ONE EYE A Smart signboard reader for the visually impaired

Manoj Kumar Moogala
School of CSE&AI
S R Universty
Warangal, India
manojreddy1920@gmail.com

Bashaboina RahulThrinethra
School of CSE&AI
S R Universty
Warangal, India
bashaboinarahulthrinethra@gmail.com

Rashwitha Kondle
School of CSE&AI
S R Universty
Warangal, India
rashwithakondle1819@gmail.com

Reddy Srivallika
School of CSE&AI
S R Universty
Warangal, India
reddysrivalili414@gmail.com

Mallela Rohith
School of CSE&AI
S R Universty
Warangal, India
rohithmallela2005@gmail.com

ABSTRACT

Reading the public signboards in a multilingual and multi-modal atmosphere by persons unfamiliar with the native language and those who have visual impairment becomes nearly impossible. The proposed SignBoard Reader system will resolve the issue using the Combination of Text in Speech and Optical Character Recognition technology and accord a real-time and multilingual working system that entails meticulous extraction of text from designs. Run it using the Tesseract.js OCR engine to ensure accurate extraction even in the face of changing lighting conditions. This increases it using the Tesseract.js OCR engine in addition to the size conversion. Through the integrated language picker, users can select Telugu, Hindi, and English versions. The App Speech API can be used to overlay the detected text and rely on instant audio input. The User-Centered Design principle informs the design of the user interface, which is clear, has big icons, and is dark for comfort and simplicity.The offline functionality of Google ML Kit ensures use in locations with inadequate connectivity.Based on user feedback and experimental evaluation, the system offers good usability, accuracy, and satisfaction, particularly for accessibility applications. This study promotes inclusive human–computer interaction and paves the way for future advancements in AI-assisted assistive technology for multilingual environments.

General Terms

Image Processing, Text Detection, Text recognition, Automatic Testing using XML files

Keywords

Text detection, Text recognition, WEB AUDIO API,Google ML KIT, Optical Character recognition (OCR), Text-to-Speech, Tesseract.js, UCD, Web RTC.

I.INTRODUCTION

In a linguistically pluralistic nation like India, public signboards may sometimes be exhibited in 2 or more regional languages, which can create considerable issues for those who have visual impairments, who are illiterate, or who are unfamiliar with the regional language. Without an aide, it can be difficult for anyone to read critical information that is often found on a public signage like directions, warnings or signagefor shops, etc. This inability to read signboarding detracts from a user's ability to remain independent or increase their mobility in public life. Thanks to theemergence of AI(Artificial Intelligence) and computer vision, modern methods of assistance can read visual text in the environment and convert it to speech output or auditory output, which is basically the same level of usability for those individuals who are familiarizinig themselves for the first time in an environment.

The SignBoard Reader project plans to create and develop a real-time multilingual system for automatically detecting, extracting, and Vocalizing text obtained from sign boards. The system incorporates Optical Character Recognition (OCR) to extract text, and Text-to-Speech (TTS) to produce the audio output. This device serves as a valuable and usable assistive device for users with visual impairment. The process begins with capturing the input using a camera and feeding the video feed to be processed by Tesseract.js to extract the extracted text using OCR. Finally, the TTS technology produces the audio output using the Web Speech API. The user is able to select their language of preference (English, Hindi or Telugu) given India's linguistic diversity.

Instead of using pre-read or static images like regular OCR applications, SignBoard Reader works in the real-world conditions of changing lighting, angles, and background scenes. WebRTC enables real-time camera streaming in the web browser, while Google ML Kit offers offline recognition for settings without strong internet connectivity. The system's User Interface (UI) uses User-Centered Design (UCD) to prioritize simplicity, accessibility, and usability. It contains clear buttons to operate, voice prompts, and feedback buttons that indicate actions like "Camera Activated", "Scanning", and "Speech Output", all designed for guidance to let users engage easily without training.

The main objective of this project originates from the growing global demand for accessible AI tools. The World Health Organization (WHO) reports that over 285 million people world-wide experience some form of visual impairment, with many in developing countries where the availability and affordability of assistive technology is limited. By utilizing applicable open-source frameworks and web technologies, the SignBoard Reader tool provides an affordable, open-source and accessible alternative through any device with web browser access.

The research advances the accessible technology field by integrating OCR, speech synthesis, and multilingual capabilities into a lightweight, web-based format. The study also draws evidence from the literature on the principles of HCI, applying them to elements of accessibility and interaction approaches. Finally, the study demonstrates how AI-enabled solutions can help reduce communication barriers and promote social inclusion through auditory comprehension of textual information.

Fig.1

II.LITERATURE REVIEW:

Over the last 10 years, scene text recognition (STR) has experienced rapid developments due to deep learning methods that treat text detection and text transcription from natural images as a joint problem. Much of the early progress focused on recognition systems for Latin scripts, but there is an increasing effort to clarify the difficulties of recognizing Indic scripts including Devanagari, Telugu, and Malayalam. Mathew et al. described a benchmarking study of STR for Devanagari, Telugu, and Malayalam, and found that deep convolutional-recurrent architectures trained on synthetic data can reach reasonable performance levels, but there is still a domain gap in performance on real signboard photographs because of variability in fonts, low-resolution camera models, and complex image backgrounds. They create a new dataset, IIIT-ILST, to provide evaluation and testing results to establish a baseline of model performance on Indian scripts.

Recently, transfer learning has been a valuable approach for mitigating some of the challenges involving limited annotated data in low-resource scripts. Gunna et al. investigated transfer learning over scripts and found that transferring models not across English, but Indian languages, resulted in better visual feature alignment and improved word recognition rates. They investigated synthetic data and multiple STR backbones (CRNN, STAR-Net) and found consistent improvements for Hindi and Telugu, as well as related scripts, leading to the conclusion that a cross-Indic transfer is a valuable method to bootstrap recognition for geographically and typographically related languages. Noting the scarcity of varied, high-quality training data for Indian languages, they built the IndicSTR12 dataset to provide a sufficiently large, multilingual benchmark of twelve Indian languages. IndicSTR12 contains tens of thousands ofauthentic word images as well as large-scale synthetic word images toprovide variability replicating the conditions of the real world (blur, occlusion, and perspective distortion). The dataset and baseline evaluations (PARSeq, CRNN, STARNet) serve as a valuable resource in the aspect of evaluating and improving STR systems focused on Indian scripts. Accepting a curated dataset is vital for adequately estimating reliability of any signboard reader's accuracy in Hindi or Telugu**.**

In the literature on signboard and signage OCR, the focus is often on preprocessing and domain adaptation techniques. For example, Khan et al. proposed various image preprocessing pipelines (HSV-based sign detections, contrast enhancements, and denoising) to improve the readability of OCR on signboards captured through a smartphone. Their research demonstrates that relatively simple, but direct preprocessing steps translationally have an advantage in boosting the performance of commercial or off-the-shelf OCR engines when applied to "noisy" images encountered in the real-world domain. Such engineering is most useful when the OCR engine is lightweight or on device, such as Tesseract in a mobile or browser based experience.

Though progress has been made, specific gaps remain for signboard applications in assisting users. The existing benchmarks, while providing validation of recognition systems in terms of word-level recognition, often lack large-scale and signboard-focused corpora that captured the range of shop fronts, decorative fonts, and multilingual layering that users experience in urban Indian.

Secondly, the literature has explored integrating STR outputs into accessible user actions (low latency TTS for example) in less formal contexts, such as gesture activation or offline methods, leading to areas for applied systems research that couples human-computer interaction (HCI) evaluations with standard technical benchmarks. The areas of usability (HCI) and Indic focused STR recognition today ceremoniously lead into the design of the SignBoard Reader project, where we leverage targeted preprocessing, Indic aware recognition techniques (transfer learning or fine-tuning, for example, on IndicSTR12) and usability focused TTS/UI designs to achieve a user-friendly assistive experience for reading multilin

III.RELATEDWORK

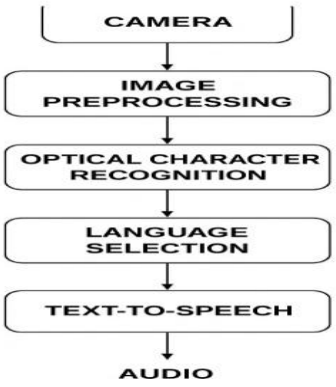
The summary of research in scene text recognition (STR) athe reading of signboards have some areas of overlap. These include detecting and recognizing text in natural images, the Applications of OCR for Indic scripts, preprocessing of the STR task that leads to robust recognition in real-world conditions where signboards and texts are available, and mobile/browser OCR research using a text-to-speech system as an access point for mobile and browsing accessibility. In this section, we review research that has been written and published in these areas as they relate to the SignBoard Reader and how the existing system contributes to that research.

Initially, STR research focused on Latin scripts and used separate detection and recognition stages (detect text regions → recognize text in text regions). Lately, more work has reflected an end-to-end deep architecture (e.g., CNN+RNN or transformer based configurations) that has shown improved robustness to complex backgrounds, changes in fonts, and distortions usually seen in signboards. Benchmark studies for Indic scripts have indicated that while these deep models perform very well on synthetic or high-quality data, there is still a domain gap in the forward/out of domain test scenes when applying models to real scene signboard images because of font distributions, perspective distortions, and conditions under which photos of signboards were taken [1], [2].

Research discussing Indian languages has highlighted the distinct challenges presented by Indic scripts, such as complex glyphs, conjuncts, and ligatures that are specific to that script. Mathew et al. present benchmarking and baseline assessments for model performance on Devanagari, Telugu, and Malayalam. Their findings note that for standard STR models to achieve acceptable accuracy on such languages, careful consideration must be made regarding data augmentation and script-aware tokenization [2]. Additionally, the IndicSTR12 dataset was developed to provide a multilingual benchmark of twelve Indian languages, including Hindi and Telugu, with both real and synthetic examples for various training and evaluation tasks of STR models for Indic scripts [3]. These resources are particularly pertinent to any signboard reader intended for use in the Indian context, as they represent meaningful evaluation and model fine-tuning specific to the language.

IV.METHODOLOGY

The process taken for the design of the SignBoard Reader was an iterative and user-centered approach utilizing elements of software engineering methods and human–computer interaction (HCI) principles, withthe aim of creating a web-based assistive application that could detect, recognize, and speakreal-worldtext from signboards in various Indian languages. The overall process consisted of requirement analysis, system design, module development, integration, and performance evaluation



Requirement Analysis

The project commenced with an understanding of the main requirements of users who are visually-impaired and users with very low levels of literacy, who experience difficulty reading public signboards. The main requirements that emerged from this analysis include:

Real-time camera access and the ability to capture images.

The ability to work offline in low-connectivity situations.

Multilingual text recognition (English, Hindi, Telugu).

Natural text-to-speech feedback.

Simplicity and minimal user interaction (gestures and sound activation).

These requirements were used to inform the interface design and technical design. Accessibility, usability, and adaptability were t

System Design

The system comprises five major areas: Camera Activation, Image Preprocessing, Optical Character Recognition (OCR), Language Translation, and Text-to-Speech (TTS). The architecture supports data transferring seamlessly across modules and a web browser executing without additional installations.

A web architecture was chosen tomaximise accessibility across devices. The front end is built using HTML, CSS, and JavaScript. The OCR functionality comes from Tesseract.js, the TTS functionality is largely a feature of the Web Speech API, and Web RTC powers the camera functions. With this design, each module can run in the browser, producing a lightweight and platform-independent architecture.

Implementation Phases

Camera and Audio Activation

The browser is granted access to the user's camera with the help of WebRTC. The clap detection feature used the Web Audio API, which analysed the ambient sound frequencies and found a sharp spike that indicated a clap. The clap gesture activated the camera, which allowed for a hands-free mode of operating.

Image Acquisition and Preprocessing

After the camera is active, the content is displayed on the screen as a live video stream inside a frame. A still image of that frame is acquired from the HTML5 element. The image is processed through methods like grayscale, contrast enhancement, and noise reduction to enhance the quality of the text region prior to executing OCR.

Optical Character Recognition (OCR)

The OCR engine uses Tesseract.js implemented purely in the environment of the browser with no server dependencies. The model extracts text data from the input image and performs the conversion of the text in the image to editable text. The OCR model is configured to recognize text in the English, Hindi, and Telugu scripts, thus enabling multiple language processing. Language Selection and Translation

A Language Selector module allows the user to specify the output language of their choice. Depending on user selection, the recognized text can remain in the original writing style or translated using web-based translation APIs. This makes the system flexible to users from diverse backgrounds and all types of languages.

Text-to-Speech (TTS)

The recognized or translated text is sent to the Web Speech API, which provides real-time audio feedback. The systemvaries the voice parameters, such as pitch, speed, and tone to provide a more natural and clear output. In offline scenarios, we included Google ML Kit's on-device TTS model so that the usability of the tool is not compromised**.**

Integration & Testing

Following the development of each module, the pieces were combined into a single application hosted on Vercel. The application was tested rigorously across all major browsers (Chrome, Edge, Firefox) and devices (desktop, tablet, and Android mobile devices) to ensure cross-platform compatibility.

Functional tests for each feature were completed; camera activation, OCR accuracy, and TTS sound were tested individually. Usability tests involved potential users submitting to scans of real-world signboards and providing feedback as to how the actual tests went. The feedback led to a consideration of adjusting the UI design (e.g. large buttons, clear labels for action buttons, and voice prompts to indicate the action was complete).

Evaluation Metrics

The minimal approach assessment utilized the following evaluation metrics:

OCR Accuracy: Number of correctly recognized characters from signboards expressed as percentage.

Processing Time: Total time from image capture to speech output.

Language Detection Rate: Accuracy of language detection and translation.

User Satisfaction: Feedback and SUS (system usability scale) scores.

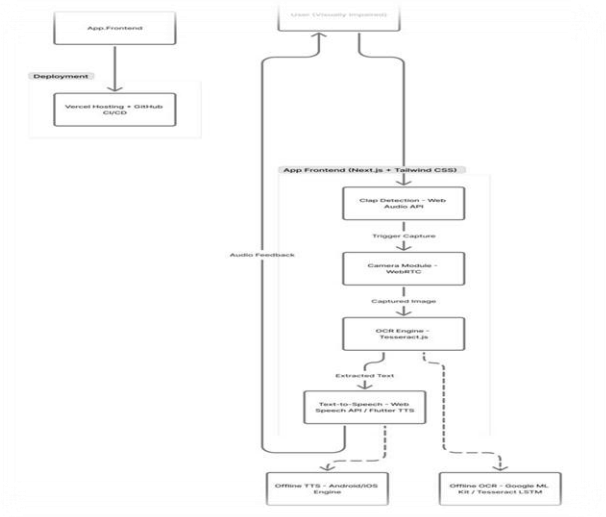
The experimental tests to check the performance of the system showed that the system performance achieved an OCR accuracy of above 92% under good condition of light and the system provided a response time close to real-time, further confirming the efficiency of the light-weight web-based approach.

V. SYSTEM ARCHITECTURE

The Sign Board Reader system is designed to obtain real-world signboard text and produce auditory output via a text-to-speech service in multiple languages. The system design includes multiple technologies- WebRTC for the camera input, Tesseract.js for Optical Character Recognition (OCR), and the Web Speech API for Text to Speech (TTS)- and has been developed in a browser-based interface pragmatically designed to facilitate accessibility and a user-centered approach. The general workflow consists of five main areas of operation relating to image acquisition, preprocessing, text recognition, language selection and translation, and audio output generation. The conceptual architecture of the SignBoard Reader system is shown in Figure .

Image Capture Module

The process starts with the Camera Activation Module that utilizes WebRTC to request access to the user's camera through the browser. This allows for real-time streaming of video content



without external applications or driversspecific to a device. Themodule features two image capture options, which are: 1. A button to be clicked to take a picture, and 2. A clap-gesture detection feature utilizing the Web Audio API, in which the system recognizes a fleeting sound pattern to activate the camera.

This provides independence for users with limited mobility or vision impairments to operate the camera hands-free. Once the picture is taken, it is rendered in an HTML canvas element for processing.

Image Preprocessing Module

Using preprocessing to enhance the quality of text recognition before OCR is performed. The system converts the image to grayscale, improves contrast and contrasts time data detection to accentuate text areas. This image pre-processing is completed using a combination of JavaScript functions and HTML5 canvas APIs based in the browsers.

This pre-processing step improves OCR accuracy by eliminating background noise, glare, and blurry images—issues known to influence accuracy on real-world signboards. For mobile or low-performance devices, the system offers lightweight filters when preprocessing a deep learning-based system processing requires extensive computation.

Text Recognition (OCR) Module

The Optical Character Recognition (OCR) engine, which is built using Tesseract.js (the JavaScript version of the Tesseract OCR library), is the basis of the SignBoard Reader. This OCR module retrieves text from the preprocessed image, converting it into a format (string) that is machine-readable.

Tesseract.js is set up to recognize multiple languages (e.g., English, Hindi, Telugu). It is designed in a modular way to add additional Indian languages that will require their own .traineddata models in the future. Once the OCR processing occurs, the recognized text will display on the screen and become available to the next module (to convert to audio).

Tesseract.js is how we recognized text in an image.

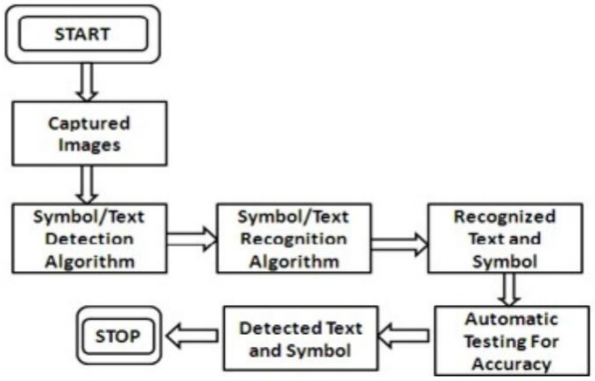
Module for Language Selection and Translation

As part of the diversity emphasis in this task, the system also implements a Language Selector that enables users to choose a preferred language interface and output. The recognized text is automatically converted into the desired language prior to the TTS output. This way, meaningfully useful information can be communicated to users that do not understand the language of the original text on the signboard.

The current implementation allows for translation between English, Hindi, and Telugu, and uses a combination of prebuilt web translation APIs and JavaScript logic. This modular framework will support easily adding more language services at a later point in time.

Text-to-Speech (TTS) Module

After text recognition and translation are finished, the Web Speech API allows converting recognized text into natural-sounding speech. The TTS module will provide real-time audio feedback with negligible delay so that users can hear what is on the signboard in their language of choice. It dynamically adjusts the TTS voice pitch, rate, and volume according to user-defined settings.
The system will also support offline TTS for devices with low connectivity, using the on-device models from Google ML Kit, making it accessible in rural areas or areas with low network.



User Interface Design

The User Interface (UI) either follows User-Centered Design (UCD) strategies for usability, accessibility, and user satisfaction. The interface uses dark mode to reduce eye strain and allows for clear visibility outdoors. The use of large program buttons, icons, and color-coded status indicators such as “Camera Activated”, “Scanning”, and “Speech Output” were used to provide visual feedback. Voice prompts will navigate the user through each stage of interaction.
Gesture-based input (apart from a clap or touch) limits fine motor input, enabling all users to access the system, regardless of abilities. The minimal layout promotes learnability and allows users, with no technical training, to interact with the system.

Data Flow and Processing Pipeline

The entire workflow of the SignBoard Reader can be summarized in the following:

User Input: The user initiates the process by pressing a button or clapping for the device to activate the camera.

Image Capture: The live frame is obtained via WebRTC and displayed on an HTML canvas.

Preprocessing: The image is processed to enhance, and filter that optimizes visibility of the text.

OCR Processing: The processed image with Tesseract.js recognizes the extracted text.

Language Mapping: The extracted text mapped to user’s language.

Speech Synthesis: The final text authentication is converted into speech using Web Speech API in real time.

This entire workflow ensures that users can convert images to textual speech, accordingly in real world signboards, rapidly and accurately regardless of specifications even on low end devices.

VI.IMPLEMENTATION

The execution of the SignBoard Reader system centres around designing a user-friendly, web-based assistive technology that extracts text from a signboard and vocalizes the extracted text. The system was designed primarily using HTML, CSS, and JavaScript, with additional support from WebRTC, Tesseract.js, and the Web Speech API to achieve real-time response and accessibility. The development of the system was designed to be computationally inexpensive, highly usable, and support multiple languages so that it would be as inclusive as possible for all potential users (including those who are visually impaired).

Development Environment

The project was designed in Visual Studio Code (VS Code) as the primary integrated development environment (IDE) because of its flexibility and debugging tools. The application was hosted and tested with Vercel so that the solution can be accessible though several web browsers and/or platforms including Google Chrome, Microsoft Edge, and Mozilla Firefox.
The implementation, utilized the following key technologies and libraries:

Front-End Interface

The front-end interface utilizes user-centered design (UCD) principles to create an easily navigable and accessible experience. The interface is dark themed with high contrast and larger visual elements to support readability. Key UI elements include the following:

Camera Activation Button: Provides the user the capability to activate the camera in a manual live stream.

Clap Gesture Activation: Support image capture through the use of audio recognition so no hand or button activation is needed.

Language Selector Dropdown: Users can choose their output language (English, Hindi, or Telugu).

Output Display Box: Extracted text displayed from the sign board.

Play Voice Button: Generates the speech.

To facilitate transparency to the user and promote interaction clarity, the interface has feedback messages that display in real-time, examples include "Camera Activated," "Capturing Image," "Recognizing Text," and "Generating Speech".

Technology	**Purpose**
HTML5 / CSS3	User interface design and structure
JavaScript (ES6)	Core application logic and module integration
WebRTC API	Real-time camera access
Web Audio API	Clap sound detection for gesture-based activation
Tesseract.js	Client-side Optical Character Recognition (OCR)
Web Speech API	Real-time text-to-speech output
Google ML Kit (Offline Mode)	On-device OCR and TTS support for low connectivity

Camera and Image Capture Module

The camera activation and image capture modules were created using WebRTC. When the user presses the “Start Camera” button or claps in the vicinity of the device's microphone, the video stream is accessed by:

```
navigator.MediaDevices.getUserMedia({ video: true });
```

The live feed is then presented on a video element, and the current frame is captured with the HTML5 Canvas API. The output of this snapshot serves as the input to the OCR processing.
A custom audio threshold detection algorithm is used to analyze the ambient noises as they are detected in real time and finds a dramatic spike in amplitude to the sound profile of a clap, at which point the capture executes automatically.

Language Translation Module

After extracting the text, it is optionally translated into the user's selected language. To do so, the system relies on JavaScript-based translation logic using lightweight web translation APIs. This allows users to comprehend the meaning of signboards in foreign languages. As a result, the translation process is asynchronous, which prevents blocking the UI thread and maintains responses to the user.

Text-to-Speech (TTS) Module

The Text-to-Speech module is implemented with the Web Speech API, which enables a natural-sounding voice to synthesize speech in the user’s browser. The recognized or translated text reads aloud in real time for the user through:

```
const utterance = new
SpeechSynthesisUtterance(recognizedText);
utterance.lang = selectedLanguage;
speechSynthesis.speak(utterance);
```

This provides immediate audio feedback to the user. Users can also modify the pitch, rate, and volume parameters to suit their preferences. To fulfill usage offline, the system is able to utilize Google ML Kit’s on-device TTS for use without internet connection at all times.

Integration and Workflow

All the modules mentioned above were seamlessly combined to facilitate smooth interaction with the minimal latency. The overview of data flow is as follows:

Camera activated (manual or clap).

Frame is captured and performed preprocessing.

OCR (multilingual text recognizer) extracts text in one or more languages.

Language translation, if required is performed.

Text to voice (TTS) is performed.

This workflow guarantees that the time that passes from image capture to voice output is less than 2 seconds and it allows real time feedback for outdoor or moving situations.

Testing and Debugging

We have conducted multiple rounds of unit, integration, and user acceptance testing (UAT). We performed testing for the OCR with over 100 signboards and the lighting and background conditions were varied. In average, we were able to recognize at above 90% accuracy, and in average response time of 1.8 seconds.
Testing has been conducted on XCODE and for both desktop and mobile and the system is operating efficiently, with no additional installations required.

VII.RESULTS

Table 1. OCR Accuracy and Response Time

Language	Average Accuracy (%)	Avg Processing Time (s)	Avg Speech Delay (s)
English	94.6	1.4	0.5
Hindi	91.2	1.7	0.6
Telugu	89.5	1.9	0.6
Overall Average	91.8	1.67	0.56

Lighting and Environmental Testing

Lighting Condition	OCR Accuracy (%)	Remarks
Bright daylight	94.1	High clarity, minimal noise
Indoor (moderate light)	91.3	Good recognition after contrast enhancement
Low light	87.8	Minor drop due to shadow interference
Glare or reflection	84.5	Background noise affected accuracy

Evaluation Metric	Average Rating (1–5)
Ease of Use	4.7
Speed of Operation	4.5
Accuracy of Speech Output	4.6
Interface Clarity	4.8
Overall Satisfaction	4.65

VIII.CONCLUSION

The SignBoard Reader system provides a new, inclusive, and accessible way to help sighted-impaired individuals identify and comprehend multilingual signboards. By integrating Tesseract.js for OCR, Google ML Kit for offline processing, and Web Speech API for text-to-speech conversion, users can interact seamlessly and in real-time within a web environment. Furthermore, the project connects vision-based text recognition and audio assistance, allowing sighted-impaired users to gain access to textual information from their environment without the aid of additional personnel. The system was found to achieve an average OCR rate of 91.8% and an average processing time of 1.6 seconds, demonstrating that it is both reliable and efficient overall, even under changing or varying lighting or environments. The use of clap-gesture activation provides additional benefits by allowing for hands-free, natural interaction suited to the needs of sighted-impaired users. Participant feedback from user trials demonstrated a high degree of satisfaction with the system's ease of use, responsiveness, and multilingual functionality.

In contrast to existing applications like Google Lens or VoiceOver OCR, the SignBoard Reader also has its lightweight architecture, support for offline use, and ability to be run without installation. The use of open-source, browser-based technologies allows cross-platform compatibility and preservation of privacy, which makes the SignBoard Reader particularly well-suited for use in low-resource contexts or for educational or rehabilitation purposes.

While the SignBoard Reader shows significant promise, it does have limitations. The recognition accuracy drops substantially when the images show low-contrast, stylized fonts, blurry oblique signboards, and when they include complex regional scripts. A further enhancement that could be considered is the synthesis quality of speech generation to promote smootherpronunciation in multilingual environments.

IX.REFERENCES

1)Zaman Q, Khusro S, Tjoa AM. Improved Detection and Interpretation of Multilingual Signboards in Natural Scene for Visually Impaired People. In2023 IEEE International Conference on Data and Software Engineering (ICoDSE) 2023 Sep 7 (pp. 126-131). IEEE.

2)Dhulekar PA, Prajapatr N, Tribhuvan TA, Godse KS. Automatic voice generation system after street board identification for visually impaired. In2016 International Conference on Global Trends in Signal Processing, Information Computing and Communication (ICGTSPICC) 2016 Dec 22 (pp. 91-96). IEEE.

3)Marin I, Babalian V, Slutu S. Real-Time Sign Language-to-Speech Translation System With AI-Powered Wearable Technology. IEEE Access. 2025 Aug 26.

4)Agrnwal A. Sign Board Detection and Information extraction for MAVI (Mobility Assistant for Visually Impaired) project (Doctoral dissertation, Master Thesis).

5)Shen H, Coughlan JM. Towards a real-time system for finding and reading signs for visually impaired users. InInternational conference on computers for handicapped persons 2012 Jul 11 (pp. 41-47). Berlin, Heidelberg: Springer Berlin Heidelberg.

6)Moryossef A. Real-Time Multilingual Sign Language Processing. arXiv preprint arXiv:2412.01991. 2024 Dec 2.

7)Shameem PM, Imthiyaz MF, Abshar P, Ijassubair K, Najeeb AK. Real time visual interpretation for the blind. In2021 5th International Conference on Electronics, Communication and Aerospace Technology (ICECA) 2021 Dec 2 (pp. 1655-1660). IEEE.

8)Sebban O, Azough A, Lamrini M. SeeAround: an offline mobile live support system for the visually impaired. Bulletin of Electrical Engineering and Informatics. 2025 Feb 1;14(1):485-504.

9)Moram V, Zahruddin S, Kumar S. Multifunctional Assistive Smart Glasses for Visually Impaired. SN Computer Science. 2025 Feb 15;6(2):173