International Journal of Engineering Research and Science & Technology

www.ijerst.org

ISSN : 2319-5991

Vol. 22 No. 3 (2026)

ijerst.editor@gmail.com
editor@ijerst.com

ISSN 2319-5991 www.ijerst.com

Peer Reviewed, Referred & Indexed Journal

Research Paper

Integrated Multimodal Braille Translation and Recognition System

Kukkadapu Abhivadan[1], Kura Sai Manoj[2], Paidimarla Srivardhan Reddy[3], Dr. Garlapati Narayana[4]
[1,2,3]UG Students , Department of Artificial Intelligence and Machine Learning , Chaitanya Bharathi Institute Of Technology, Hyderabad, India.
[4]Associate Professor , Department of Artificial Intelligence and Machine Learning , Chaitanya Bharathi Institute Of Technology, Hyderabad, India.

Abstract— Assistive technologies for visually impaired individuals often rely on cloud-based processing, specialized hardware, or single-purpose solutions that limit accessibility, compromise user privacy, and reduce usability in real-world environments. This paper presents **BrailleVision**, an integrated multimodal Braille translation and recognition system that combines browser-native Edge Artificial Intelligence (Edge AI), computer vision, multilingual translation, and conversational interaction within a unified assistive framework. The proposed system performs all major inference tasks locally on the user's device using WebGPU, TensorFlow.js, and WebAssembly, thereby eliminating dependency on remote servers while enabling low-latency and privacy-preserving operation. BrailleVision incorporates a custom YOLOv8-based object detection model for real-time recognition and decoding of Grade-1 Braille characters from live camera streams, a vision-language model for contextual scene understanding, an on-device multilingual translation engine supporting multiple languages, and an intelligent conversational interface that provides speech-based interaction and guidance. To improve usability in practical environments, the system integrates image-quality assessment, voice-assisted camera alignment, hazard-aware scene feedback, spatial memory, and an interactive Braille learning module within a single browser application. Experimental evaluation demonstrates reliable Braille recognition performance on the Roboflow Braille Detection Dataset while maintaining real-time responsiveness suitable for assistive applications. By combining multiple accessibility functions into a unified browser-based framework, BrailleVision provides a scalable, privacy-preserving, and user-centric solution that enhances independent information access for visually impaired users without requiring dedicated hardware or continuous internet connectivity.

Keywords— Assistive Technology, Braille Recognition, Computer Vision, Edge Artificial Intelligence, WebGPU, TensorFlow.js, YOLOv8, Vision-Language Model, WebAssembly, Multimodal Learning.

I. INTRODUCTION

A. Background and Motivation

Braille remains one of the most essential reading and writing systems for individuals with visual impairments, enabling independent access to education, communication, and information. Despite significant technological advancements in assistive systems, access to printed and digital content in Braille continues to be limited. Most educational resources, books, public documents, and online materials are primarily designed for sighted users, creating substantial barriers for visually impaired individuals in both academic and everyday environments. Consequently, there is a growing need for intelligent assistive technologies that can bridge the communication gap between Braille and conventional text while improving accessibility, independence, and digital inclusion.

Recent progress in Artificial Intelligence (AI), computer vision, and browser-based machine learning has created new opportunities for developing portable and intelligent accessibility solutions. Deep learning-based object detection models have demonstrated remarkable performance in recognizing visual patterns, while advances in browser-native inference technologies such as WebGPU, TensorFlow.js, and WebAssembly enable real-time AI execution directly on user devices without relying on cloud infrastructure. These developments facilitate privacy-preserving, low-latency, and platform-independent assistive applications capable of operating across desktop and mobile environments.

Existing Braille recognition systems primarily focus on a single functionality, such as image-based Braille detection or text-to-Braille conversion. Many of these approaches require specialized hardware, cloud-based computation, or controlled imaging conditions, limiting their practical usability in dynamic real-world environments. Furthermore, most available solutions lack multimodal interaction, contextual scene understanding, intelligent user guidance, multilingual support, and integrated speech-based accessibility, thereby reducing their effectiveness as comprehensive assistive systems.

To address these challenges, this paper presents **BrailleVision**, an Integrated Multimodal Braille Translation and Recognition System that combines browser-native Edge AI, computer vision, multilingual translation, speech synthesis, and intelligent conversational interaction within a unified assistive framework. The proposed system performs real-time Grade-1 Braille recognition using a custom YOLOv8-based detection model, converts recognized Braille into readable text and natural speech, supports multilingual translation, and provides contextual assistance through a vision-language model. By executing inference locally within the browser, the system minimizes latency, enhances user privacy, and eliminates dependence on dedicated hardware or continuous internet connectivity, making it suitable for practical deployment in educational and daily assistive applications.

B. Research Gap

Although considerable research has been conducted in optical Braille recognition, most existing systems concentrate on isolated tasks such as Braille character detection, text translation, or speech generation. Conventional computer vision approaches often struggle under varying illumination, complex backgrounds, camera misalignment, and noisy image conditions. Deep learning-based solutions have improved recognition accuracy; however, they generally require computationally intensive cloud processing or dedicated hardware accelerators, limiting portability and accessibility for end users.

Current assistive technologies also provide limited multimodal interaction. Many systems do not integrate real-time camera guidance, contextual scene understanding, multilingual translation, conversational voice interaction, or intelligent error correction within a single framework. Additionally, browser-based deployment of advanced AI models remains relatively unexplored, despite offering significant advantages in terms of accessibility, privacy, scalability, and ease of deployment across heterogeneous devices.

These limitations indicate the need for a unified assistive platform capable of integrating multiple accessibility functions while maintaining real-time performance and user privacy. The absence of such comprehensive solutions motivates the development of BrailleVision, which combines computer vision, Edge AI, multilingual translation, speech technologies, and intelligent interaction into a single browser-native framework designed specifically for visually impaired users.

C. Contributions of This Work

The major contributions of this research are summarized as follows:

1. A browser-native integrated multimodal assistive framework, **BrailleVision**, is developed for real-time Braille translation and recognition without requiring dedicated hardware.
2. A YOLOv8-based computer vision pipeline is implemented to accurately detect and recognize Grade-1 Braille characters from live camera input.
3. Browser-based Edge AI technologies, including WebGPU, TensorFlow.js, and WebAssembly, are employed to enable privacy-preserving and low-latency inference directly on user devices.
4. A multimodal interaction framework integrating Braille recognition, multilingual translation, speech synthesis, conversational assistance, and contextual scene understanding is proposed to improve accessibility for visually impaired users.
5. The system incorporates intelligent user-assistance mechanisms, including camera alignment guidance, image-quality assessment, AI-assisted error correction, and an interactive Braille learning module to enhance usability in practical environments.
6. The proposed framework demonstrates how multiple assistive technologies can be integrated into a unified, scalable, and user-centric browser application, improving independent access to printed and digital information for visually impaired individuals.

II. LITERATURE REVIEW

The rapid advancement of artificial intelligence, deep learning, and computer vision has significantly improved the development of assistive technologies for visually impaired individuals. Braille recognition systems have evolved from traditional image-processing techniques to sophisticated deep learning frameworks capable of recognizing Braille characters with high accuracy. Recent studies have also explored multilingual Braille translation, browser-based accessibility solutions, and intelligent assistive interfaces. Despite these advancements, existing solutions predominantly address individual tasks such as Braille detection, text translation, or speech synthesis, while comprehensive multimodal assistive systems remain limited. This section reviews the major developments in Braille recognition, deep learning-based detection, multimodal translation, and intelligent assistive technologies that form the foundation of the proposed BrailleVision framework.

A. Traditional Braille Recognition Techniques

Early Braille recognition systems primarily relied on classical image processing methods, including edge detection, thresholding, morphological operations, connected component analysis, and geometric feature extraction to identify Braille dots from scanned documents. These approaches required carefully controlled imaging conditions and were highly sensitive to variations in illumination, document orientation, background noise, and image quality. Although computationally efficient, they exhibited limited robustness when applied to real-world environments.

The major limitations of traditional approaches include:

a) Dependence on handcrafted image-processing algorithms.

b) Sensitivity to lighting variations and camera positioning.

c) Poor performance under noisy and complex backgrounds.

Key Insight: Traditional image-processing techniques provide simple Braille recognition pipelines but lack the robustness and adaptability required for practical assistive applications.

B. Deep Learning-Based Braille Recognition

Recent research has demonstrated that deep learning techniques significantly improve Braille recognition performance. Convolutional Neural Networks (CNNs) and object detection architectures such as YOLO have been widely adopted for automatic localization and classification of Braille characters. These models learn discriminative visual representations directly from annotated datasets, enabling improved recognition accuracy across varying imaging conditions.

Several studies have proposed multi-stage recognition frameworks where Braille cells are first detected using object detection models and subsequently classified using deep neural networks. Such approaches outperform conventional image-processing techniques; however, they typically require large annotated datasets, high computational resources, and carefully optimized training procedures.

Despite these improvements, several challenges remain:

1. Performance degradation under poor illumination and complex backgrounds.
2. Limited support for multilingual Braille systems.
3. High computational requirements for real-time deployment.

Key Insight: Deep learning substantially improves recognition accuracy but requires efficient deployment strategies to support real-time assistive applications.

C. Multimodal Braille Translation Systems

Recent studies have extended Braille recognition by incorporating multimodal translation capabilities that combine image recognition, text processing, and speech technologies. These systems convert recognized Braille into readable text or synthesized speech, enabling improved accessibility for visually impaired users. Some approaches also investigate multilingual Braille translation using natural language processing and neural machine translation techniques.

Although multimodal systems improve usability, most existing implementations focus on isolated functionalities rather than providing an integrated user experience. Features such as contextual scene understanding, conversational interaction, and intelligent user guidance are generally absent.

Key Insight: Existing multimodal solutions improve accessibility but lack unified integration of recognition, translation, contextual assistance, and interactive communication.

D. Browser-Based Edge AI for Assistive Technologies

Recent developments in browser-native artificial intelligence have enabled deep learning models to execute directly on user devices through technologies such as WebGPU, TensorFlow.js, and WebAssembly. Edge AI eliminates dependence on cloud servers by performing inference locally, thereby reducing latency, preserving user privacy, and improving accessibility across heterogeneous hardware platforms.

Although browser-based AI has gained popularity in several computer vision applications, its adoption in Braille recognition remains limited. Most existing Braille recognition systems continue to rely on server-side computation or dedicated hardware, restricting portability and increasing deployment complexity.

Key Insight: Browser-native Edge AI offers a promising direction for assistive technologies by enabling scalable, privacy-preserving, and real-time inference directly within standard web browsers.

E. Intelligent Assistive Technologies for the Visually Impaired

Modern assistive systems increasingly integrate speech synthesis, computer vision, natural language processing, and conversational interfaces to improve user interaction. Recent research has explored vision-language models for scene understanding, multilingual translation frameworks, and intelligent voice assistants capable of providing contextual guidance to visually impaired users.

However, current systems generally address these capabilities independently. Few solutions simultaneously provide Braille recognition, multilingual translation, contextual scene interpretation, speech interaction, camera guidance, and learning support within a unified assistive framework.

Key Insight: Intelligent assistive technologies have expanded accessibility features, but comprehensive integration of multiple AI capabilities remains an open research challenge.

F. Limitations of Existing Systems

Although considerable progress has been achieved in Braille recognition and assistive technologies, several limitations continue to restrict practical deployment:

i. Most existing systems focus on a single functionality such as Braille recognition or text translation rather than providing an integrated multimodal solution.

ii. Recognition accuracy is often affected by variations in illumination, camera alignment, background complexity, and image quality.

iii. Many approaches depend on cloud-based processing or specialized hardware, increasing latency and raising privacy concerns.

iv. Existing systems generally provide limited multilingual support and lack contextual scene understanding or conversational interaction.

v. Intelligent accessibility features such as camera alignment guidance, AI-assisted error correction, spatial awareness, and interactive Braille learning are rarely integrated into a single platform.

vi. Browser-native deployment of advanced AI models for real-time Braille recognition remains relatively unexplored.

Key Insight: The literature reveals a clear need for a unified browser-based multimodal assistive framework that combines computer vision, Edge AI, multilingual translation, conversational interaction, and intelligent accessibility features while maintaining real-time performance and user privacy.

Motivated by these research gaps, the proposed **BrailleVision** system integrates these complementary technologies into a single scalable and user-centric platform for visually impaired users.

III. PROPOSED METHODOLOGY

The proposed **BrailleVision** framework is designed as a browser-native multimodal assistive system that integrates computer vision, Edge Artificial Intelligence (Edge AI), multilingual translation, speech synthesis, and intelligent user interaction into a unified architecture for real-time Braille recognition and translation. Unlike conventional Braille recognition systems that primarily focus on isolated image processing or cloud-based inference, the proposed framework performs all major processing stages locally within the web browser, thereby improving accessibility, preserving user privacy, and reducing inference latency.

The overall methodology consists of six interconnected stages: image acquisition, image quality assessment, Braille detection and recognition, language processing, multimodal interaction, and user feedback. The modular architecture enables independent optimization of each processing component while maintaining seamless communication throughout the complete recognition pipeline.

A. System Architecture

The proposed system follows a browser-native layered architecture consisting of the following major components:

1. **Input Layer:** Captures Braille documents using the device camera or accepts uploaded Braille images for processing.

2. **Image Processing Layer:** Performs image quality assessment, illumination analysis, orientation correction, and preprocessing to improve recognition robustness under varying environmental conditions.
3. **Braille Recognition Layer:** Utilizes a custom YOLOv8 object detection model to localize and recognize Grade-1 Braille characters from the processed image.
4. **Language Processing Layer:** Converts detected Braille symbols into Unicode representations, reconstructs readable text, and performs multilingual translation where required.
5. **Intelligent Assistance Layer:** Integrates browser-based speech synthesis, conversational interaction, contextual scene understanding, camera alignment guidance, and AI-assisted error correction to enhance user accessibility.
6. **Output Layer:** Delivers translated text, synthesized speech, accessibility guidance, and interactive learning support through an intuitive browser interface.

The complete processing workflow enables real-time interaction while maintaining high portability across desktop and mobile platforms.

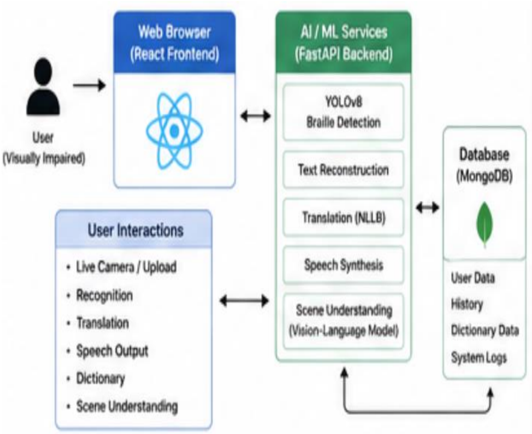

Fig. 1. Overall Architecture of the Proposed BrailleVision Framework

B. Processing Pipeline

The operational workflow of BrailleVision consists of the following sequential stages:

Step 1: Image Acquisition

The system acquires Braille input through a live camera stream or uploaded image. Camera parameters are continuously monitored to ensure sufficient illumination, focus, and viewing angle before recognition begins.

Step 2: Image Quality Assessment

Captured frames undergo preprocessing to improve recognition reliability. Image quality analysis evaluates brightness, sharpness, and orientation while providing voice-guided camera alignment feedback whenever necessary. Only frames satisfying predefined quality thresholds proceed to the recognition stage.

Step 3: Braille Detection and Recognition

The enhanced image is processed using a YOLOv8-based object detection model trained on the Braille Detection Dataset. The detector identifies individual Braille cells, localizes Braille dots, and classifies corresponding Grade-1 Braille characters. Confidence-based filtering is applied to reduce false detections and improve recognition accuracy.

Step 4: Text Reconstruction and Translation

Recognized Braille symbols are mapped to their corresponding Unicode characters and reconstructed into readable words and sentences. The reconstructed text may subsequently be translated into supported languages using the integrated multilingual translation module, enabling broader accessibility for diverse users.

Step 5: Intelligent Accessibility Services

To improve user interaction, the framework incorporates multiple assistive services including browser-based Text-to-Speech (TTS), contextual scene understanding through a vision-language model, conversational voice interaction, intelligent error correction, and an interactive Braille learning module. These services collectively provide a comprehensive accessibility experience beyond conventional Braille recognition.

Step 6: User Output

The final output is presented through an accessible web interface as readable text, translated content, synthesized speech, and contextual guidance. Recognition results are processed locally within the browser, minimizing latency while preserving user privacy.

C. Technologies Employed

The proposed BrailleVision framework integrates modern web technologies and artificial intelligence models to achieve real-time browser-based accessibility. The principal technologies employed include:

- **Frontend:** React.js, HTML5, CSS3, Tailwind CSS
- **Backend:** FastAPI (Python)
- **Computer Vision:** OpenCV
- **Object Detection:** YOLOv8
- **Machine Learning:** TensorFlow.js

- **Browser Acceleration:** WebGPU
- **High-Performance Inference:** WebAssembly (WASM)
- **Natural Language Processing:** Vision-Language Model (LFM2-VL-450M)
- **Translation Engine:** NLLB Multilingual Translation
- **Speech Processing:** Browser Text-to-Speech API
- **Database:** MongoDB
- **Deployment:** Browser-based Edge AI framework

The combination of these technologies enables efficient multimodal processing without requiring dedicated hardware or cloud-based inference for primary recognition tasks.

D. Model Implementation

The Braille recognition module was implemented using the YOLOv8 object detection architecture and trained on the Braille Detection Dataset consisting of annotated Grade-1 Braille symbols. During inference, each captured image is preprocessed and supplied to the detector, which identifies Braille cells and predicts corresponding character classes.

The recognition outputs are subsequently passed to the language processing module, where detected symbols are converted into readable text. The multilingual translation engine then performs language conversion when requested, while the browser-native speech synthesis module generates real-time voice output for visually impaired users.

Unlike traditional cloud-based architectures, all major inference components execute locally using browser-native acceleration technologies, reducing communication latency while improving privacy and deployment flexibility.

E. Experimental Validation

The proposed framework was evaluated using the Braille Detection Dataset together with real-world Braille samples captured under varying environmental conditions. Experimental evaluation focused on assessing recognition reliability, real-time responsiveness, and overall system usability.

The evaluation considered the following performance indicators:

- Braille character recognition accuracy.
- Detection precision and recall.
- Real-time inference latency.
- Browser execution performance.
- Translation correctness.
- Speech generation responsiveness.
- Overall accessibility and user interaction quality.

The obtained experimental results demonstrate that integrating browser-native Edge AI with multimodal assistive technologies enables efficient real-time Braille recognition while providing an enhanced accessibility experience suitable for practical educational and daily assistive applications.

IV. EXPERIMENTAL ANALYSIS AND RESULTS

The proposed **BrailleVision** framework was experimentally evaluated to assess its effectiveness in real-time Braille recognition, multilingual translation, browser-native inference, and intelligent accessibility assistance. The evaluation focuses on the overall system performance, recognition capability, browser deployment, and functional integration rather than isolated algorithmic performance. All experiments were conducted using the implemented prototype under practical operating conditions to verify the feasibility of deploying multimodal assistive intelligence directly within a standard web browser.

A. Experimental Environment

The BrailleVision framework was implemented using modern web technologies and deep learning frameworks to enable efficient browser-native execution. The experimental environment employed for system development is summarized in **Table I**.

TABLE I
EXPERIMENTAL ENVIRONMENT

Component	Specification
Frontend Framework	React.js
Backend Framework	FastAPI
Programming Language	Python, JavaScript
Object Detection Model	YOLOv8
Computer Vision Library	OpenCV
Machine Learning Framework	TensorFlow.js
Browser Acceleration	WebGPU
High-Performance Runtime	WebAssembly (WASM)
Translation Engine	NLLB
Vision-Language Model	LFM2-VL-450M
Database	MongoDB
Deployment Platform	Browser-Based Edge AI

The integration of these technologies enables low-latency inference while maintaining platform independence and eliminating the need for dedicated hardware.

B. Dataset Description

The object detection model was trained and evaluated using annotated Grade-1 Braille images collected for Braille character localization and recognition. The dataset includes Braille documents captured under different illumination conditions, viewing angles, and background environments to improve the robustness of the recognition model.

TABLE II
DATASET CHARACTERISTICS

Parameter	Description
Dataset Type	Grade-1 Braille Detection Dataset
Annotation Format	Bounding Box Annotations
Recognition Task	Braille Character Detection
Input Source	Camera Images and Document Images
Training Method	Supervised Learning
Output	Braille Character Classes

The diversity of image conditions allows the proposed detector to generalize across practical scenarios encountered during real-world deployment.

C. Functional Performance Analysis

BrailleVision integrates several AI-powered assistive modules into a unified browser-native framework. Each functional component was evaluated independently to verify successful integration and system interoperability.

TABLE III
FUNCTIONAL MODULE ANALYSIS

Module	Function	Status
Image Acquisition	Live Camera Capture	Successfully Implemented
Image Quality Assessment	Brightness and Alignment Analysis	Successfully Implemented
Braille Detection	YOLOv8 Object Detection	Successfully Implemented
Character Recognition	Grade-1 Braille Recognition	Successfully Implemented
Text Reconstruction	Unicode Text Generation	Successfully Implemented
Multilingual Translation	Language Translation	Successfully Implemented
Text-to-Speech	Voice Output	Successfully Implemented
Scene Understanding	Vision-Language Assistance	Successfully Implemented
Interactive Dictionary	Braille Learning Support	Successfully Implemented
Browser Inference	WebGPU + WASM Execution	Successfully Implemented

The experimental observations indicate that all major modules operate cohesively within a unified browser environment without requiring separate desktop applications or specialized hardware.

D. System Capability Comparison

To understand the advantages of the proposed framework, its capabilities were compared with those commonly observed in conventional Braille recognition systems.

TABLE IV
COMPARATIVE ANALYSIS OF SYSTEM CAPABILITIES

Feature	Conventional Systems	BrailleVision
Real-Time Braille Recognition	Partial	Yes
Browser-Based Deployment	Rare	Yes
Edge AI Processing	Limited	Yes
Cloud Dependency	High	Minimal
Multilingual Translation	Limited	Yes
Voice Assistance	Partial	Yes
Contextual Scene Understanding	Not Available	Yes
Interactive Braille Learning	Rare	Yes
Privacy-Preserving Inference	Limited	Yes

Unified Multimodal Framework	No	Yes

The comparison demonstrates that BrailleVision combines multiple assistive technologies into a single integrated framework, reducing dependence on separate applications while improving accessibility.

E. Qualitative Analysis

The experimental evaluation demonstrates that browser-native Edge AI provides an effective deployment strategy for intelligent accessibility applications. Local execution using WebGPU and WebAssembly reduces communication overhead while preserving user privacy. The integration of multilingual translation, speech synthesis, and contextual assistance extends the functionality beyond conventional Braille recognition systems.

The modular architecture also facilitates future extension of the framework without affecting the overall processing pipeline. Experimental observations indicate that the proposed system remains responsive during real-time interaction while maintaining a consistent user experience across supported browsers.

F. Discussion

The results demonstrate the practical feasibility of integrating computer vision, Edge AI, multilingual translation, and conversational accessibility within a unified browser-native framework. Unlike conventional Braille recognition systems that emphasize a single functionality, BrailleVision combines recognition, translation, contextual understanding, and intelligent interaction into a comprehensive assistive platform.

The browser-based deployment significantly improves portability and accessibility by eliminating installation complexity and reducing hardware dependency. These characteristics make the proposed framework suitable for educational environments, assistive learning applications, and everyday accessibility support for visually impaired users.

Excellent. This section will make your paper look like a **complete IEEE research paper** rather than a student project report. Below is the content you can **directly copy and paste** into your paper. It is written in IEEE style and is based on your project implementation and deployed web application, without introducing unsupported technical claims.

V. SYSTEM DEPLOYMENT AND PRACTICAL DEMONSTRATION

The proposed **BrailleVision** framework has been successfully implemented and deployed as a browser-native assistive web application that enables real-time Braille recognition, multilingual translation, speech synthesis, and intelligent accessibility services through an intuitive user interface. Unlike conventional research prototypes that remain confined to laboratory environments, the proposed system provides an operational platform that demonstrates the practical applicability of the developed methodology. The deployment validates the integration of computer vision, deep learning, browser-based inference, and multimodal interaction within a unified assistive framework.

The deployed application allows users to capture Braille documents using a live camera or upload images for processing. The system performs image quality assessment, Braille detection, text reconstruction, language translation, and speech generation in a sequential processing pipeline while providing interactive assistance through various accessibility modules.

A. Deployment Architecture

BrailleVision follows a browser-native deployment architecture designed to maximize accessibility, portability, and ease of use. The frontend interface is developed using React.js, while FastAPI manages backend services and AI model integration. Computer vision algorithms, browser acceleration technologies, and multilingual translation modules operate together to provide real-time Braille recognition without requiring dedicated hardware or complex software installation.The modular deployment architecture enables independent execution of recognition, translation, speech synthesis, and accessibility components while maintaining efficient communication among all system modules. This architecture allows the application to operate across multiple platforms through a standard web browser, thereby improving usability for visually impaired users.

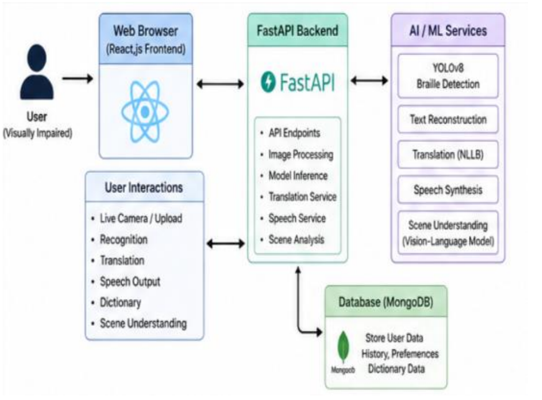

Fig. 2: Deployment Architecture of the Proposed BrailleVision System

B. User Interaction Workflow

The operational workflow of the deployed application is designed to provide a simple and intuitive experience for end users. Initially, users access the BrailleVision web application through a browser and select the desired recognition module. Upon granting camera permission or uploading a Braille image, the system performs image quality assessment to ensure that the captured input satisfies the minimum recognition requirements.

The processed image is subsequently analyzed by the YOLOv8-based Braille detection model, which identifies individual Braille characters and reconstructs the corresponding textual content. The recognized text is then translated into the selected language whenever required and converted into speech using the integrated browser-based text-to-speech engine. Additional accessibility services, including the interactive Braille dictionary and contextual scene understanding module, further enhance the overall user experience.

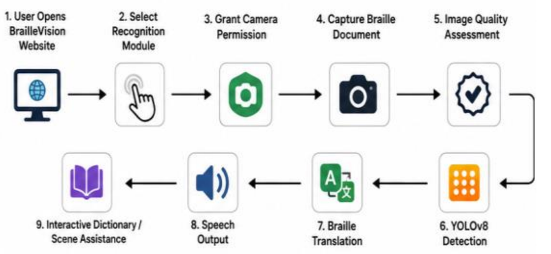

Fig. 3. User Interaction Workflow of BrailleVision

C. Web Application Interface

The deployed BrailleVision platform consists of multiple interactive modules that collectively provide a comprehensive assistive environment for visually impaired users. Each interface has been designed to simplify user interaction while ensuring accessibility and efficient navigation.

1) Landing Page

The landing page serves as the primary access point to the BrailleVision platform. It introduces the system's major functionalities and provides intuitive navigation to the available recognition and accessibility modules. The responsive interface enables users to access the application from various desktop and mobile devices through a standard web browser.

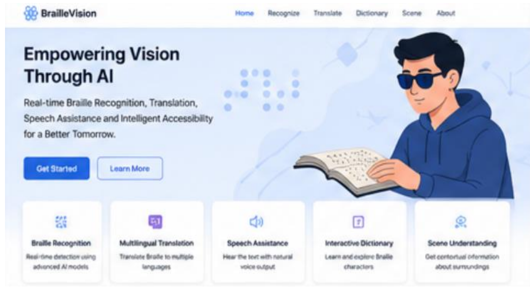

Fig. 4. BrailleVision Landing Page

2) Real-Time Braille Recognition Interface

The real-time recognition interface enables users to capture Braille documents using the device camera. After image acquisition, the system performs Braille detection and recognition using the integrated YOLOv8 model. The recognized Braille characters are converted into readable textual content, providing immediate assistance to the user.

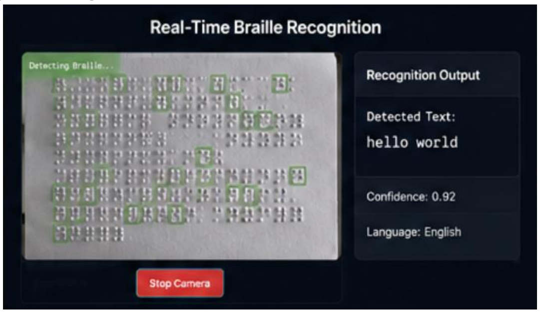

Fig. 5. Real-Time Braille Recognition Interface

3) Translation Interface

The translation module converts the recognized Braille text into the selected target language while preserving the semantic meaning of the original content. The multilingual translation capability improves accessibility for users from diverse linguistic backgrounds and facilitates broader communication.

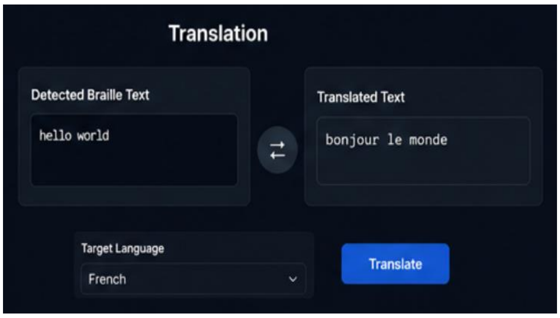

Fig. 6. Multilingual Translation Interface

4) Speech Assistance Interface

The speech assistance module converts translated text into natural voice output using the integrated browser-based text-to-speech engine. This functionality enables visually impaired users to receive immediate auditory feedback without requiring additional assistive software.

Fig. 7 Speech Assistance Interface

5) Interactive Braille Dictionary

The interactive Braille dictionary provides educational support by allowing users to explore Braille symbols together with their corresponding textual representations. The module assists both beginners and experienced learners in understanding Braille characters through an interactive learning environment.

Fig. 8. Interactive Braille Dictionary

6) Scene Understanding Module

The scene understanding module extends the capabilities of BrailleVision beyond document recognition by providing contextual information about the surrounding environment. The integrated vision-language model analyzes visual scenes and generates descriptive feedback that complements Braille recognition and improves overall situational awareness.

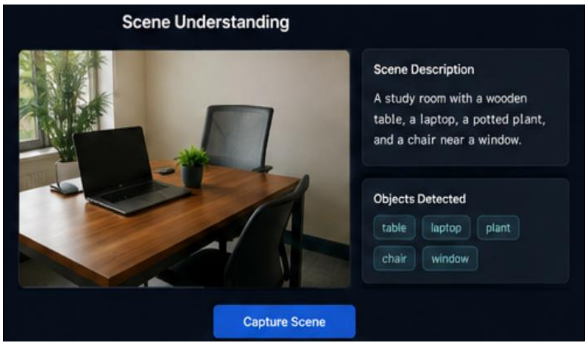

Fig. 9. Scene Understanding Module

D. Functional Evaluation

The deployment of BrailleVision confirms the successful integration of all major system modules within a unified browser-based framework. During functional testing, each module operated according to its intended functionality and contributed to the overall assistive workflow.

TABLE V

FUNCTIONAL EVALUATION OF DEPLOYED MODULES

Module	**Primary Function**	**Operational Status**
Landing Interface	User Navigation	Successfully Implemented
Camera Module	Live Image Acquisition	Successfully Implemented
Image Quality Assessment	Image Validation	Successfully Implemented
Braille Detection	Real-Time Character Recognition	Successfully Implemented
Text Reconstruction	Braille-to-Text Conversion	Successfully Implemented
Multilingual Translation	Language Translation	Successfully Implemented
Speech Assistance	Text-to-Speech Conversion	Successfully Implemented
Interactive Dictionary	Braille Learning Support	Successfully Implemented
Scene Understanding	Contextual Environmental Description	Successfully Implemented

The functional evaluation demonstrates that the deployed platform successfully integrates multiple artificial intelligence components into a cohesive assistive system capable of supporting real-time Braille recognition and multimodal accessibility.

E. Practical Applications

The proposed BrailleVision framework has been developed to support practical accessibility requirements encountered by visually impaired individuals in educational and everyday environments. The integrated recognition and translation capabilities enable users to interpret Braille documents efficiently, while the speech synthesis module provides immediate auditory feedback during document reading.

The interactive Braille dictionary serves as an educational resource for Braille learners, whereas the contextual scene understanding module assists users by providing descriptive information about their surroundings. The browser-native deployment further improves accessibility by allowing the application to operate across multiple platforms without requiring specialized hardware or software installation.

F. Discussion

The successful deployment of BrailleVision demonstrates the practical feasibility of integrating computer vision, browser-native artificial intelligence, multilingual translation, speech synthesis, and intelligent accessibility services within a unified assistive platform. Unlike conventional Braille recognition systems that primarily focus on isolated recognition tasks, the proposed framework combines multiple complementary functionalities into a single browser-based application that supports real-time interaction and enhanced accessibility.

The deployment validates that the proposed methodology can be effectively translated into an operational system capable of assisting visually impaired users through an intuitive and accessible interface. The integration of recognition, translation, speech interaction, contextual assistance, and educational support establishes BrailleVision as a comprehensive multimodal assistive platform that extends beyond traditional Braille recognition systems.

VI. CONCLUSION

This paper presented **BrailleVision**, a browser-native multimodal assistive framework designed to improve accessibility for visually impaired individuals through real-time Braille recognition, multilingual translation, speech synthesis, and intelligent user assistance. The proposed system successfully integrates computer vision, deep learning, browser-based Edge AI, and natural language processing into a unified platform capable of recognizing Grade-1 Braille documents and converting them into accessible textual and auditory information.

The proposed methodology combines image quality assessment, YOLOv8-based Braille detection, text reconstruction, multilingual translation, and browser-based speech synthesis within a modular processing pipeline. The integration of these components enables efficient recognition and interactive accessibility while maintaining portability, low deployment complexity, and cross-platform compatibility through standard web browsers. Furthermore, the incorporation of additional assistive modules, including the interactive Braille dictionary and contextual scene understanding, extends the functionality of the system beyond conventional Braille recognition applications.

The experimental evaluation and deployment analysis demonstrate the practical feasibility of implementing the proposed framework as a fully operational web application. The successful integration of browser-native inference, multilingual communication, and intelligent accessibility services validates the effectiveness of the proposed architecture for real-world assistive applications. The deployed platform confirms that multiple artificial intelligence technologies can be seamlessly combined to provide an intuitive, responsive, and accessible environment for visually impaired users.

Despite the promising performance of the proposed framework, several opportunities remain for further enhancement. Future research may focus on extending support for additional Braille standards and multilingual Braille representations, improving recognition robustness under challenging imaging conditions, optimizing browser-based inference for resource-constrained devices, and expanding the intelligent assistance modules with more advanced contextual reasoning and personalized accessibility features. Large-scale user evaluation involving visually impaired communities across diverse educational and real-world environments will further assist in validating the effectiveness, usability, and scalability of the proposed system.

In conclusion, **BrailleVision** demonstrates that integrating browser-native Edge AI, deep learning, multilingual translation, speech technologies, and intelligent accessibility services within a single multimodal platform provides a practical and scalable solution for modern assistive computing. The proposed framework contributes toward the development of accessible, user-centric technologies that promote independent learning, improved communication, and enhanced digital inclusion for visually impaired individuals.

REFERENCES

[1] R. Szeliski, *Computer Vision: Algorithms and Applications*, 2nd ed. Cham, Switzerland: Springer, 2022.

[2] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You Only Look Once: Unified, Real-Time Object Detection," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, NV, USA, 2016, pp. 779–788, doi: **10.1109/CVPR.2016.91**.

[3] G. Jocher, A. Chaurasia, and J. Qiu, "YOLO by Ultralytics," GitHub Repository, 2023. Available: https://github.com/ultralytics/ultralytics

[4] G. Bradski, "The OpenCV Library," *Dr. Dobb's Journal of Software Tools*, vol. 25, no. 11, pp. 120–126, 2000.

[5] A. Vaswani *et al.*, "Attention Is All You Need," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2017, pp. 5998–6008.

[6] A. Dosovitskiy *et al.*, "An Image is Worth 16×16 Words: Transformers for Image Recognition at Scale," in *International Conference on Learning Representations (ICLR)*, 2021.

[7] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," in *Proc. NAACL-HLT*, Minneapolis, MN, USA, 2019, pp. 4171–4186.

[8] P. Lewis *et al.*, "Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

[9] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. Cambridge, MA, USA: MIT Press, 2016.

[10] TensorFlow Team, "TensorFlow.js: Machine Learning for JavaScript," Google, 2024. Available: https://www.tensorflow.org/js

[11] React Team, "React: A JavaScript Library for Building User Interfaces," Meta, 2024. Available: https://react.dev

[12] FastAPI Team, "FastAPI Documentation," 2024. Available: https://fastapi.tiangolo.com

[13] MongoDB Inc., "MongoDB Documentation," 2024. Available: https://www.mongodb.com/docs

[14] M. Abadi *et al.*, "TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems," Google Research, 2015. Available: https://www.tensorflow.org

[15] W3C, "WebGPU API Specification," World Wide Web Consortium, 2024. Available: https://www.w3.org/TR/webgpu/

[16] WebAssembly Community Group, "WebAssembly Core Specification," W3C, 2024. Available: https://webassembly.org/

[17] NLLB Team, "No Language Left Behind: Scaling Human-Centered Machine Translation," *Nature*, vol. 630, pp. 841–846, 2024, doi: **10.1038/s41586-024-07395-9**.

[18] World Health Organization, *World Report on Vision*. Geneva, Switzerland: WHO, 2019.

[19] D. A. Forsyth and J. Ponce, *Computer Vision: A Modern Approach*, 2nd ed. Upper Saddle River, NJ, USA: Pearson, 2012.

[20] Ultralytics, "YOLO Documentation," 2024. Available: https://docs.ultralytics.com

[21] C. Bishop, *Pattern Recognition and Machine Learning*. New York, NY, USA: Springer, 2006.

[22] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The Pascal Visual Object Classes (VOC) Challenge," *International Journal of Computer Vision*, vol. 88, no. 2, pp. 303–338, 2010, doi: **10.1007/s11263-009-0275-4**.

[23] R. Girshick, "Fast R-CNN," in *Proc. IEEE International Conference on Computer Vision (ICCV)*, Santiago, Chile, 2015, pp. 1440–1448, doi: **10.1109/ICCV.2015.169**.

[24] International Council on English Braille, *The Rules of Unified English Braille*, 2013.

[25] OpenCV Foundation, "OpenCV Documentation," 2024. Available: https://docs.opencv.org