dyslexia-friendly reading workflow | 6 |
|  | **1.6 INCLUSIVE LEARNING FEATURES AND OPERATIONAL HIGHLIGHTS** | 6 |
|  | **1.7 CONCEPTUAL FRAMEWOK OF THE SMART ASSISTIVE READING SYSTEM** | 6 |
|  | 1.7.1 Sensory-integrated learning approach | 6 |
|  | 1.7.2 User-centered design considerations | 7 |
|  | 1.7.3 Community supported ecosystem | 7 |
|  | 1.7.4 Multilingual inclusivity | 7 |
|  | **1.8 EDUCATIONAL AND SOCIAL IMPACT ON CHILDREN** | 7 |
|  | 1.8.1 Educational inclusion | 7 |
|  | 1.8.2 Psychological and emotional benefits | 8 |
|  | 1.8.3 Diverse language accessibility | 8 |
|  | 1.8.4 Economic accessibility | 8 |
|  | 1.8.5 Parental and teacher support | 8 |
|  | 1.8.6 Source scalability and adaptability | 9 |
|  | 1.8.7 Societal impact | 9 |
| **2** | **CURRENT TECHNOLOGICAL LANDSCAPES FOR DYSLEXIA SUPPORT** | 10 |
|  | **2.1 REVIEW OF RELATED LITERATURE AND TECHNOLOGIES** | 10 |
|  | **2.2 DYSLEXIA AND LEARNING CHALLENGES** | 10 |
|  | **2.3 ASSISTIVE TECHNOLOGY FOR DYSLEXIA** | 11 |
|  | 2.3.1 Text-to-speech technology | 11 |
|  | 2.3.2 Mobile applications and digital solutions | 12 |
|  | 2.3.3 Dedicated hardware devices | 12 |
|  | **2.4 OPTICAL CHARACTER RECOGNITION TECHNOLOGY** | 13 |
|  | 2.4.1 Tesseract OCR engine | 13 |
|  | 2.4.2 Preprocessing and optimization | 14 |
|  | **2.5 TEXT-TO-SPEECH SYNTHESIS SYSTEMS** | 15 |
|  | 2.5.1 Neural network-based TTS | 15 |
|  | 2.5.2 Piper TTS framework | 15 |
|  | **2.6 RASPBERRY PI AS AN EDUCATIONAL PLATFORM** | 16 |
|  | 2.6.1 Hardware capabilities | 16 |
|  | 2.6.2 Educational adoption and accessibility | 16 |
|  | 2.6.3 Multilingual support in assistive technology | 17 |
|  | **2.7 GAPS IN EXISTING RESEARCH AND SOLUTIONS** | 18 |
|  | 2.7.1 Offline functionality | 18 |
|  | 2.7.2 Cost accessibility | 18 |
|  | 2.7.3 Language support | 18 |
|  | 2.7.4 Educational context | 19 |
|  | 2.7.5 Age-appropriate design | 19 |
|  | **2.8 RESEARCH INSIGHTS AND IDENTIFIED GAPS** | 19 |
| **3** | **RASPBERRY PI 4 MODULE** | 20 |
|  | **3.1 PROCESSING LOGIC AND SYSTEM FUNCTIONALITY OVERVIEW** | 20 |
|  | **3.2 CIRCUIT CONNECTION LAYOUT** | 21 |
|  | **3.3 FUNCTIONAL BLOCK REPRESENTATION** | 22 |
|  | **3.4 HARDWARE REQUIREMENTS** | 23 |
|  | 3.4.1 Raspberry Pi 4 Model B (4GB RAM) | 23 |
|  | 3.4.2 Raspberry Pi camera rev 1.3 | 24 |
|  | 3.4.3 Micro SD card (32GB) | 25 |
|  | 3.4.4 Power supply (5V, 3A) | 26 |
|  | 3.4.5 3.5mm wired earphones | 26 |
|  | 3.4.6 Push button | 27 |
|  | **3.5 SOFTWARE REQUIREMENTS** | 28 |
|  | 3.5.1 Operating system: Raspberry Pi OS – Bullseye version | 28 |
|  | 3.5.2 Programming language: Python | 28 |
|  | 3.5.3 Optical character recognition: Tesseract OCR | 29 |
|  | 3.5.4 Text-to-speech engine: Piper TTS | 29 |
|  | 3.5.5 Supporting libraries and dependencies | 29 |
| **4** | **DESIGN & FUNCTIONAL FLOW OF THE DYSLEXIA READING ASSISTANCE SYSTEM** | 31 |
|  | **4.1 ALGORITHM** | 31 |
|  | 4.1.1 Multi-stage OCR-to-TTS pipeline algorithm | 31 |
|  | **4.2 PSEUDO CODE** | 33 |
|  | **4.3 CODE** | 34 |
|  | **4.4 SEQUENTIAL WORKFLOW** | 40 |
| **5** | **SCREENSHOTS** | 41 |
| **6** | **CONCLUSION** | 43 |
|  | **6.1 FINAL EVALUATION AND PEDAGOGICAL IMPLICATIONS** | 43 |
|  | **6.2 FUTURE ENHANCEMENTS** | 44 |
|  | 6.2.1 Advanced machine learning & AI integration | 44 |
|  | 6.2.2 Expanded language support and script recognition | 45 |
|  | 6.2.3 Enhanced user interface and accessibility features | 45 |
|  | 6.2.4 Personalization and adaptive learning | 45 |
|  | 6.2.5 Hardware improvements and alternative form factors | 46 |
| **7** | **REFERENCE** | 47 |

**LIST OF TABLES**

| **TABLE NO.** | **TITLE** | **PAGE NO.** |
| --- | --- | --- |
| 3.1 | POWER SUPPLY REQUIREMENTS FOR RASPBERRY PI 4 MODEL B | 22 |
| 3.2 | RASPBERRY PI 4 MODEL B SPECIFICATIONS | 23 |

**LIST OF FIGURES**

| **FIGURE NO.** | **TITLE** | **PAGE NO.** |
| --- | --- | --- |
| 3.1 | Circuit Diagram of Raspberry Pi 4 Model B | 21 |
| 3.2 | Block Diagram of Raspberry Pi | 22 |
| 3.3 | Raspberry Pi 4 Model B | 23 |
| 3.4 | Pi Camera Rev 1.3 | 24 |
| 3.5 | Micro SD Card | 25 |
| 3.7 | Power Supply | 26 |
| 3.8 | Wired Earphones | 26 |
| 3.9 | Push Button | 27 |
| 4.1 | Sequential Workflow | 40 |
| 5.1 | English Audio Response | 41 |
| 5.2 | Hindi Audio Response | 41 |
| 5.3 | Malayalam Audio Response | 42 |
| 5.4 | Telugu Audio Response | 42 |

**LIST OF ABBREVATIONS & NOMENCLATURE**

| Abbreviation | Full Form | Description |
| --- | --- | --- |
| OCR | Optical Character Recognition | Technology that converts printed or handwritten text into machine-readable text. |
| TTS | Text-to-Speech | System that converts written text into spoken audio output. |
| Pi | Raspberry Pi | A small, affordable single-board computer used for embedded applications. |
| GPIO | General Purpose Input/Output | Pins on the Raspberry Pi used to interface with external hardware like buttons or sensors. |
| ONNX | Open Neural Network Exchange | Open format for representing machine learning models for interoperability across platforms. |
| LSTM | Long Short-Term Memory | A type of recurrent neural network (RNN) used in sequence modeling, like OCR text recognition. |
| CPU | Central Processing Unit | The main processor of a computer or embedded system that executes instructions. |
| RAM | Random Access Memory | Memory used by the processor to store and quickly access running programs and data. |
| BGR | Blue Green Red | Color format used in OpenCV for images, representing the three color channels. |
| cv2 | OpenCV Library | Python library used for computer vision tasks, including image processing and analysis. |
| PIL | Python Imaging Library (Pillow) | Python library for image manipulation and processing tasks. |
| SD | Secure Digital | Storage medium used to store OS, project code, and data for Raspberry Pi. |
| VITS | Variational Inference with adversarial learning for Text-to-Speech | Neural TTS architecture for generating natural-sounding speech. |
| Hz | Hertz | Unit of frequency, used to describe sampling rate in audio processing. |
| WAV | Waveform Audio File Format | Standard digital audio file format used for storing waveform data. |
| PyAudio | Python Audio Library | Python library for audio playback and recording. |
| sd | sounddevice Library | Python library used for real-time audio playback and recording. |
| CLI | Command Line Interface | Text-based interface to interact with the operating system or software. |
| TTS Engine | Text-to-Speech Engine | Software component that synthesizes audio from text. |
| CSI | Camera Serial Interface | Dedicated port on Raspberry Pi for connecting camera modules. |

**CHAPTER 1**

**INTRODUCTION**

## 1.1 SIGNIFICANCE OF EARLY INTERVENTION IN DYSLEXIA

## Dyslexia is a neurological learning disorder affecting 5–10% of the global population, characterized by persistent difficulties in accurate and fluent word recognition, spelling, and decoding. In educational and professional environments, individuals with dyslexia often face barriers that hinder their academic progress and career growth, despite having normal intelligence and sufficient instruction. This challenge underscores an increasing need in the education technology (EdTech) industry for accessible, adaptive, and inclusive learning solutions that can support diverse learner profiles.

## The global demand for assistive technologies tailored to learning disabilities has surged, driven by the push for inclusive education policies and advancements in artificial intelligence, speech technology, and embedded systems. Industry trends indicate a growing market for intelligent reading aids, text-to-speech devices, and personalized learning systems that can cater to learners with reading disorders like dyslexia.

## In response to this need, the Smart Assistive Tool for Dyslexia Kids project aims to develop an affordable and portable solution that bridges the accessibility gap. The device integrates Optical Character Recognition , Text-to-Speech synthesis, and machine learning on a Raspberry Pi 4 Model B platform to convert printed or digital text into natural-sounding speech in real time. This innovation not only empowers dyslexic learners to access written content independently but also demonstrates how AI-driven assistive technologies can contribute to a more inclusive educational ecosystem.

## By addressing a critical gap in the EdTech industry, this project aligns with the broader vision of developing intelligent, user-centric solutions that enhance literacy and learning outcomes for children with dyslexia

## 1.2 CHALLENGES FACED BY DYSLEXIC LEARNER

Children with dyslexia encounter substantial barriers in conventional educational environments where text-based learning predominates. The primary challenges include:

**1.2.1 Reading Difficulty**

Dyslexic children experience persistent difficulties in decoding written words, resulting in slow reading speeds and frequent errors that impede comprehension.

**1.2.2 Limited Accessibility**

Traditional learning materials are predominantly text-based, with insufficient multimodal alternatives available for children with reading disabilities.

**1.2.3 Psychological Impact**

Repeated reading failures contribute to anxiety, frustration, and avoidance behaviors, negatively affecting academic motivation and self-esteem.

**1.2.4 Language Diversity**

Existing assistive tools predominantly support English, leaving non-English speaking dyslexic children with limited resources, particularly in multilingual regions of India.

**1.2.5 Dependence on human assistance**

Constant reliance on parents, teachers, or peers for reading support limits independent learning opportunities and may not always be feasible.

There exists a critical need for an affordable, portable, and multilingual assistive technology solution that enables dyslexic children to access written content independently, thereby fostering academic inclusion and confidence.

## 1.3 OBJECTIVES OF THE SMART ASSISTIVE TOOL FOR DYSLEXIC KIDS

The primary objective of this project is to design and develop a portable smart assistive reading tool that enables dyslexic children to independently access and comprehend written text through auditory feedback.

**1.3.1 Development of an Image-Based Text Extraction System**

To develop an image-based text extraction system using Optical Character Recognition (OCR) technology capable of processing printed text from physical sources.  
This enables the system to accurately identify and convert printed characters into machine-readable text, forming the foundation for text-to-speech conversion.

### 1.3.2 Implementation of Multilingual Text Recognition

To implement multilingual text recognition supporting English, Hindi, Malayalam, and Telugu languages, addressing the linguistic diversity of Indian users.  
 This ensures accessibility for children from various linguistic backgrounds and promotes inclusivity in education across different regions of India.

### 1.3.3 Integration of High-Quality Text-to-Speech Synthesis

To integrate high-quality text-to-speech synthesis that converts recognized text into natural-sounding speech in the corresponding source language.  
 By producing human-like voice output, the system provides a realistic and engaging auditory learning experience that enhances reading comprehension.

**1.3.4 Design of a Portable and Standalone Device**

To create a portable, standalone device using cost-effective hardware components that can operate without continuous internet connectivity.  
 This design supports offline use, making the system reliable in rural or low-connectivity areas and practical for classroom environments.

### 1.3.5 Development of a User-Friendly Interface

To provide an intuitive user interface that minimizes complexity and enables children to operate the device with minimal training.  
 A simple and interactive interface ensures that even young dyslexic learners can use the tool independently without external assistance.

**1.3.6 Evaluation of System Performance and Accuracy**

To evaluate the system's accuracy in text recognition and speech synthesis quality across multiple languages and document types.  
 This evaluation validates the system’s effectiveness, ensuring high reliability, linguistic accuracy, and smooth speech output under various conditions.

**1.4 COMPARATIVE ANALYSIS OF CURRENT READING AIDS**

Current assistive solutions for dyslexic individuals can be broadly categorized into software applications, dedicated electronic devices, and human-assisted interventions:

**1.4.1 JAWS Screen Readers**:

Commercial screen readers such as JAWS (Job Access With Speech) and NVDA (NonVisual Desktop Access) are primarily designed for visually impaired users and require pre-digitized text. Mobile applications like Microsoft Immersive Reader and Learning Ally offer read-aloud functionality but necessitate smartphone operation and often require internet connectivity, which may not be consistently available or appropriate for young children.

**1.4.2 C-Pen Reader**

Specialized reading pens such as C-Pen Reader and ReaderPen scan and read text aloud but are limited to line-by-line reading, making them impractical for longer passages. These devices are also relatively expensive, with costs ranging from $200 to $400, placing them beyond reach for many families.

**1.4.3 Limitations of C-Pen Reader & JAWS Application**

Current solutions exhibit several shortcomings including high cost, dependence on internet connectivity, complexity of operation, limited multilingual support (especially for Indian languages), requirement for pre-processed digital content, and inadequate portability. Furthermore, most existing tools are not specifically optimized for the cognitive and ergonomic needs of children with dyslexia. The internet dependency of most modern assistive applications also renders them unsuitable for school environments where internet access may be restricted or prohibited for students, limiting their practical utility in educational settings.

**1.5 PROPOSED COGNITIVE READING MODEL**The proposed model addresses the limitations of existing systems through an integrated hardware-software architecture:

**1.5.1 Raspberry Pi–Based Assistive Hardware Module**

The system employs a Raspberry Pi 4 Model B (4GB RAM) as the central processing unit, providing sufficient computational power for real-time OCR and TTS operations. A Raspberry Pi Camera Module v2 captures high-resolution images of text documents. Audio output is delivered through standard 3.5mm wired earphones, ensuring privacy and focused listening.

**1.5.2 Multilingual Vision-to-Speech Software Engine**

The system operates on Raspberry Pi OS (Linux-based), ensuring stability and open-source flexibility. Text extraction is performed using Tesseract OCR, an open-source optical character recognition engine capable of recognizing over 100 languages with high accuracy. Speech synthesis is implemented using the Piper TTS model, a lightweight, neural network-based text-to-speech engine that generates natural-sounding speech without requiring cloud connectivity.

**1.5.3 Dyslexia-Friendly Reading Workflow**

Upon activation, the user positions a text document within the camera's field of view. The system captures the image, processes it through Tesseract OCR to extract text while automatically detecting the language (English, Hindi, Malayalam, or Telugu). The recognized text is then passed to the Piper TTS engine, which synthesizes speech in the detected language. The audio output is delivered through earphones in real-time, allowing the child to hear the content without manual reading.

**1.6 INCLUSIVE LEARNING FEATURES AND OPERATIONAL HIGHLIGHTS**

The proposed system offers multilingual support for four major Indian languages, complete offline operation without internet dependency making it school-friendly and suitable for classroom use, portability through compact form factor, low-cost implementation using affordable open-source components, user-friendly operation requiring minimal technical knowledge, and real-time processing with minimal latency between image capture and audio output. The absence of internet connectivity requirements ensures compliance with school policies regarding electronic devices and eliminates concerns about inappropriate content access or online distractions.

## 1.7 CONCEPTUAL FRAMEWORK OF THE SMART ASSISTIVE READING SYSTEM

The proposed smart assistive tool draws upon multiple theoretical and technical foundations identified in the literature:

**1.7.1 Sensory-Integrated Learning Approach**

The system adopts a compensatory rather than purely remedial approach, recognizing that while reading instruction remains important, dyslexic children benefit from immediate access to content through alternative sensory channels (auditory rather than visual decoding).

Research on learning disabilities emphasizes the value of multimodal approaches that engage multiple sensory pathways. The combination of visual text capture and auditory presentation supports diverse learning preferences and reinforces comprehension through dual channels.

**1.7.2 User-Centered Design Considerations**

Drawing from user experience research, the system prioritizes simplicity, offline operation, and independent use—factors identified as critical for sustained adoption in educational settings.

**1.7.3 Community Supported Ecosystem**

Leveraging open-source technologies (Tesseract, Piper, Raspberry Pi OS) addresses cost barriers while enabling customization and community-supported improvement, aligning with research emphasizing accessibility and scalability.

**1.7.4 Multilingual Inclusivity**

Supporting multiple languages directly addresses the gap identified in literature regarding linguistic diversity in assistive technology, particularly relevant for multilingual educational contexts.

**1.8 EDUCATIONAL AND SOCIAL IMPACT ON CHILDREN**

The development of this assistive tool addresses significant educational and social challenges, offering substantial benefits to multiple stakeholders:

**1.8.1 Educational Inclusion**

By providing independent access to written content, the system enables dyslexic children to participate more fully in mainstream education. This reduces educational disparities and promotes inclusive learning environments where diverse learning needs are accommodated through appropriate technological interventions. The offline functionality ensures seamless integration into school environments, allowing children to use the device during classroom activities, library sessions, and examinations without violating institutional policies regarding internet-connected devices. This school-friendly design removes barriers to adoption and enables consistent use across home and educational settings.

**1.8.2 Psychological and Emotional Benefits**

The ability to access text independently reduces frustration and anxiety associated with reading difficulties. Success in comprehending content through auditory means builds confidence, encourages academic persistence, and fosters a more positive attitude toward learning.

**1.8.3 Diverse Language Accessibility**

In India's linguistically diverse context, support for regional languages (Hindi, Malayalam, Telugu) alongside English ensures that children from various linguistic backgrounds can benefit from the technology. This is particularly relevant in rural and semi-urban areas where English proficiency may be limited.

**1.8.4 Economic Accessibility**

Utilizing cost-effective open-source hardware and software components makes the technology financially accessible to middle and lower-income families who cannot afford expensive commercial solutions. This democratizes access to assistive technology and reduces socioeconomic barriers to educational support.

**1.8.5 Parental and Teacher Support**

The device reduces the constant demand on parents and teachers to assist with reading tasks, allowing them to focus on higher-order learning support such as comprehension strategies and critical thinking development.

**1.8.6 Source Scalability and Adaptability**

The open-source nature of the system allows for future enhancements, including additional language support, integration with educational content repositories, and customization for specific learning needs. The project also serves as a foundation for further research in assistive technology for learning disabilities.

**1.8.7 Societal Impact**

By addressing dyslexia-related challenges early in childhood, the system contributes to improved literacy rates, enhanced academic outcomes, and better long-term employment prospects for affected individuals. This has broader implications for social equity and human capital development in society.

# 

# CHAPTER 2

# CURRENT TECHNOLOGICAL LANDSCAPES FOR DYSLEXIA SUPPORT

# 

## 2.1 REVIEW OF RELATED LITERATURE AND TECHNOLOGIES

This chapter presents a comprehensive review of existing literature related to assistive technology for dyslexic children, optical character recognition systems, text-to-speech synthesis, and embedded computing platforms. The review examines recent developments in technological interventions for learning disabilities, identifies gaps in current solutions, and establishes the theoretical and technical foundation for the proposed smart assistive tool.

## 2.2 DYSLEXIA AND LEARNING CHALLENGES

Dyslexia is a neurological learning disorder that affects 15-20% of the global population, characterized primarily by difficulties in phonological processing, word recognition, and decoding abilities. The condition manifests as persistent challenges in reading fluency and accuracy despite normal intelligence and adequate educational opportunities. Research indicates that dyslexic individuals struggle with the automatic segmentation and mapping of phonemes—the smallest units of sound—to written words, a process that occurs effortlessly in typical readers.

The cognitive impact of dyslexia extends beyond mere reading difficulties. Children with dyslexia often experience secondary effects including reduced academic confidence, anxiety related to reading tasks, and avoidance behaviors that further limit their exposure to text-based learning. These psychological consequences can be as debilitating as the primary reading difficulties themselves, creating a cycle of diminished performance and motivation. The prevalence and severity of dyslexia underscore the critical need for effective intervention strategies that address both the functional reading challenges and the associated emotional barriers to learning.

Contemporary research emphasizes that dyslexia is not a visual perception problem but rather a language processing disorder. This distinction is crucial for developing appropriate assistive interventions. While traditional remediation focuses on intensive phonics instruction and reading practice, technological interventions offer complementary approaches by bypassing the decoding difficulties through auditory presentation of text, thereby enabling access to content while cognitive remediation progresses.

## 2.3 ASSISTIVE TECHNOLOGY FOR DYSLEXIA:

### 2.3.1 Text-to-Speech Technology

Text-to-speech technology has emerged as one of the most widely adopted assistive tools for individuals with reading disabilities. Recent systematic reviews demonstrate that TTS applications yield promising results in supporting dyslexic students, particularly in educational contexts. A five-year longitudinal study examining dyslexic students' experiences with assistive technology found that audiobooks and TTS systems were consistently beneficial throughout the school years, with students reporting improved text comprehension and reduced reading fatigue when using these tools.

The effectiveness of TTS technology lies in its ability to circumvent the decoding deficit while maintaining access to written content. Research indicates that listening to text through TTS can enhance comprehension, particularly for narrative texts, as it allows students to focus cognitive resources on understanding meaning rather than struggling with word recognition. However, the utility of TTS varies by content type and individual user characteristics, with some studies noting that expository texts may require additional support strategies beyond simple audio presentation.

Contemporary TTS systems have evolved significantly from earlier mechanical-sounding voices. Neural network-based synthesis models now produce highly natural-sounding speech that closely approximates human vocalization patterns. This improvement in voice quality is not merely aesthetic; studies suggest that natural-sounding TTS output reduces cognitive load and improves comprehension compared to robotic-sounding alternatives, as listeners expend less effort processing the audio signal itself.

### 2.3.2 Mobile Applications and Digital Solutions

The proliferation of smartphones and tablets has democratized access to assistive technology, with numerous mobile applications now available for reading support. Recent research indicates that technological-based interventions, particularly mobile applications and augmented reality solutions, represent the leading edge of assistive technology development for dyslexia. The portability and ubiquity of mobile devices make them attractive platforms for delivering reading assistance in diverse settings.

However, most current mobile-based solutions suffer from significant limitations. Many require continuous internet connectivity to function, relying on cloud-based OCR and TTS services. This dependency renders them unsuitable for classroom environments where internet access may be restricted or unavailable. Additionally, the complexity of many applications presents usability barriers for children, particularly younger users who require intuitive interfaces with minimal configuration requirements. The user experience research reveals that emotional responses to using assistive technology in classroom settings are influenced by dyslexia self-acceptance and attitudes toward the technology itself, suggesting that design simplicity and discretion are crucial factors in adoption.

### 2.3.3 Dedicated Hardware Devices

Specialized reading devices such as scanning pens and electronic readers offer dedicated functionality for reading assistance. These devices typically operate independently without requiring general-purpose computing devices, making them potentially simpler to use. However, scanning pen devices are generally limited to line-by-line reading, which proves impractical for longer passages and does not support holistic comprehension of multi-paragraph text. Furthermore, these dedicated devices are typically expensive, with costs ranging from $200 to $400, placing them beyond the financial reach of many families and educational institutions in developing regions.

The research literature reveals a critical gap in available solutions: while high-end commercial devices offer sophisticated features, their cost and complexity limit accessibility, and while mobile applications are relatively affordable, their internet dependency and operational complexity create barriers to effective use in educational settings. This gap creates an opportunity for intermediate solutions that balance functionality, usability, and affordability.

## 2.4 OPTICAL CHARACTER RECOGNITION TECHNOLOGY

### 2.4.1 Tesseract OCR Engine

Optical Character Recognition technology serves as the foundation for converting printed text into machine-readable format. Tesseract OCR, originally developed by Hewlett-Packard and subsequently maintained by Google, represents one of the most accurate and widely-used open-source OCR engines available. Tesseract has evolved significantly through multiple versions, with version 4.0 introducing a Long Short-Term Memory (LSTM) neural network-based recognition engine that substantially improved accuracy over previous character-pattern-based approaches.

Benchmark studies evaluating OCR accuracy across multiple platforms indicate that Tesseract performs competently on well-formatted digital screenshots, achieving accuracy rates exceeding 95% on clear, high-contrast text. The engine supports over 100 languages through trained models, making it suitable for multilingual applications. This extensive language support is particularly relevant for regions with linguistic diversity, where assistive tools must accommodate multiple scripts and languages.

However, Tesseract's performance is highly dependent on input image quality. Research demonstrates that accuracy decreases significantly with factors such as low resolution (below 300 DPI), poor contrast, image skew, noise artifacts, and unusual fonts. Studies have shown that preprocessing techniques including binarization, noise reduction, deskewing, and resolution enhancement can substantially improve recognition accuracy. One investigation demonstrated that adaptive preprocessing guided by reinforcement learning could boost character-level accuracy from 13.4% to 61.6% on challenging datasets, highlighting both the engine's sensitivity to input quality and the potential for improvement through proper image preparation.

### 2.4.2 Preprocessing and Optimization

The literature emphasizes that successful OCR implementation requires careful attention to image preprocessing. Critical factors include adequate resolution (minimum 300 DPI recommended), proper contrast between text and background, horizontal text alignment, and removal of noise artifacts. Tesseract performs various image processing operations internally using the Leptonica library, but these automatic adjustments may be insufficient for suboptimal input images.

Effective preprocessing strategies documented in research include grayscale conversion, Otsu's binarization for optimal thresholding, morphological operations (erosion and dilation) for noise removal, and geometric transformations to correct skew and perspective distortion. The selection and sequencing of these preprocessing steps significantly impact final recognition accuracy, with empirical studies showing that combinations of resizing, sharpening, and binarization can improve accuracy from 70% to over 90% on scanned documents.

Page segmentation mode configuration represents another critical parameter affecting Tesseract's performance. The engine offers multiple segmentation modes optimized for different document layouts, from full automatic page analysis to single-line or single-word recognition. Selecting appropriate segmentation modes based on the expected input format can substantially improve both accuracy and processing speed.

## 

## 2.5 TEXT-TO-SPEECH SYNTHESIS SYSTEMS

### 2.5.1 Neural Network-Based TTS

Recent advances in text-to-speech synthesis have been driven by deep learning approaches, particularly neural network architectures based on the VITS (Variational Inference with adversarial learning for end-to-end Text-to-Speech) framework. These models generate speech directly from text through learned representations, producing significantly more natural-sounding output than traditional concatenative or parametric synthesis methods.

Contemporary neural TTS systems model prosodic features including intonation, rhythm, and speaking rate more effectively than earlier approaches, resulting in speech that closely approximates human vocalization patterns. This naturalness is not merely aesthetic; cognitive research indicates that natural-sounding synthetic speech reduces listening effort and improves comprehension, particularly for sustained listening tasks such as reading extended passages.

### 2.5.2 Piper TTS Framework

Piper represents a fast, local neural text-to-speech system specifically optimized for resource-constrained devices such as the Raspberry Pi 4. Unlike cloud-based TTS services that require continuous internet connectivity, Piper operates entirely locally, processing text and generating speech on-device without external dependencies. This offline capability is crucial for educational applications where internet access may be restricted or unreliable.

Piper utilizes ONNX (Open Neural Network Exchange) runtime for efficient model inference, enabling high-quality speech synthesis on modest hardware. The framework supports multiple languages and voice profiles, with pre-trained models available through open-source repositories. Voice models are trained using the VITS architecture and exported to ONNX format for optimal performance on edge devices. The system is designed to deliver natural-sounding speech with minimal latency, making it suitable for real-time applications where immediate audio feedback is required.

The offline operation and computational efficiency of Piper make it particularly well-suited for educational assistive devices. Unlike commercial cloud-based alternatives, Piper's open-source nature and local processing eliminate concerns about data privacy, internet dependency, and recurring service costs, while maintaining voice quality comparable to more resource-intensive systems.

## 2.6 RASPBERRY PI AS AN EDUCATIONAL PLATFORM

### 2.6.1 Hardware Capabilities

The Raspberry Pi 4 Model B represents a significant advancement in single-board computing, offering quad-core ARM Cortex-A72 processor architecture with configurations ranging from 2GB to 8GB of RAM. This computational capacity is sufficient for real-time machine learning inference, image processing, and audio synthesis tasks. The platform includes integrated hardware acceleration for video decoding and GPIO (General Purpose Input/Output) interfaces for connecting cameras, sensors, and peripheral devices.

Research on assistive technology implementations using Raspberry Pi demonstrates the platform's viability for computer vision and machine learning applications. Studies have successfully deployed TensorFlow-based object detection systems on Raspberry Pi 4 for assistive navigation, demonstrating real-time performance sufficient for practical applications. The platform's combination of adequate computational power, low cost (approximately $35-75 depending on RAM configuration), and extensive community support makes it an attractive foundation for educational assistive devices.

### 2.6.2 Educational Adoption and Accessibility

The Raspberry Pi Foundation has established the platform as a standard tool in computing education worldwide, with extensive documentation, teaching resources, and community support available. This educational ecosystem reduces barriers to adoption and facilitates knowledge transfer for developing and maintaining assistive technology applications.

The platform's affordability is particularly significant for assistive technology applications intended for resource-constrained settings. At a fraction of the cost of commercial assistive devices or general-purpose tablets, Raspberry Pi-based solutions can be deployed at scale in schools and communities with limited technology budgets. The open-source software ecosystem surrounding the platform further reduces costs by eliminating licensing fees and enabling customization for specific use cases.

Research on educational technology adoption indicates that cost, ease of use, and local technical support are primary factors determining successful deployment in educational settings. Raspberry Pi addresses these factors through its low cost, extensive documentation, and global community of developers and educators who can provide support and share implementation experiences.

## 2.6.3 Multilingual Support in Assistive Technology

The linguistic diversity of many regions, particularly in multilingual countries like India, necessitates assistive technology solutions that support multiple languages and scripts. Research indicates that most commercial assistive tools predominantly support English, leaving non-English speaking populations underserved. This linguistic gap is particularly problematic for dyslexic children whose primary language is not English, as reading difficulties are compounded by the challenge of accessing support tools that function only in unfamiliar languages.

Both Tesseract OCR and neural TTS systems have made significant strides in multilingual support. Tesseract provides pre-trained models for numerous languages including various Indian languages such as Hindi, Telugu, and Malayalam, enabling text recognition across different scripts. Similarly, modern TTS frameworks including Piper offer voice models for multiple languages, though the quality and availability of models varies across languages.

The technical challenges of multilingual support extend beyond simple model availability. Different languages employ distinct phonological systems, orthographic conventions, and text layouts that may require language-specific preprocessing and post-processing. Indian languages in particular present unique challenges due to their complex scripts, ligatures, and diacritical marks. Research demonstrates that recognition accuracy for Indian language scripts generally lags behind Latin-based languages, though recent advances in deep learning-based OCR have narrowed this gap.

## 2.7 GAPS IN EXISTING RESEARCH AND SOLUTIONS

Despite significant advances in assistive technology for dyslexia, several critical gaps remain in current solutions:

**2.7.1 Offline Functionality**

Most advanced assistive applications require internet connectivity, limiting their applicability in schools and regions with unreliable internet access. Research specifically addressing offline capable assistive reading tools for dyslexia remains limited.

**2.7.2 Cost Accessibility**

Commercial assistive devices typically cost hundreds of dollars, placing them beyond reach for many families and schools in developing regions. Research on low-cost, open-source alternatives specifically optimized for educational settings is relatively sparse.

**2.7.3 Language Support**

While technologies support multiple languages individually, integrated systems providing seamless multilingual support—particularly for Indian languages—are uncommon in the research literature and commercial market.

**2.7.4 Educational Context**

Much assistive technology research focuses on clinical or home settings, with less emphasis on school-specific requirements such as compliance with institutional policies, integration with curriculum, and support for classroom workflows.

**2.7.5 Age-Appropriate Design**

Limited research addresses the specific usability needs of elementary school-aged children (ages 6-12) for assistive reading tools, with most studies focusing on adolescents or adults.

## 2.8 RESEARCH INSIGHTS AND IDENTIFIED GAPS

The literature review establishes that while substantial research and development efforts have advanced assistive technology for dyslexia, significant gaps remain in affordable, offline-capable, multilingual solutions suitable for educational settings. Text-to-speech technology has demonstrated clear benefits for dyslexic readers, particularly when implemented with high-quality neural synthesis. Optical character recognition technology, specifically Tesseract OCR, provides mature open-source capabilities for text extraction across multiple languages, though performance depends critically on input quality and preprocessing.

The Raspberry Pi 4 platform offers sufficient computational resources for real-time OCR and TTS processing at a fraction of the cost of commercial alternatives, with extensive community support and educational adoption. However, integration of these components into a cohesive, user-friendly assistive reading tool specifically designed for dyslexic children in multilingual educational contexts represents an underexplored area with substantial potential for impact.

# 

# CHAPTER 3

# RASPBERRY PI 4 MODULE

# 

## 3.1 PROCESSING LOGIC AND SYSTEM FUNCTIONALITY OVERVIEW

The Smart Assistive Tool for Dyslexia Kids employs a modular architecture that integrates image acquisition, optical character recognition, language detection, and neural text-to-speech synthesis into a cohesive assistive reading system. The architecture is designed around a centralized processing model where the Raspberry Pi 4 Model B serves as the primary computational unit, coordinating all peripheral devices and executing the core processing pipeline.

The system follows a sequential processing paradigm comprising four primary stages: image acquisition, preprocessing and optical character recognition, text processing and language identification, and speech synthesis with audio output. This linear workflow ensures predictable operation and minimizes latency between user action (capturing text) and system response (audio playback). The architecture prioritizes offline operation, eliminating dependencies on cloud services or internet connectivity, thereby ensuring consistent performance in diverse educational environments including classrooms, libraries, and examination halls.

The modular design facilitates maintainability and future extensibility. Each functional module—image capture, OCR processing, TTS synthesis, and audio output—operates as a discrete component with well-defined interfaces. This modularity enables independent testing, debugging, and enhancement of individual components without affecting the overall system integrity. The architecture also supports potential future enhancements such as additional language models, alternative TTS voices, or integration with external storage devices for saving processed text.

Power efficiency constitutes a critical architectural consideration given the system's portable nature. The architecture incorporates power management strategies including selective activation of peripherals, optimized processing algorithms that minimize CPU utilization, and efficient memory management to reduce system load. These considerations enable extended operation on battery power, supporting typical classroom usage patterns of 4-6 hours on a single charge.

**3.2 CIRCUIT CONNECTION LAYOUT**

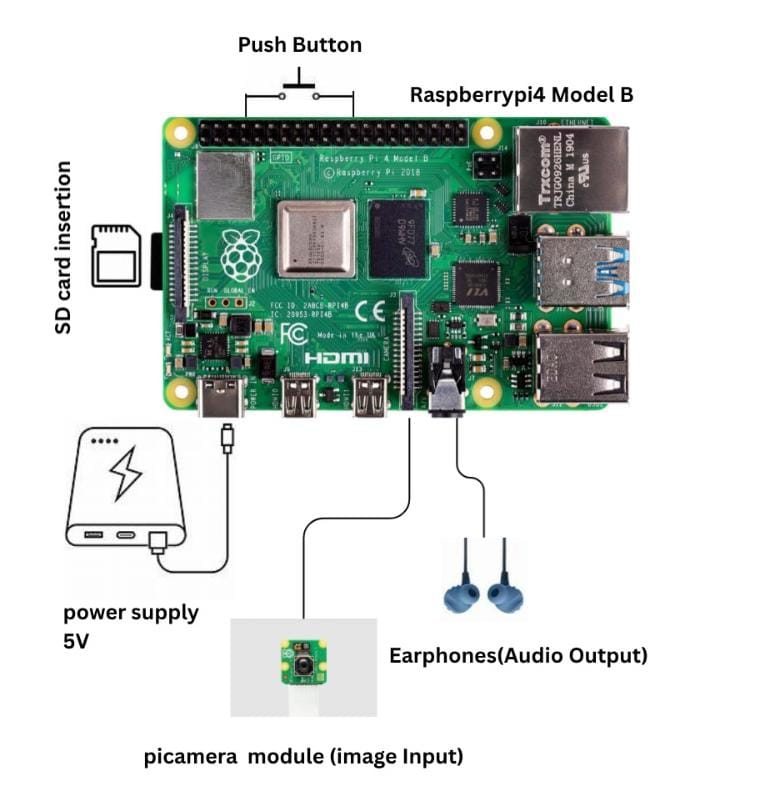
****

Fig. 3.1 Circuit Diagram Of Raspberry Pi 4 Model B

**Step 1**: The Raspberry Pi 4 Model B functions as the main processing unit and manages all connected components.

**Step 2**: The Camera Module is connected to the CSI port of the Raspberry Pi, enabling image capture for processing and analysis.

**Step 3**: The Push Button is connected between GPIO 17 (pin 11) and GND (pin 9)

**Step 4**: The 3.5mm Earphones are connected to the audio jack of the Raspberry Pi to deliver voice feedback through the text-to-speech engine.

**Step 5**: The Raspberry Pi receives power through the USB Type-C port using a 5V, 3A power adapter or a portable power bank, ensuring consistent and reliable operation.

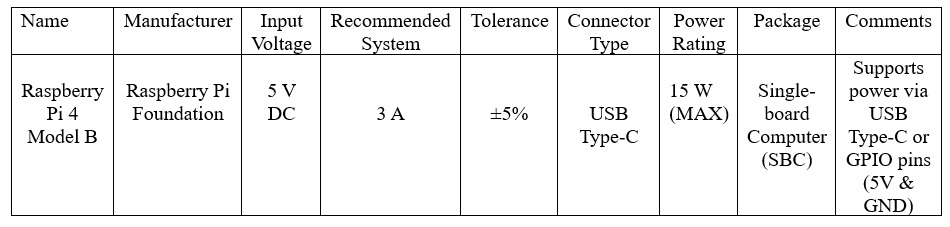
****

Table 3.1 Power Supply Requirements for Raspberry Pi 4 Model B

**3.3 FUNCTIONAL BLOCK REPRESENTATION**

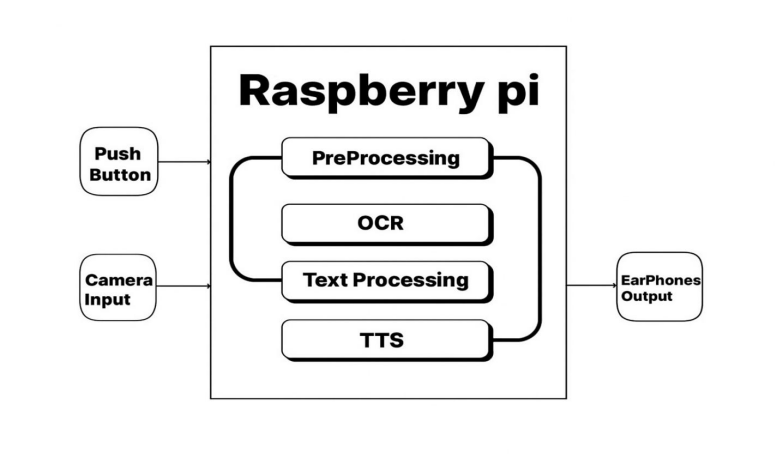
****

Fig. 3.2 Block Diagram Of Raspberry Pi

**3.4 HARDWARE REQUIREMENTS**

**3.4.1 Raspberry Pi 4 Model B (4GB RAM)**

****

Fig. 3.3 Raspberry Pi 4 Model B

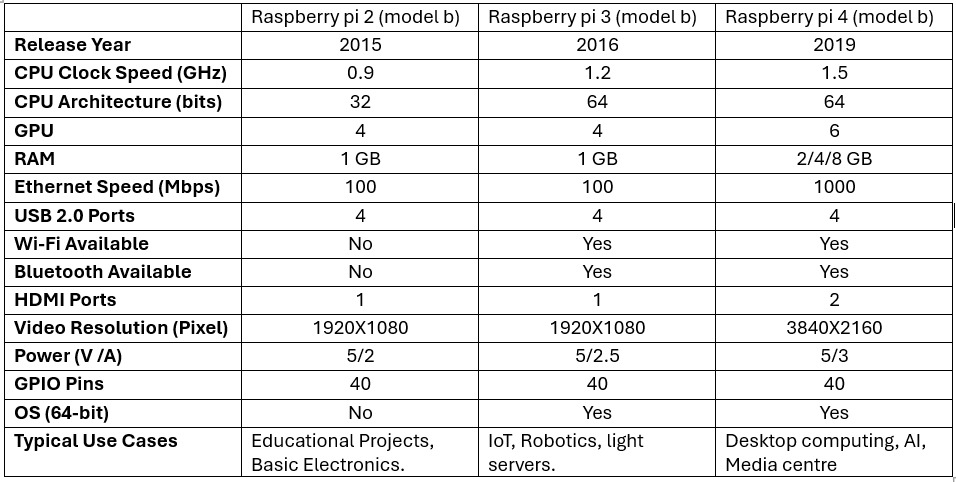
**RASPBERRY PI SPECIFICATIONS**

Table 3.2 Raspberry Pi 4 Model B Specifications

The Raspberry Pi 4 Model B serves as the central processing unit of the project, controlling all connected components and executing software tasks.  
It is equipped with a 1.5GHz quad-core ARM Cortex-A72 processor, 4GB of LPDDR4 RAM, dual-band Wi-Fi, Bluetooth 5.0, multiple USB 2.0 and 3.0 ports, GPIO pins, and a CSI camera interface.  
These features allow it to efficiently interface with hardware components such as cameras, buttons, and audio devices, while also running AI and OCR algorithms in real-time.  
Its compact size, low power consumption, and high computational ability make it ideal for embedded vision systems, portable assistive devices, and applications requiring real-time processing.  
The board’s Linux-based environment provides flexibility, security, and access to a large repository of libraries, making development faster and more reliable.

**3.4.2 Raspberry Pi Camera rev 1.3**

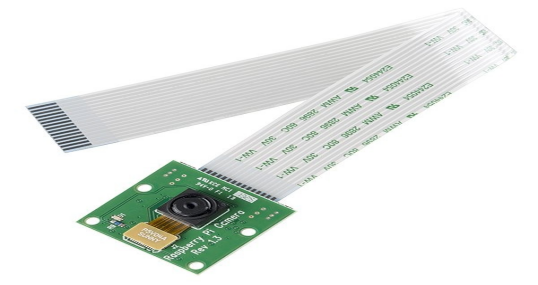
****

Fig. 3.4 Pi Camera rev 1.3

This camera module is connected to the CSI (Camera Serial Interface) port on the Raspberry Pi using a flat ribbon cable, with metal contacts facing the HDMI port.  
It captures high-resolution images and videos with its 8-megapixel Sony IMX219 sensor, providing clear input for image recognition or text extraction.  
The module’s compact design and plug-and-play compatibility make it well-suited for computer vision applications such as OCR, object detection, and scene description.

**3.4.3 Micro SD Card (32GB)**

****

Fig. 3.5 Micro SD Card

The Micro SD card is inserted into the slot beneath the Raspberry Pi board and acts as the primary storage medium.  
It stores the operating system (Raspberry Pi OS), project code, configuration files, and any additional software libraries.  
A 32GB capacity ensures ample space for the system, temporary data storage, and image or audio files generated during execution.  
This component provides fast data access, reliability, and compact storage, ensuring smooth operation and minimal delays during image capture, processing, or audio playback.

**3.4.4 Power Supply (5V, 3A)**

****

Fig. 3.6 Power Supply

The Raspberry Pi is powered through the USB Type-C port using a stable 5V, 3A power supply.  
For portable operation, a power bank with USB output can be used, allowing the system to run without a fixed electrical connection.  
Reliable power supply ensures continuous and stable performance, preventing sudden shutdowns during critical tasks such as image processing or text-to-speech synthesis.  
This flexibility allows the device to be used in field applications, outdoor environments, or wearable assistive devices, increasing its usability and convenience.

**3.4.5 3.5mm wired earphones**

****

Fig. 3.7 Wired Earphones

The earphones connect to the 3.5mm audio jack on the Raspberry Pi and provide auditory feedback to the user.  
They convert processed visual information or recognized text into speech output using the Text-to-Speech (TTS) engine, enabling real-time interaction.  
This functionality enhances accessibility, particularly for visually impaired or dyslexic users, by delivering immediate and understandable audio feedback.  
The use of wired earphones ensures low-latency and consistent audio quality, crucial for accurate interpretation of text-to-speech output.

**3.4.6 Push Button**

****

Fig. 3.8 Push Button

The push button serves as a manual input device for the user, allowing interaction without a keyboard or touchscreen.  
It is connected between a GPIO pin (e.g., GPIO 17) and GND, enabling the Raspberry Pi to detect user input through a simple press.  
When activated, it can trigger image capture, start text-to-speech output, or initiate other programmed tasks, providing intuitive and accessible control over the system.  
The push button adds ease of operation, reliability, and tactile feedback, making the device user-friendly and suitable for assistive applications.

### 

### 3.5 SOFTWARE REQUIREMENTS

### 3.5.1 Operating System: Raspberry Pi OS – Bullseye Version

Raspberry Pi OS, formerly known as Raspbian, serves as the base operating system for the project. It is a Debian-based Linux distribution specifically optimized for Raspberry Pi hardware. The operating system provides kernel drivers for various peripherals such as the camera, GPIO pins, and audio interface. It also includes pre-configured services and the APT package manager, enabling convenient installation and updates of required software packages.

The Lite (headless) version of Raspberry Pi OS is preferred to reduce resource consumption and eliminate graphical interface overhead. Being based on Linux, the system offers reliable process management, excellent hardware compatibility, and access to a vast repository of Python libraries. It provides a secure, customizable, and open-source development environment without licensing restrictions, making it ideal for embedded AI applications.

#### 3.5.2 Programming Language: Python

Python 3 is used as the main programming language for software development. It was chosen for its simplicity, readability, and extensive support for AI, computer vision, and text-to-speech libraries. The codebase is modularized into separate components for image capture, optical character recognition, text-to-speech, and audio playback.

Python’s interpreted nature simplifies testing and debugging, while its performance remains sufficient for sequential real-time tasks. Since the project follows a single-threaded execution model, the Global Interpreter Lock (GIL) does not impact performance.

#### 

#### 3.5.3 Optical Character Recognition: Tesseract OCR

Tesseract OCR, version 4.1 or above, is employed for text recognition. It utilizes LSTM-based neural network models to improve accuracy, especially when processing complex or multi-language text. The integration is achieved through the Python library pytesseract, which allows direct access to OCR functions and output handling.

Language data packages for English, Hindi, Malayalam, and Telugu are used to enable multilingual text recognition. The OCR engine automatically detects page layouts, performs segmentation, and extracts readable text from captured images.

#### 3.5.4 Text-to-Speech Engine: Piper TTS

Piper Text-to-Speech (TTS) is used for generating high-quality offline speech output. It is optimized for Raspberry Pi using the ONNX Runtime framework and produces natural-sounding WAV audio in multiple languages, including English, Hindi, Malayalam, and Telugu.

The tool is invoked through Python subprocess calls to generate near real-time voice synthesis. Piper TTS ensures clear pronunciation and consistent prosody, making it an efficient and lightweight alternative to cloud-based TTS services.

#### 3.5.5 Supporting Libraries and Dependencies

**OpenCV (cv2):** Used for image capture, resizing, grayscale conversion, and noise r

**pytesseract:** Acts as a Python interface for Tesseract OCR, simplifying integration and output handling.

**sounddevice / PyAudio:** Manages audio playback through the ALSA sound system, supporting real-time voice streaming.

**NumPy:** Handles numerical operations efficiently for processing image arrays and audio data.

**subprocess:** Executes Piper TTS as a child process and manages input/output streams for speech generation.

**RPi.GPIO:** Controls the Raspberry Pi’s GPIO pins, enabling button input detection and event handling.

**Pillow (PIL):** Performs lightweight image operations, such as format conversion, rotation, and transformations.

**CHAPTER 4**

**DESIGN & FUNCTIONAL FLOW OF THE DYSLEXIA READING ASSISTANCE SYSTEM**

**4.1 ALGORITHM**

## 4.1.1 Multi-Stage OCR-to-TTS Pipeline Algorithm

The Smart Assistive Tool implements the Multi-Stage OCR-to-TTS Pipeline Algorithm, which converts captured text into auditory feedback through a sequence of well-defined stages. Each stage performs a critical part of the process, ensuring accurate recognition and smooth user experience.

### Stage 1: Image Capture

The tool first activates the Pi Camera to capture an image of the text, whether handwritten or printed. The raw image is initially in color (BGR format) and may include uneven lighting, shadows, or background noise. Capturing a high-quality image at this stage is crucial for the success of subsequent stages.

### Stage 2: Grayscale Conversion

Once the image is captured, it is converted into a grayscale image. This reduces the data complexity by collapsing the color channels into a single intensity channel, enhancing the contrast between text and background. Grayscale conversion simplifies the image for further processing and improves OCR accuracy.

### 

### Stage 3: Image Enhancement

The grayscale image is optionally processed to improve clarity. Thresholding may be applied to convert the image into a binary format, where text is distinctly separated from the background. Noise removal techniques are employed to eliminate artifacts, smudges, or distortions that could interfere with text recognition. At the end of this stage, the image is clean and optimized for OCR.

### Stage 4: Text Recognition

The preprocessed image is passed to Tesseract OCR, which identifies individual characters and converts them into machine-readable text. This stage forms the core of the algorithm, where visual information is transformed into a textual representation.

### Stage 5: Language Detection

After extracting text, the algorithm determines the script or language of the text, such as English, Hindi, Telugu, or Malayalam, using Unicode-based character detection. This ensures that the correct language-specific processing is applied, improving the accuracy of recognition and speech synthesis.

### Stage 6: Text Cleaning and Processing

The recognized text is cleaned to remove unnecessary spaces, correct common OCR errors, and optionally apply dyslexia-friendly formatting. This processing step ensures that the text is accurate, readable, and suitable for auditory conversion.

### Stage 7: Text-to-Speech Conversion

The processed text is sent to a TTS engine, such as PiperVoice, which converts the text into audio. The resulting speech is played through the earphones, providing immediate feedback to the user. This stage completes the assistive process by delivering the recognized text in an accessible audio format.

### Stage 8: Continuous Operation

The algorithm continuously loops through these stages as new text input is captured. This allows the tool to provide real-time reading assistance for multiple lines or pages of text without interruption.

**4.2 PSEUDO CODE**

BEGIN

1. Initialize Raspberry Pi and Pi Camera

2. Initialize TTS engine

WHILE tool is active DO

Step 1: Capture Image

- Capture image using Pi Camera

- Ensure proper focus and lighting

Step 2: Preprocess Image

- Convert image to grayscale

- Apply thresholding to separate text from background

- Remove noise if necessary

Step 3: Recognize Text (OCR)

- Extract text using Tesseract OCR

- Clean recognized text (remove extra spaces, correct OCR errors)

Step 4: Process Text

- Analyze text readability

- Apply optional spell-checking or dyslexia-friendly formatting

Step 5: Provide Audio Feedback

- Convert processed text to speech using TTS

- Play audio through earphones

END WHILE

END

**4.3 CODE**

**Stage 1: Image Capture**

This stage captures an image from the Pi Camera.

from picamera2 import Picamera2, Preview

import time

import cv2

def main():

picam = Picamera2()

# Configure preview

preview\_config = picam.create\_preview\_configuration(main={"size": (640, 480)})

picam.configure(preview\_config)

picam.start\_preview(Preview.QTGL)

picam.start()

print(" Camera preview started.")

print("Press 'c' + Enter to capture an image, 'q' + Enter to quit.")

while True:

key = input("Enter command: ")

if key.lower() == 'c':

# Capture image using Picamera2

frame = picam.capture\_array()

filename = "captured\_image.jpg"

cv2.imwrite(filename, frame)

print(f" Image captured and saved as '{filename}'")

# ===== Run multi3.py logic after capture =====

run\_multi3() # Call the function defined below

elif key.lower() == 'q':

break

picam.stop\_preview()

picam.stop()

print("Camera stopped.")

if \_\_name\_\_ == "\_\_main\_\_":

main()

**Stage 2: Image Loading & Grayscale Conversion**

This stage loads the captured image and converts it to grayscale for OCR processing.

import cv2

import pytesseract

from piper import PiperVoice

import sounddevice as sd

import numpy as np

import io

import wave

import re

IMAGE\_PATH = r"/home/mypi/Music/captured\_image.jpg"

# Map languages to Piper models and Tesseract codes

LANG\_CONFIG = {

"en": {"tess": "eng", "model": r"en\_US-amy-low.onnx"},

"hi": {"tess": "hin", "model": r"hi\_IN-rohan-medium.onnx"},

"te": {"tess": "tel", "model": r"te\_IN-venkatesh-medium.onnx"},

"ml": {"tess": "mal", "model": r"ml\_IN-arjun-medium.onnx"}}

DEFAULT\_LANG = "en"

def load\_image(image\_path: str):

"""Reads image from disk and converts to grayscale."""

image = cv2.imread(image\_path)

if image is None:

raise FileNotFoundError(f"Image not found: {image\_path}")

gray = cv2.cvtColor(image, cv2.COLOR\_BGR2GRAY)

return gray

**Stage 3: Text Recognition & Language Detection**

Detects the script/language and extracts text using OCR.

def detect\_script\_and\_language(gray\_image) -> str:

text\_all = pytesseract.image\_to\_string(

gray\_image, lang="eng+hin+tel+mal"

).strip()

print("Initial OCR Text for Detection:\n", text\_all)

counts = {

"hi": len(re.findall(r'[\u0900-\u097F]', text\_all)), # Hindi

"te": len(re.findall(r'[\u0C00-\u0C7F]', text\_all)), # Telugu

"ml": len(re.findall(r'[\u0D00-\u0D7F]', text\_all)), # Malayalam

"en": len(re.findall(r'[A-Za-z]', text\_all))} # English

lang = max(counts, key=counts.get)

if counts[lang] == 0:

lang = DEFAULT\_LANG

print(f"Detected language: {lang}")

return lang, text\_all

**Stage 4: Refined OCR for Detected Language**

def extract\_text\_for\_language(gray\_image, lang\_code: str) -> str:

"""Runs OCR again for the detected language to improve accuracy."""

tess\_lang = LANG\_CONFIG[lang\_code]["tess"]

text = pytesseract.image\_to\_string(gray\_image, lang=tess\_lang).strip()

print(f" Extracted Text ({lang\_code}):\n{text}")

if not text:

raise ValueError(f"No text found for language {lang\_code}.")

return text

**Stage 5: Text-to-Speech Conversion**

def speak\_text\_with\_piper(text: str, lang\_code: str):

"""Converts text to speech using Piper and plays it."""

model\_path = LANG\_CONFIG[lang\_code]["model"]

print(f" Speaking using {lang\_code} model: {model\_path}")

voice = PiperVoice.load(model\_path)

with io.BytesIO() as wav\_buffer:

with wave.open(wav\_buffer, 'wb') as wav\_file:

wav\_file.setnchannels(1)

wav\_file.setsampwidth(2)

wav\_file.setframerate(22050)

voice.synthesize\_wav(text, wav\_file)

wav\_buffer.seek(0)

with wave.open(wav\_buffer, 'rb') as wf:

data = wf.readframes(wf.getnframes())

audio\_data = np.frombuffer(data, dtype=np.int16)

sd.play(audio\_data, samplerate=wf.getframerate())

sd.wait()

**Stage 6: Main Function to Run All Stages**

def run\_multi3():

try:

gray = load\_image(IMAGE\_PATH)

lang, \_ = detect\_script\_and\_language(gray)

refined\_text = extract\_text\_for\_language(gray, lang)

speak\_text\_with\_piper(refined\_text, lang)

except Exception as e:

print(" Error:", e)

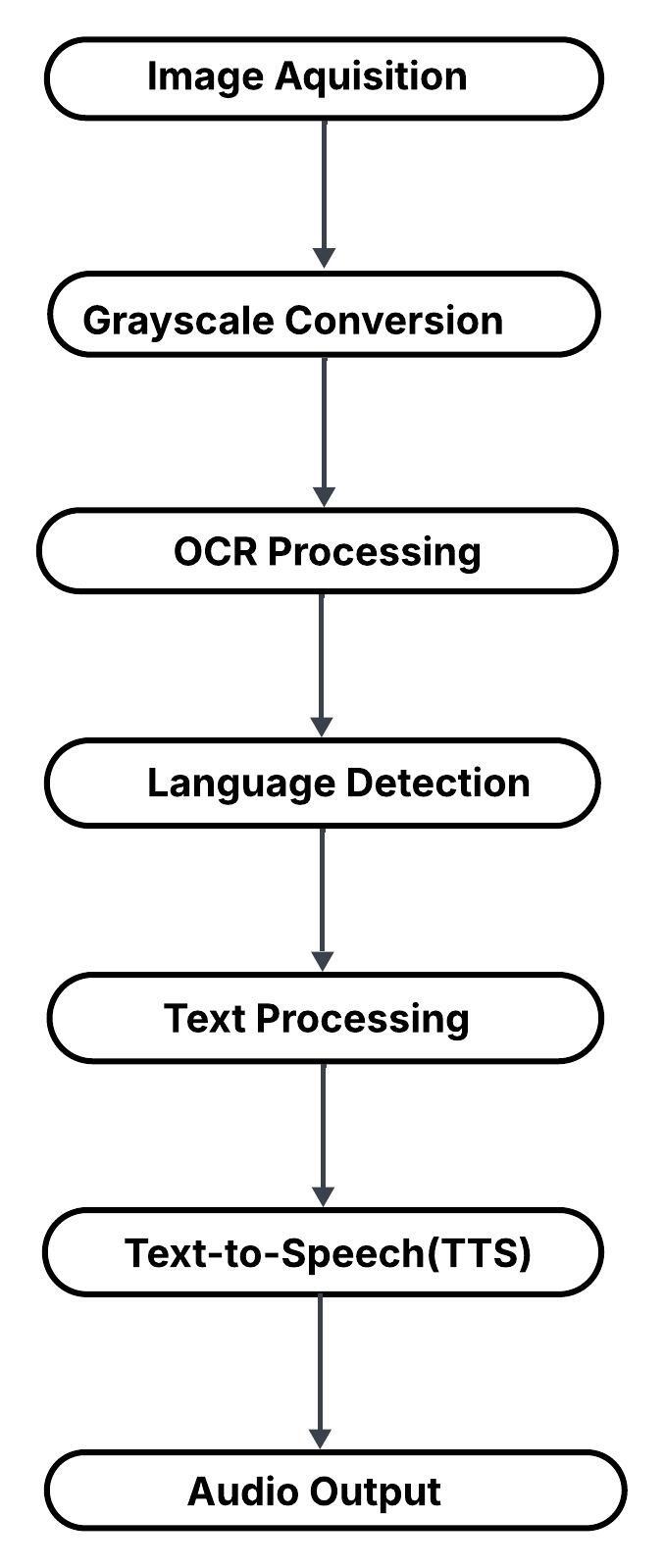
**4.4 SEQUENTIAL WORKFLOW**

Fig. 4.1 Sequential Work Flow

**CHAPTER 5**

**SCREENSHOTS**

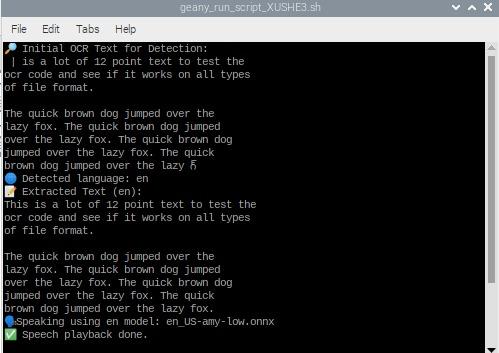
****

Fig. 5.1 English Audio Response

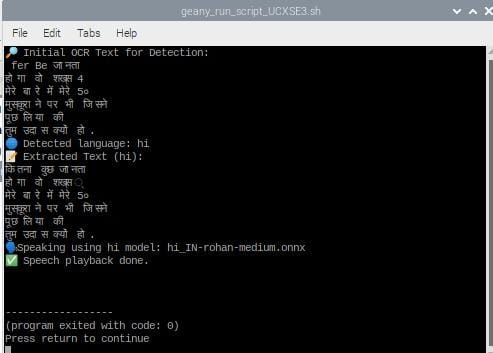


Fig 5.2 Hindi Audio Response

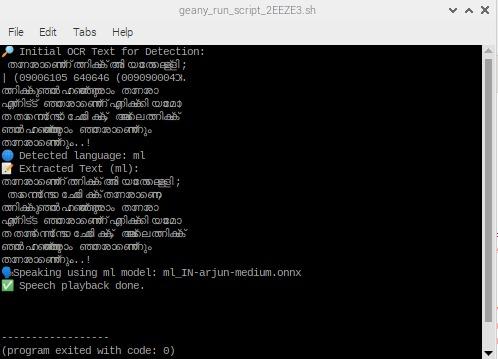
****

Fig. 5.3 Malayalam Audio Response

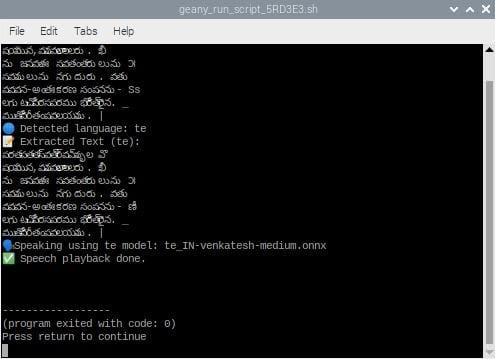
****

Fig. 5.4 Telugu Audio Response

**CHAPTER 6**

**CONCLUSION**

**6.1 FINAL EVALUATION AND PEDAGOGICAL IMPLICATIONS**Scan n Learn represents a significant advancement in assistive technology for children with dyslexia, offering a practical and scalable solution to one of the most persistent challenges in inclusive education. By integrating high-resolution image acquisition, script-aware Optical Character Recognition, and low-latency Text-to-Speech synthesis into a cohesive Multi-Stage OCR-to-TTS Pipeline, the system effectively transforms printed text into accessible auditory content. This transformation addresses the core difficulties faced by dyslexic learners, including word recognition, decoding accuracy, and cognitive fatigue, thereby creating a more equitable learning environment.

The multilingual capability of Scan n Learn, supporting English, Hindi, Telugu, and Malayalam, demonstrates its adaptability to diverse linguistic contexts and makes it particularly valuable in multilingual educational settings. The use of Raspberry Pi as the hardware platform ensures cost-effectiveness and portability, making the tool accessible to schools, learning centers, and households with limited technological infrastructure. Through structured preprocessing techniques such as grayscale conversion and contrast enhancement, combined with language-specific Tesseract models, the system achieves robust text extraction with minimal recognition errors, even from materials with varying print quality.

Beyond technical functionality, Scan n Learn empowers dyslexic children by reducing the cognitive burden associated with reading, allowing them to focus on comprehension rather than decoding. The auditory feedback mechanism provides immediate support, fostering independent learning and building reading confidence over time. This shift from visual to auditory processing leverages the strengths of dyslexic learners, who often exhibit strong verbal and listening comprehension despite challenges with written text. By enabling children to engage with printed materials autonomously, the system promotes self-reliance and reduces dependency on continuous adult supervision.

Furthermore, the modular architecture of Scan n Learn ensures that the system is not a static solution but a foundation for continuous improvement. The pipeline's deterministic design facilitates debugging, optimization, and extension, making it well-suited for iterative development and research-driven enhancements. The integration of error-handling routines and text normalization mechanisms ensures reliability in real-world usage scenarios, where variations in text formatting, fonts, and print quality are common.

In conclusion, Scan n Learn successfully bridges the gap between visual text and auditory comprehension, offering dyslexic children a reliable, user-friendly, and effective assistive tool. By addressing both technical and pedagogical challenges, the system contributes meaningfully to the broader goal of inclusive education and demonstrates the potential of accessible technology to transform learning experiences for children with neurodevelopmental differences.

## 6.2.FUTURE ENHANCEMENTS

While Scan n Learn provides a robust foundation for assistive reading technology, several enhancements can further improve its functionality, accessibility, and educational impact. These proposed enhancements span technical improvements, user experience refinements, and educational integration strategies.

**6.2.1. Advanced Machine Learning & AI Integration**

Incorporating deep learning-based OCR models such as EasyOCR or PaddleOCR could significantly improve text recognition accuracy, particularly for degraded or low-quality prints, handwritten notes, and documents with complex layouts. Neural network-based approaches offer superior performance in handling diverse fonts, stylistic variations, and multi-column text structures. Additionally, implementing real-time error correction using Natural Language Processing (NLP) techniques such as contextual spell-checking and grammar correction would enhance the quality of extracted text before TTS conversion, reducing confusion caused by OCR misrecognition.

**6.2.2. Expanded Language Support and Script Recognition**

Extending the system's multilingual capabilities to include additional Indian and international languages would broaden its applicability across diverse regions. Support for languages such as Tamil, Kannada, Bengali, Urdu, and regional scripts would make the tool more inclusive. Furthermore, implementing automatic language detection would eliminate the need for manual language selection, streamlining the user experience and enabling seamless transitions between multilingual documents.

**6.2.3. Enhanced User Interface and Accessibility Features**

Developing a mobile application or web-based interface would improve usability, allowing parents, teachers, and children to interact with the system more intuitively. Features such as adjustable speech speed, voice modulation options, and dyslexia-friendly font rendering on-screen would cater to individual preferences and learning styles. Incorporating visual highlighting synchronized with audio playback could support dual-coding learning strategies, reinforcing word recognition and comprehension through simultaneous visual and auditory cues.

**6.2.4. Personalization and Adaptive Learning**

Integrating user profiles with tracking mechanisms would enable the system to monitor reading progress, identify frequently misread words, and provide personalized vocabulary-building exercises. Adaptive learning algorithms could adjust the complexity of recommended reading materials based on the child's proficiency level, ensuring age-appropriate and skill-appropriate content delivery. Gamification elements such as reading badges, progress milestones, and interactive rewards could further motivate children and sustain engagement over extended periods.

**6.2.5. Hardware Improvements and Alternative Form Factors**

Exploring alternative hardware configurations, such as wearable cameras or smartphone-based implementations, would enhance portability and user convenience. Lightweight, ergonomic designs would make the tool more practical for daily use in classrooms and homes. Additionally, incorporating higher-resolution cameras and improved lighting mechanisms would ensure consistent performance across varying environmental conditions.

**CHAPTER 7**

**REFERENCE**

These are the references of the Dyslexia Project*,* highlighting important studies and technological advancements that contributed to the development of assistive tools for dyslexic individuals.They provide valuable insights into mobile applications, AI-based reading aids, and speech recognition technologies.Each reference supports the research foundation and design principles implemented in this project.

#### ****References****

[1] Allafi, R. & Newbury, P. “Importance of Assistive Mobile Applications for Dyslexic Students in Saudi Arabia.” **PUPIL: International Journal of Teaching, Education and Learning**, 5(3), 1–12 h(2021). DOI: 10.20319/pijtel.2021.53.0112. GRDS Publishing.Looks at mobile app usage for dyslexic students in an Arabic context—indicates cross-language relevance.

[2] Baulch, M. “OrCam Read: An Assistive Technology Device to Increase Access to Reading for Students with Dyslexia.” **White Paper**, OrCam Technologies Ltd (2021). Available at: explore.orcam.com - Real-world evaluation of the OrCam Read device on students aged 7–15 with dyslexia. Demonstrated improvements in independent reading and comprehension; supports the use of AI-based reading aids like Scan n Learn.

[3] Christopher, D. F. X. & Zeema, J. L. “Study on the Frontiers of Assistive Technologies for Smart Learning in Learning Impairment of Dyslexic Children.” **International Journal of Computer Applications**, Vol. 182, No. 23, October 2018. IJCA Online .Explores types of assistive tech (text-to-speech, multisensory, games) for dyslexic learners—useful for feature and technology decisions in your tool.

[4] Gupta, T., Aflatoony, L., & Leonard, L. “‘A Helping Hand’: Design and Evaluation of a Reading Assistant Application for Children with Dyslexia.” **Proceedings of the 33rd Australian Conference on Human-Computer Interaction**, 242–251 (2021). DOI: 10.1145/3520495.3522703. OUCI.  
Practical application design and evaluation for dyslexic children.

[5] Hornickel, J., Zecker, S. G., Bradlow, A. R., & Kraus, N. “Assistive Listening Devices Drive Neuroplasticity in Children with Dyslexia.” **Proceedings of the National Academy of Sciences of the U.S.A.,** 109(41), 16731–16736 (2012). DOI: 10.1073/pnas.1206628109. PMC - Demonstrates that assistive devices can lead to measurable neural and reading improvements—great for supporting your tool’s efficacy justification.

[6] Mohamad, M., Abdullah, N., Rahim, N., Sanmugam, M., & Takajo, H. “Mobile Assistive Technology for Dyslexic Children: A Significant Review.” **Journal of Advanced Research in Applied Sciences and Engineering Technology,** 51(2), 57–70 (2024). DOI: 10.37934/araset.51.2.5770. Semarak Ilmu - A comprehensive review of mobile assistive tech for dyslexic children—useful for your tool’s mobile/app component.

[7] Ramlan, S. A., Isa, I. S., Harron, N. A., Mohd Saod, A. H., & Azid, M. F. “Reading Assistive Tool (ReaDys) for Dyslexic Children: Speech Recognition Performance.” **Journal of Creative Practices in Language Learning and Teaching**, 11(2) (2023/24). DOI: 10.24191/cplt.v11i2.2045. UiTM Journal.Specific prototype system for dyslexic children using speech-to-text—relevant for interaction and interface design in your tool.