AI-Driven Voice Assistance for Visually Impaired Individuals by Automating Bengali Scene Text Detection and Audio Feedback

by

Maisha Iffat Chowdhury
22101497

A thesis submitted to the Department of Computer Science and Engineering
in partial fulfillment of the requirements for the degree of
B.Sc. in Computer Science and Engineering

Department of Computer Science and Engineering
Brac University
April 2026.

Declaration

It is hereby declared that

1. The thesis submitted is my/our own original work while completing degree at Brac University.
2. The thesis does not contain material previously published or written by a third party, except where this is appropriately cited through full and accurate referencing.
3. The thesis does not contain material which has been accepted, or submitted, for any other degree or diploma at a university or other institution.
4. We have acknowledged all main sources of help.

Student's Full Name & Signature:

Maisha Iffat Chowdhury
22101497

Approval

The thesis/project titled "AI-Driven Voice Assistance for Visually Impaired Individuals by Automating Bengali Scene Text Detection and Audio Feedback " submitted by

1. Maisha Iffat Chowdhury (22101497)

of Spring, 2025 has been accepted as satisfactory in partial fulfillment of the requirement for the degree of B.Sc. in Computer Science & Engineering on April, 2026.

Examining Committee:

Supervisor:
(Member)

Dr. Chowdhury Mofizur Rahman

Professor
Department of Computer Science and Engineering
Brac University

Thesis Coordinator:
(Member)

Dr. Md. Golam Rabiul Alam

Professor
Department of Computer Science and Engineering
Brac University

Head of Department:
(Chair)

Dr. Sadia Hamid Kazi

Chairperson and Associate Professor
Department of Computer Science and Engineering
Brac University

Abstract

Visually impaired individuals in Bangladesh face significant challenges navigating urban environments independently, where critical information — establishment names, addresses, and directions — is conveyed almost entirely through Bengali signboards. Despite recent advances in assistive technology, no lightweight, end-to-end system existed that could detect Bengali signboards, extract their text, and deliver it as spoken audio in real time. This work presents a five-stage pipeline that addresses this gap: a YOLO26s model detects signboards in street scene images, a second YOLO26s model isolates name and address regions of text interest (RTI), EasyOCR extracts raw Bengali text, the Gemini API corrects diacritical errors and typographical mistakes in the OCR output, and gTTS synthesizes the corrected text into natural Bengali speech. The system is trained on the publicly available SbNet dataset introduced by Mazumderet al. (2025) [49], achieving 97.14% mAP50 on signboard detection and 97.1% mAP50 on RTI detection — competitive with the state of the art despite using only 40% of the available training data. Qualitative evaluation demonstrates that the Gemini post-processing stage meaningfully improves raw OCR output, correctly resolving address separators, abbreviated place names, and character-level diacritical errors. The pipeline runs on a standard laptop CPU at inference, requiring no dedicated GPU for deployment. This research contributes a novel LLM-based correction layer for Bengali signboard OCR, a data efficiency benchmark for the YOLO26 architecture, and a complete assistive pipeline that extends prior detection-only work to full spoken output.

Keywords: Bengali Signboard Detection, Region of Text Interest, YOLO26, EasyOCR, Gemini API, LLM Post-Processing, Bengali Text-to-Speech, gTTS, Assistive Technology, Visually Impaired, Urban Navigation, Scene Text Recognition, SbNet Dataset.

Table of Contents

Chapter 1

Introduction

1.1 Background

In the contemporary era of intelligent systems and human-centered computing, enhancing accessibility for individuals with visual impairments continues to be a critical area of interdisciplinary research [22]. Although various smart navigation technologies have been introduced to support visually impaired users within structured environments such as homes, offices, and predefined pathways, public spaces—including subway stations, airports, railway terminals, and densely populated urban areas in developing countries such as Bangladesh—still lack comprehensive assistive infrastructures. This deficiency presents significant challenges for visually impaired individuals attempting to navigate these dynamic and often unstructured settings safely.

Bengali, the sixth most spoken language globally [42], faces a distinct technological shortcoming: the absence of robust and accessible scene-text recognition systems compared to those available for widely studied languages such as English and Chinese. Existing Optical Character Recognition (OCR) frameworks are predominantly optimized for clean, printed documents and are typically unable to maintain accuracy in uncontrolled, real-world conditions characterized by variations in font style, color, illumination, and background complexity. Furthermore, many of these models demonstrate limited performance in real-time recognition scenarios involving dynamic or cluttered environments [33].

To address these limitations, the present study proposes the development of a real-time Bengali scene-text recognition system capable of detecting and extracting textual information from signboards in public spaces and converting it into natural voice outputs. The objective is to provide reliable, context-aware auditory navigation assistance for visually impaired individuals, thereby enhancing their autonomy and safety in urban mobility contexts.

1.2 Rational of the Study or Motivation

A significant proportion of the Bengali-speaking population in Bangladesh and adjacent regions experiences visual impairments, which substantially limits their ability to interpret environmental signage [29]. Existing assistive technologies, includ-

ing GPS-based navigation systems, are predominantly designed for predefined and static environments, and consequently, they often fail to provide effective support in scenarios that require spontaneous, real-time recognition of textual cues in public spaces. Examples of such signage encompass exit indicators, platform numbers, train station identifiers, and directional boards.

Motivated by recent advancements in Bengali Optical Character Recognition (OCR) frameworks, such as bbOCR, and the development of multilingual scene-text recognition datasets, including Indic-STR12 [41], the present study seeks to integrate multiple artificial intelligence technologies—comprising scene-text recognition, text-to-speech (TTS) synthesis, and post-processing—into a unified, real-time assistive navigation system. This system is intended to facilitate the comprehension of environmental information for visually impaired users by converting textual content on signboards into audible guidance, thereby enhancing the navigability and safety of public spaces [22].

1.3 Problem Statement

Despite significant advancements in Optical Character Recognition (OCR) and AI-driven assistive technologies, there remains a lack of reliable and comprehensive systems capable of consistently recognizing Bengali signboard text in diverse real-world conditions and delivering it as Bengali audio in real time [9]. Current solutions are limited in their ability to manage the variability and complexity inherent in natural urban environments, rendering them impractical for deployment, particularly for the Bengali-speaking visually impaired population. Most conventional OCR systems are optimized for clean, structured, and printed text, and consequently exhibit poor performance under uncontrolled, real-world conditions. These systems frequently struggle to accommodate the wide range of font styles, sizes, orientations, and colors commonly observed on public signboards within transportation hubs, marketplaces, and urban streets.

Furthermore, many existing frameworks are restricted to the processing of static images and lack the capability to handle live video streams, which are essential for context-aware navigation in dynamic and fast-paced environments [36]. In such complex scenarios—such as navigating dimly lit train stations or crowded marketplaces—traditional vision-based systems often fail to accurately detect and interpret textual cues. This limitation disproportionately affects visually impaired individuals, who rely heavily on textual and spatial indicators to move safely and independently. Moreover, most commercial OCR and Text-to-Speech (TTS) systems primarily support globally dominant languages, such as English and Chinese, leaving low-resource languages like Bengali critically underserved. Given that Bengali is spoken by over 230 million people worldwide [43], this linguistic underrepresentation exacerbates accessibility gaps and constrains independent mobility for the Bengali-speaking visually impaired community.

In response to these challenges, the present research proposes an integrated system that combines advanced OCR, real-time video-based scene-text recognition, Bengali-optimized TTS, and adaptive post-processing. The system is designed to

extract text from complex, dynamic environments and convert it into intelligible Bengali audio in real time, thereby enhancing urban navigation, autonomy, and overall quality of life for visually impaired users [27].

1.4 Objective

The present research focuses on the development of a real-time assistive system based on Bengali Optical Character Recognition (OCR) models, specifically designed to facilitate navigation for visually impaired individuals in dynamic environments [29]. The system is structured around six primary functional objectives.

1. **Text Extraction:** The system is designed to identify and extract Bengali text from both static images and live video streams in real time. A central component of this task is the detection of community signboards in complex scenarios, including roads, intersections, and crowded commercial areas, where text frequently appears abruptly and under varying environmental conditions [13].

2. **Environmental Adaptability:** The system must function effectively across diverse real-world settings, such as subways, airports, railway stations, and busy urban streets, where challenges such as inconsistent lighting, motion blur, occlusions, and background noise are prevalent. Ensuring robustness and consistent performance under these uncontrolled conditions represents a key research challenge [17].

3. **Text-to-Speech Conversion:** Upon text extraction, the system converts the recognized Bengali text into natural, context-sensitive speech. The synthesized output is designed to maintain clarity and intelligibility even in noisy environments, thereby ensuring that visually impaired users can accurately comprehend the information conveyed [27].

4. **Real-Time Audio Feedback:** A primary advantage of this system is its capability to deliver immediate auditory feedback, enabling users to perceive spatial and directional information within seconds. Such real-time voice guidance enhances independent navigation and situational awareness for visually impaired individuals [18].

5. **Advanced OCR and Post-Processing Integration:** The system uses EasyOCR as the primary text extraction engine, working for both Bangla English. The OCR output will then go through an llm (Gemini) for post processing to achieve high recognition accuracy in real-world conditions. Additionally, a real-time TTS module ensures fluent and natural voice output.

6. **Cross-Platform Deployment:** The system is designed for seamless operation across multiple platforms, including mobile devices, embedded systems, and edge computing units (e.g., Android smartphones and Raspberry Pi), thereby ensuring scalability and accessibility in everyday environments [32].

Overall, this research aims to bridge the gap between contemporary artificial intelligence and machine learning advancements and the practical accessibility needs of

Bengali-speaking visually impaired users. The resulting system aspires to deliver a real-time, user-centered assistive solution that is accurate, responsive, and capable of providing contextually appropriate feedback to support independent mobility [34].

1.5 Methodology in Brief

The system is developed based on EasyOCR as the main optical character recognition engine with Bengali scene text and then a post-processing step based on Gemini API that removes diacritical errors, spacing anomalies, and misrecognized characters in the raw OCR result. The fixed text is then turned into natural audio Bengali output by a gTTS module.

The current study deals with the problem of navigation among the visually impaired people by creating a multi-stage artificial intelligence pipeline, which has the ability to identify and recognize Bengali scene text in images. The system is in such a way that physically challenged users are not required to walk up to signboards in order to understand the content posted on them.

The pipeline is run in a series of five steps. The initial step detects the signboards on an image of the input with the help of a YOLO26s object detector trained on full street-scene images, and locates Bengali signboards in large urban scenes. The second step undergoes a Region of Text Interest (RTI) detection in which a second YOLO26s model that has been trained on already cropped signboard images detects the name and address sub-regions of each signboard detected.

After the text regions are cropped, EasyOCR is used to extract character strings of raw Bengali characters in each region. The output of the raw OCR is subsequently sent to a Gemini API-based post-processing process that is used to fix diacritical errors, irregular spacing, and misrecognised characters while maintaining the original meaning of the text in the signboard.

To convert the corrected Bengali text to Text-to-Speech (TTS), the text is sent through the Google Text-to-Speech library (gTTS) that synthesizes natural Bengali audio. This audio will be the last output of the pipeline allowing the visually impaired user to read signboard contents without use of visual aids.

Deployment considerations include the implementation of the system on mobile and embedded platforms, such as Android devices and Raspberry Pi, ensuring portability, rapid operation, and real-time performance across diverse urban environments [46].

For evaluation, the system will be assessed using three core metrics: recognition accuracy, latency, and user satisfaction. Standard quantitative measures, including precision, recall, and F1-score, will be applied to benchmark performance against conventional standards. Additionally, trials with visually impaired users will provide qualitative feedback regarding usability, intelligibility, and overall effectiveness,

allowing iterative refinement of the user interface and interaction design.

In summary, this pipeline addresses the technological challenges inherent in Bengali scene-text recognition and TTS while maintaining a user-centered focus. The resulting system provides a practical, real-time assistive solution that enhances accessibility and mobility for visually impaired individuals in complex urban environments.

1.6 Scopes and Challenges

The main goal of the current study is to further the detection and recognition of Bengali text in signboard images in the real world by using AI-based assistive technologies, and to translate that data into the audio form of Bengali, thus allowing the visually impaired people to see and read the content of the signboards without the aid of sight.

Specifically, the project focuses on identifying Bengali text embedded within public signboards, irrespective of variations in lighting, font style, orientation, or background clutter. By supporting both pre-recorded images and live video streams, the system is designed to increase applicability and real-time performance, thereby broadening its potential use cases.

The subsequent stage in the text recognition process involves conversion into speech. This is accomplished through the integration of a Text-to-Speech (TTS) engine capable of translating Bengali sentences into natural, context-sensitive speech [1], [25]. Such functionality is essential for conveying directions, safety warnings, and other navigational cues, particularly in unfamiliar or complex environments.

Furthermore, the system is designed for cross-platform adaptability, operating efficiently on mobile devices, embedded systems, and edge computing units, including Raspberry Pi. Emphasis is placed on resource efficiency, ensuring functionality in low-cost or battery-constrained environments without compromising real-time performance or accessibility.

Despite the potential benefits of such a system, several technical and practical challenges must be addressed:

1. **Scene-Text Variability:** Public signboards exhibit substantial variation in font type, size, color, and placement. Additional complexities arise from occlusions, skewed angles, and visually noisy or cluttered backgrounds, all of which significantly increase the difficulty of performing accurate OCR in natural scenes [2].

2. **Speed and Latency:** Real-time recognition of scene text and subsequent speech synthesis necessitate low-latency processing while maintaining high accuracy. Achieving this balance is particularly challenging when operating on mobile or embedded devices with limited computational resources [5].

3. **Limited Datasets:** A critical limitation in the development of Bengali scene-text recognition systems is the scarcity of large, annotated datasets. This

shortage restricts the generalization capabilities of trained models and requires the application of data augmentation or transfer learning techniques to improve performance [37].

4. **Speech Clarity:** Generating intelligible and natural-sounding Bengali TTS output, especially for complex or compound words, remains a significant technical challenge. Ensuring clarity and contextual accuracy in the synthesized speech is essential for effective user comprehension [44].

5. **User-Centric Design:** Developing a system that is both functional and accessible necessitates careful consideration of user experience. The interface and interaction design must accommodate the diverse needs of visually impaired users and integrate seamlessly with assistive hardware and voice-controlled interfaces [14].

By systematically addressing these challenges, the present research aims to develop a robust and practical assistive system that advances the field of Bengali accessibility technologies and enhances independent navigation for visually impaired users.

Chapter 2

Literature Review

2.1 Preliminaries

Since the relatively recent introduction of Bangla Natural Language Processing (BNLP) to attempt to have computers with comprehension and creation capabilities of Bengali written and spoken forms. Although Bengali happens to be the seventh most spoken language in the world and is very instrumental in global linguistics, it is still miles behind the commonly studied languages such as English. Such deficiency is reflected in the size of computational models, annotated datasets, and other tools available in Bengali. The present BNLP research can be divided into two broad categories-text-based processing and speech-based processing. Optical Character Recognition (OCR) is an essential application on the text side, the printed, typed or handwritten Bengali text is converted to machine readable form. OCR normally divides into two subfields. Document-based OCR works with clean typeset pages, where scene-text OCR works with real-life messier text such as on road signs, shop names, or boards. OCR of scene-text is more difficult in that it is influenced by varying light, distorted reality and busy backgrounds. Equally, the importance is Handwritten Character Recognition (HCR). Due to the complexity of Bengali script, which has lots of compound characters and no recognizable capitalizations, handwritten recognition proves to be a challenging task, as compared to simpler systems. Hand writing styles are also very versatile and thus, their proper recognition is yet another difficult task. These complexities need to be solved by advanced machine learning methods. In the last couple of years, deep learning has achieved a massive advance in the field of OCR and HCR, and, as we learned, today Convolutional Neural Networks (CNNs) and the Bidirectional Long Short-Term Memory (BLSTM) networks are the tools to use when it comes to reading Bengali, whether printed or handwritten. These models nudge the accuracy with the help of delving into the archaic spatial and temporal patterns emerging within the data. Nevertheless, it is becoming a bit challenging to construct very strong Bengali NLP systems since Bengali trails with regard to large, high-quality linguistic data such as annotated corpora and various training materials, the very stuff that is required to construct scalable, high-performance models. There is no standardized, one-stop datasets to rely on and hence the BNLP tech is not able to generalize as well and even in real life situations.

2.2 Review of Existing Research

In a paper [26], Guezouli (2022) proposes a Reading Signboards System (RSbS) to enable visually impaired users to read text on signboards from camera video. RSbS uses Pseudo-Zernike Moments (PZMs) and stroke-width analysis in a three-stage pipeline: first, the system locates and segments the entire signboard region using PZM-based descriptors; next, it localises text regions within the cropped board via PZM clustering on the HSV Value channel; and finally, non-text components are filtered out by stroke-width transform (SWT) and morphological criteria. The method was evaluated on a custom database of over 80 low-resolution video frames depicting real signboards under varied lighting, perspective and textured backgrounds. In practice, each frame is divided into fixed-size blocks whose PZMs (computed for each RGB channel) are clustered by k-means into signboard vs background. The largest candidate cluster is refined via Canny edge detection and a fitted polygon to precisely crop the signboard. For text localisation, the cropped signboard is converted to HSV, and its V channel is similarly partitioned and PZM-clustered to yield text-like patches. SWT (per Epshtein et al.) is then applied to these patches: regions with inconsistent stroke widths or spacing (unlikely text) are discarded. This pipeline produces clean text-region crops intended for an OCR engine (e.g. Tesseract) and a text-to-speech API. Justification for PZMs is their orthogonality and invariance to noise, scale, rotation and perspective distortion, giving a compact, robust shape descriptor; SWT reliably characterises stroke-based text against background clutter. Detection performance is quantified per text-box (correct if 80% overlap with ground truth). On this data, RSbS achieves 0.861 precision and 0.723 recall (F1 0.785) in text-box localisation, indicating a strong balance of accuracy and coverage given the challenging inputs. This fulfils the goal of robust signboard reading under adverse capture conditions. Compared to standard orthogonal moments, PZMs matched Zernike moments in recall but ran faster (FPS24 vs 12) and clearly outperformed Hu moments (P=0.709, R=0.680). Against published methods, RSbS yields higher precision/F1: e.g. Wang et al. achieved (P 0.709, R 0.680, F1=0.694), Merler et al. (P=0.834, R=0.720, F1=0.773), and Liu et al. (P=0.600, R=0.800, F1=0.686), all below RSbS's F1. It also marginally exceeds a recent saliency-based detector by Zhang et al. (F1 0.781) on the same test data. These improvements stem from RSbS's robust signboard localisation (reducing false alarms) and the multi-scale invariance of PZMs. Notably, RSbS operates directly on low-resolution frames without specialised preprocessing. Overall, the evidence indicates that Guezouli's PZM/SWT-based RSbS approach is effective at accurately extracting signboard text in practice.

In [12], Debabrata Paul and Bidyut B. Chaudhuri put forward an OCR system for printed Bengali text, with some support for English, that uses a bidirectional LSTM (BLSTM) network and a connectionist temporal classification (CTC) output layer. The goal of the work is to get high recognition accuracy on full-line inputs without having to break the characters up into smaller pieces. There is one BLSTM layer with 128 units in each direction, and then there is a CTC layer with 166 output classes. These classes include Bengali vowels, consonants, symbols, English characters, punctuation, and a blank label. The authors exclude LSTM peephole connections and dropout, stating that these alterations diminish CTC loss and enhance

recognition performance. Xavier (Glorot) initialization is used to set the weights, and stochastic gradient descent with momentum (learning rate of 1×10 , momentum 0.9) is used for training. The system learns from 47,720 line images taken from a variety of printed sources, such as books, newspapers, letters, and notices, scanned at 300 dpi. These images show about 20 Bengali fonts. The inputs are grayscale images that have been height-normalized (48 pixels), and the ground-truth annotations are written in Unicode. Validation employs 1,500 lines, whereas evaluation is conducted on a reserved test set featuring 20 unique fonts. Training in TensorFlow can go on for up to 80 epochs, but it can stop early. The model's character accuracy (CA) is 99.32% and its word accuracy (WA) is 96.65%, as measured by the minimum edit distance. These results are better than those of basic OCR systems like Tesseract 4.0 (91.79% CA, 76.31% WA) and Google Drive Indic OCR (98.54% CA, 92.86% WA). The suggested architecture fixes common script-confusion mistakes in current tools and shows that a simpler, single-layer BLSTM-CTC framework can do a better job of recognizing Bengali text. This work is a big step forward for OCR research on Indic scripts because it shows that there is a better and more useful way to do things than the old ones.

In another paper [40], Imam Mohammad Zulkarnain et al. introduce bbOCR, an open-source, multi-domain OCR pipeline designed to convert scanned Bengali documents into structured, searchable digital text. The system is trained and evaluated on diverse synthetic and real data: two novel synthetic word-image corpora (Bengali SynthTIGER and SynthINDIC) for recognition, plus a multi-domain BaDLAD layout dataset and a new 228-image BCD3 set (9 document genres, 88.5K words) for evaluation. The pipeline uses specialised models at each stage: geometric distortions are corrected via the CNN-based PaperEdge model (Ma et al. 2022) for best unwarping accuracy, and illumination artefacts are removed by the DocTR illumination transformer (Feng et al. 2021) to improve visual quality. Document layout analysis is performed by YOLOv8, which identifies paragraphs, text boxes, images and tables, leveraging its state-of-the-art detection capability. Detected regions are then segmented into lines and words (using PaddlePaddle's DB-Net with ResNet50vd/MobileNetV3 backbones). Each word image is decoded by a novel APSIS-Net (CNN encoder + multi-head attention with positional alignment) into Unicode text. In component-level evaluation on BCD3, the YOLOv8 DLA achieves overall box mAP 0.61 (e.g. 0.72 on paragraphs) and the APSIS-Net recogniser attains 75% accuracy. System-level performance is measured by word-level (precision, recall, F1) and text-level (CER, WER) metrics. bbOCR substantially outperforms the open-source Tesseract baseline: on average its word-F1 0.46 vs 0.32 and CER/WER 0.59/0.80 vs 0.78/0.97, corresponding to 11.8% and 8.4% absolute reductions in CER and WER. These gains hold across all tested domains (bbOCR's F1 is higher in every category except one), indicating that the pipeline meets its digitisation goals and "is preferable over the current state-of-the-art" Bengali OCR systems. Overall, these results indicate that the bbOCR approach is effective.

Safir et al. proposed an end-to-end Optical Character Recognition (OCR) system aimed at recognising Bengali handwritten words directly from word-level images without segmenting individual characters [23]. To achieve this, they utilised the BanglaWriting dataset, which comprises 21,234 word images written by 260 indi-

viduals, from which 16,975 samples were used after preprocessing and filtering. The dataset was augmented using advanced image transformation techniques, including cutouts, noise, distortion, and affine transformations to improve robustness. The authors experimented with four CNN architectures—MobileNet, Xception, NASNet-Mobile, and DenseNet121—as feature extractors, and combined them with two types of bidirectional recurrent layers (LSTM and GRU), optimised using the Connectionist Temporal Classification (CTC) loss function to accommodate unsegmented sequence learning. DenseNet121 was chosen due to its dense connectivity and superior feature propagation, while GRU was selected for its lower computational cost and performance stability compared to LSTM. The models were evaluated using Character Error Rate (CER) and Word Error Rate (WER), the two standard metrics in OCR tasks. The best-performing configuration—DenseNet121 with GRU—achieved a test CER of 0.091 and WER of 0.273, significantly outperforming previous Bengali handwritten OCR systems, which often relied on rule-based or segmented pipelines. For comparison, traditional segmentation-based methods like those by Pramanik and Bag achieved around 94.01% word-level accuracy but on only 2000 samples [16], whereas this study used over eight times more data. Furthermore, the system's architecture ensured efficient training and inference using optimised FLOPs, with DenseNet121+GRU reaching only 6.4 million FLOPs while maintaining high accuracy [23]. These results confirmed that deeper CNN architectures with residual connections paired with GRU provided the best trade-off between performance and complexity. Compared to contemporary models like gated CNNs or encoder-decoder networks that reported higher CERs (e.g., >12%) on similar tasks [15] [21], the proposed system achieved state-of-the-art performance. Based on the results and methodology, the paper demonstrates that its approach to Bengali handwritten word recognition is currently one of the most effective solutions in the domain.

The paper [35] by Hasmot Ali, AKM Shahariar Azad Rabby, Md. Majedul Islam, A.K.M Mahamud, Nazmul Hasan, and Fuad Rahman focuses on enhancing Bangla Optical Character Recognition (OCR) systems by creating high-quality, multi-faceted datasets. The authors addressed the challenge of the lack of a sufficiently large dataset for Bangla OCR, which has hindered the development of efficient recognition systems. To overcome this, they compiled an extensive dataset consisting of over 4 million manually annotated images. This dataset includes a variety of document types such as computer-generated text, letterpress, typewritten, handwritten (both isolated and connected words), dynamic handwriting, and outdoor signs like posters and banners. The images, captured using high-resolution scans and camera-based imaging, were meticulously segmented and annotated through a combination of manual and semi-automatic methods. This comprehensive dataset is the largest of its kind and aims to serve as the gold standard for Bangla OCR systems.To assess their models, the authors employed the CRNN-VDS (Convolutional Recurrent Neural Network with Visual Deep Supervision) model, trained with 2 million synthetic samples. This model was specifically chosen for its strong performance in other OCR tasks and its suitability for recognizing sequential characters in complex scripts like Bangla. Performance was evaluated using Character Recognition Rate (CRR) and Word Recognition Rate (WRR). The CRR for the computer-composed subset was 93.04%, with a WRR of 79.03%. Letterpress documents achieved a CRR of 83.61% and WRR of 57.86%, while typewritten documents

performed the least, with a CRR of 70.60% and WRR of 28.05%. These results indicate that the model's effectiveness varies based on the document type and text complexity. Furthermore, inter-annotator reliability was high, with Fleiss Kappa scores of 0.91, 0.93, and 0.78 for computer-composed, letterpress, and typewritten documents, respectively. This research significantly contributes to Bangla OCR, offering the most comprehensive and reliable dataset to date, with strong validation quality and solid baseline outcomes. It serves as an important resource for developing OCR systems for low-resource languages such as Bangla.

The article [4] Bangla Text Extraction by Digital Image Processing, authored by A. K. M. Rashedul Hoque, Proma Mrittika Pandit, Najia Nasreen, and Hasin Raihan, under the guidance of Dr. Jia Uddin, focus on developing an Optical Character Recognition (OCR) system for the Bangla language. The complexity of Bangla, with its intricate and varied character designs, poses significant challenges for text recognition. The authors aimed to improve recognition accuracy to approximately 95%, enabling the conversion of scanned Bangla texts into editable formats.To achieve this, they used a custom-built dataset based on the Bangla Nikosh font. This dataset contained 340 characters, including 50 basic characters, 10 vowel modifiers, and 270 compound characters. They created approximately 3200 training data units by combining simple and compound characters with modifiers. Additionally, to enhance the system's functionality, they compiled a dictionary of about 1200 commonly used Bengali words. This comprehensive approach was designed to support the OCR system in recognizing and processing the complex structure of Bangla textThe primary technology used in the study was the Tesseract OCR engine (version 3.04). The authors chose Tesseract because it is open-source, has proven accuracy, and is extendable to non-Latin scripts. Additionally, Tesseract offers custom training tools, which enabled the authors to fine-tune the OCR engine for the Bangla script. Several tests were performed to evaluate the performance of the system, including tests on single-line images, multi-line images, scanned documents, newspaper clippings, and book covers. According to the authors, the word-level recognition accuracy reached around 95% for high-resolution, clean documents. The accuracy was assessed through visual inspection and textual comparison with the original image, focusing on error detection and character matching.The paper successfully achieved its goal, particularly in a controlled environment using clean images and familiar fonts. The evaluation parameters were based on the precision of character matching and the success of word-level extraction, with a recognition rate of approximately 9%. This result suggests that the system is viable for the digitalization of Bangla text in real-world applications.

In [11], Khalid Amirul Islam, Nadia Farha Mubin, and Saima Zaman propose a method for locating traffic signs in Bangladesh, translating them into Bangla (or English, if desired), and converting the translated text into audio instructions. The system's purpose is to stop automobile accidents that occurs when drivers don't pay enough attention to and follow traffic signs. The collection has roughly 9,000 pictures of 23 distinct kinds of traffic signs, including warning, educational, supplementary, and signal signs. The Bangladesh Road Transport Authority (BRTA) is the group that produced these signs. The dataset include pictures shot in multiple settings, such as varied perspectives, resolutions, and lighting, to make sure it is pow-

erful. The system employed a lot of various models and tools. We used OpenCV to retrieve the picture, transform it to black and white, and detect the edges. We used Python and NumPy to modify the pictures, and Support Vector Machine (SVM) to sort them and match them to templates. We utilized the PlaySound library to produce Bangla audio instructions for signs that were identified and translated. This helped drivers right away. The authors used SVM because it works well with datasets that aren't too vast and features that have a lot of dimensions. To detect road signs, the picture was broken up into pieces, and template matching was utilized to match shapes and edges. The initial testing with 2,000 samples revealed that the accuracy was 70%. The accuracy rose up to 90% when the training data was expanded to 4,000 samples, and it continued at that level even after the data was increased to 5,000 samples. Some more strategies that made the classification results even better include gamma correction, thresholding, block size modification, and gradient features. The algorithm could find and interpret traffic signs in the actual world with an accuracy of up to 90%. Comparative evaluations with prior research confirmed the system's effectiveness. The recommended strategy is a good way to make roads safer by offering drivers in Bangladesh clear and easy-to-find information regarding traffic signs.

Morshed Jaman Adnan, Abir Al Quraishi and Arifur Rahman in the paper [3] under the supervision of Abu Mohammad Hammad Ali, pay attention to the creation of the effective Text-to-Speech (TTS) synthesis system that will be used to work with the Bangla language. To achieve a working TTS system, the authors drew upon linguistic resources and also combined a variety of modules, such as text normalization, grapheme-to-phoneme (G2P) conversion, prosody modeling, and waveform synthesis, all running within the Festival speech synthesis system.To help with the implementation, the authors developed a phonetic dataset and a diphone dataset. They built a Bangla diphone database with specially constructed nonsense words in order to get all transitions between phonemes. These words were computerized, identified and transformed into the required pitch marks and LPC parameters. The system did employ a mixture of manually developed and automatically trained letter-to-sound (LTS) rules, along with existing lexicons and new entries.The primary technology was the Festival TTS framework, selected due to its flexibility and extensibility, and prior use in multilingual systems. Additional instruments that were employed were speech instruments such as SPTK and the Festival diphone schema. Since there are not many resources to develop the Bangali language, the authors decided to use the diphone synthesis technique, as this technique is the simplest, although it does not sound as natural as the unit selection synthesis. This approach enabled the authors to build a full end-to-end pipeline of the Bangla TTS and, therefore, it is feasible to use with languages with fewer resources such as BangaliRegarding performance, the paper has determined that the speech that was produced was not yet as clear as it could be but was leaning towards naturalness. The objective measures of success was not presented numerically; it was evaluated qualitatively. The system has been shown to be viable because it has been able to produce Bengali speech when the text input information is included in it and the task required to be performed involves processing of numbers, intonation and syntactic tokens.

Harsh Lunia, Ajoy Mondal and C. V. Jawahar introduce in [38] IndicSTR12, a

large scale dataset aimed at mitigating the imbalance in Scene Text Recognition (STR) in Indian languages. The dataset is based on 12 large Indic languages and seeks to improve the STR models by including real and synthetic word images in various scenarios like blur, occlusion, and perspective distortion. This combination enables models to generalize more to the complexities of multilingual scene text.Two categories of datasets were created. The real-world dataset consists of over 27,000 word images, of at least 1,000 samples of each language, including natural scenes like signage, banners, and advertisements, found by Google Image search. Synthetic dataset, on the other hand, consists of more than 3 million word images per language, created by rendering with various fonts, transformations, and colors. In order to benchmark IndicSTR12, three common STR models were tested, including PARSeq, a transformer-based model with Permutation Language Modeling (PLM); CRNN, a lightweight CNN-RNN-CTC-based model; and STARNet, a more advanced version of CRNN with spatial transformers and Inception-ResNet CNNs. The measures were Character Recognition Rate (CRR) and Word Recognition Rate (WRR). PARSeq has been shown to be more successful on synthetic data than STARNet and CRNN on all languages tested. To give an example, in Gujarati PARSeq obtained 95.25% WRR in comparison to 91.40% of STARNet and 81.85% of CRNN. On actual IndicSTR12 data, PARSeq once again obtained the lead, having 78.70% WRR in Punjabi and 71.94% in Telugu. The dataset was also more challenging than other multilingual benchmarks like MLT-19 and Urdu-Text, and is also more widely language-covered. In general, IndicSTR12 is a great milestone in improving STR in Indic languages. Combining massive amounts of synthetic data with real-world samples, the dataset is used to create powerful STR models that can manage various conditions in multilingual contexts.

The article [45] titled Enhancement of Bengali OCR by Specialised Models and Advanced Methodologies of Various Types of Documents, authored by AKM Shahariar Azad Rabby, Hasmot Ali, Md. Majedul Islam, Sheikh Abujar, and Fuad Rahman, focuses on creating a state-of-the-art Bengali OCR system. The goal was to develop an OCR that performs exceptionally well at recognizing text across various types of documents commonly encountered in the real world, such as computer-generated, letterpress, typewriter, and handwritten texts. Additionally, the system was designed to accurately restore document layout and handle graphical entities like images, tables, and signatures.The authors created a large, diverse dataset consisting of approximately 2.6 million images, which included four types of documents: computer-generated, typewriter, letterpress, and handwritten texts. The dataset was carefully annotated by over 100 annotators and supervisors, ensuring high-quality labels using an open-source annotation tool. In addition, they used a database containing 1 million samples of dynamic handwritten words, which contributed to enhancing the database's strength in terms of diverse writing styles and sources.For modeling, the authors employed a self-attentional, multi-headed neural network built on a VGG-based backbone. This network used a self-attention mechanism to capture character-level dependencies, making it particularly effective for Bengali, which features a complex script and compound characters. The system also included automatic perspective correction, character segmentation, and layout reconstruction modules, supported by a rule-based system. To enable deployment, the team used tools such as Apache Kafka, Zookeeper, ONNX, and NVIDIA Triton to

ensure compatibility across multiple platforms.The system's performance was evaluated using a confusion matrix and Levenshtein distance accuracy. The proposed system achieved an impressive mean accuracy of 98.05% (confusion matrix) and 87.20% (Levenshtein distance) on document types. In contrast, Tesseract, when tested on similar Bengali records, showed a mean accuracy of 57.56%. The system showed particularly high accuracy on computer-generated text, achieving 90.06% Levenshtein accuracy, surpassing existing models.

Sharmin et al. [47] introduced a detailed cross-platform proposal of the Bangali Optical Character Recognition (OCR) specifically developed to suit mobile applications. Their article, Bangla Optical Character Recognition on Mobile Platforms: A Multidisciplinary Cross-platform Solution, was an attempt to bring further the digitization of the Bangla text through the development of a safe, dependable and convenient cell phone app that can recognize both computer-printed and handwritten Bangla fonts.In the pursuit of this goal, the authors developed a special database specific to the Bangla script, including 340 basic and compound characters, as well as vowel and consonant modifiers. The dataset was a combination of computer-generated images (CGI) and scanned images (SI) so that the dataset is robust to different input conditions. It included changes in font type, font size, level of degradation and image resolution. The OCR engine was trained with 3,200 labeled image units and segmentation methods were used to represent real-world document structures and enhance the generalization of the model.The system was powered by the Tesseract OCR engine (version 4.0), and embedded into a Flutter-based mobile platform. The Dart programming language had core functions. Tesseract was chosen because it has demonstrated efficiency, flexibility, and scalability to the limited capabilities of computationally limited mobile devices. The model training followed a multi-pass recognition approach, which incorporates text segmentation, Unicode reordering, and post-processing phases, including spelling correction, to improve text accuracy.Results of the experiments showed that the system exhibited good performance in regards to a variety of evaluation measures. The accuracy rate obtained with the proposed application was 92 and 85 percent of printed and handwritten Bangla text respectively. Moreover, the system scored at about 9 out of 10, when tested with distorted or noisy images, which validates its robustness despite poor imaging conditions. Computational efficiency of the system was about 1.5 seconds per image on the standard mobile hardware.System feedback confirmed system effectiveness in terms of speed, accuracy, and usability. The research article concluded that the developed OCR application provides a viable, effective and convenient text recognition and digitization of Bangla texts on mobile devices and thus plays an important role in advancing the technology of language within the context of Bangali.

The article [49] Automated and Efficient Bangla Signboard Detection, Text Extraction, and Novel Categorization Method for Underrepresented Languages in Smart Cities, authored by Tanmoy Mazumder, Fariha Nusrat, Abu Bakar Siddique Mahi, Jolekha Begum Brishty, Rashik Rahman, and Tanjina Helaly, presents a comprehensive system for detecting Bangla signboards in natural scenes, extracting the associated text, and using Named Entity Recognition (NER) to categorize the institutions listed on those signboards. The authors aimed to enhance the use of underrepresented languages in smart city applications by enabling the digitization

and tagging of signage, making it more accessible in urban environments.The dataset used for the study was divided into two phases: Phase 1 consisted of full signboard images, while Phase 2 contained cropped Regions of Text Interest (RTI). Additionally, the authors compiled a large dataset of establishment names (42,548 names) across 10 categories, such as hospitals, restaurants, and schools, using web scraping and manual verification. The dataset covered various real-world challenges like low illumination, perspective distortion, and physical obstructions such as wires.For the object detection task, the authors employed YOLOv9-s in two stages: signboard detection and RTI detection. This model was selected for its balance between speed and accuracy, and its ability to adapt to edge devices through quantization. Tesseract OCR, enhanced with a custom binarization technique, was used for text extraction. To perform categorization, a Multilingual BERT model was fine-tuned for NER, allowing it to classify establishments by understanding contextual semantics without relying on rule-based methods.The evaluation process considered Average Precision (mAP), inference time, and accuracy. YOLOv9-s achieved impressive results, with 99.1% mAP for signboard detection and 97.7% mAP for RTI detection. The quantized model processed images in 21.2-21.4 ms per inference, making it suitable for real-time applications. The NER model based on BERT achieved a 92.89% accuracy in categorizing the establishment labels.

A deep learning-based end-to-end system to detect, recognise and parse Bangla address information contained in natural scene images of signboards and a survey of simple methods to deal with Bangla address information in natural scene imagery were presented by Hasan Murad and Mohammed Eunus Ali [39]. The research was expected to overcome the issues of complicated contexts, light changes, and the shortage of Bangla-language web resources, with the future prospects of automatic map labelling and navigation support for the visually impaired people. The authors created several data sets to provide the components of the system. They had prepared the Bn-SD dataset, composed of more than 16,000 annotated natural scene images of the signboard detection, and the Bn-AD dataset, composed of more than 8,000 cropped signboard images of address texts. Another one, Syn-Bn-OCR (used to train address recognition models) and Syn-Bn-AC (used to train a correction model) were created using a Raw Address Text Corpus in and around the city of Dhaka. Lastly, the Bn-AP dataset was designed to facilitate address parsing using address-specific tokens having address tokens. The authors applied YOLOv3, YOLOv4, and YOLOv5 models to detect the signboard and address region. The YOLOv4 delivered the best results in getting an average precision (AP) of 98.2 and 97.0 in signboard and address-region detections, respectively. In Bangla address recognition, they contrasted segmentation-free models: CTC-based models and Encoder-Decoder by using architectures, i.e. VGG, RCNN, ResNet and GRCL are made of Bi-LSTM units. The ResNet+Bi-LSTM+CTC model achieved the highest accuracy of recognition of 94.5% [18].In case of better performance of recognition, a transformer-based sequence-to-sequence correction model was introduced and trained on the Syn-Bn-AC dataset, which recorded 98.1% accuracy . To parse, they came up with RNN-based and transformer-based models. Transformer-based sequence-to-sequence transformer using BanglaBERT ranked first with 97.32 percent execution accuracy that superseded other sequence-to-sequence models, and its precision was 0.96, recall was 0.97, and F1-score was 0.965. The evaluation parameters

included accuracy, average precision (AP), and F1-score of each of the components. The intended system was able to achieve its goals by achieving high performance across all modules in the real-world environment. To sum up, the method applied by the authors is very effective. The deep learning pipeline is a reliable extraction and processing of Bangla addresses in complex natural scene images, and it adds a solid solution to the low-capacity language environment.

In the 2017 master thesis Increasing the Position Precision of a navigation Device using a camera-based landmark detection method [6], K. R. Jumani aimed at mitigating the residual drift of an inertial navigation system (INS) in a GPS-denied environment by augmenting with a front-facing camera which has learned pre-mapped road so called positioning signs. In training the landmark detector, Jumani used his personal positive samples of the target sign and retrieved various negative samples of the same on the internet, then resized them correctly and labelled them to train supervised learning. He used his discretion to select a Haar-cascade classifier, in the OpenCV/C++ (rejecting SIFT and SURF as too slow to run in real time in a mobile app), because cascade stages have built-in controls of high recall at video frame rates. The single-camera triangle-similarity estimate of range to the sign was based upon a known physical size of the sign to perform the calculation that was used. The trained detector was robust in systematic tests: it had high performance when the sign was very dark/bright and when the pose changed significantly and in multi-distance shots, showing invariance to light, orientation and scale. It gave an error of 3 cm (1-2% of the true range) only at a live distance test of 210 cm 210 cm 1 Tabulated precision-recall was not performed in the thesis, but the text asserts to have achieved what it calls an acceptable or good performance in all scenarios, meaning approximate 100% correct detection of targets in the trials. Through combining these vision-based landmark observations with INS information, the entire system was able to significantly reduce the positional error that would not have been possible to be removed by the extended Kalman filter alone. Thus, the task achieved its intended goals: it achieved real-time, illumination- and pose-invariant landmark recognition, centimetre-level distance estimation and a measurable increase of navigation accuracy under realistic driving conditions.

In 2017, Abdulmalik D. Mohammed, in his master's thesis, used the imagery of smartphones to avoid collisions and identify emergency exit signs to support autonomous navigation indoors [7]. In order to classify exit-signs, he compiled a 256 256 dataset of 175 phone-acquired photos (100 green exits, 75 non-exits); the performance of the detectors was then evaluated using the Oxford image collections that offer controlled variation in scale, rotation, viewpoint, illumination and blur, and the optical-flow algorithms were benchmarked against corresponding Middlebury non-rigid sequences, e.g. RubberWhale or Hydrangea, whose ground-truth flow is publicly available. Handset live corridor testing got 30 fps. Mohammed suggested two sparse detectors of features that are lightweight, Upright FAST Harris + BRIEF (U F a H B) and Scale Interpolated FAST Harris + BRIEF (S i F a H B), as well as a sparse optical flow algorithm named Nearest flow. U-FaHB used FAST corners and then Harris filter to choose strong upright corners; it processed through the scale-space to get scale invariance; it employed BRIEF descriptors to reduce computational overheads. Nearestflow matched U-FaHB key-points of one frame

to another with a 2-NN ratio test, compromising a bit of precision against speed, which makes sense in mobile hardware 1. There was a lot of satisfaction in meeting experimental results with design targets. When the histogram was concatenated to a naive-Bayes classifier was tested, 83% sensitivity and 90% specificity were found towards exit-sign recognition (geometric mean 86%), versus a k-NN baseline [20]. U-FaHB was the most repeatable detector of 12 standards during rotations below 30 degrees, and on scale transformations, it was a little better than pure FAST, and both were much better than other approaches; during blurring it again was leading by 1 percent over FAST. Precision-recall curve demonstrated that U-FaHB+ BRIEF matching is as resistant to viewpoint and illumination as BRISK and ORB, with the time to compute the descriptors: 0.09 ms against 0.04 ms of BRISK 1). During the Middlebury test, Nearest-Flow had a mean angular error of 1.6, which is comparable to LucasKanade, but with the lowest runtime when compared to tested techniques. Phone tests carried out in real time showed straightforward flow separation in the case of doors and floor panels but the estimates of time-to-contact were not always monotonic. As indicated, the reported metrics of classification accuracy, feature repeatability, precision-recall, descriptor speed and flow angular error are also used to show that the proposed flow algorithm and detectors can be used to provide high accuracy at very low computational cost showing that they are indeed suitable in mobile vision navigation tasks.

In Md. Sadrul Islam Toaha et al.'s article [31], an automatic signboard detection and localization solution with strong robustness and scalability in challenging urban use-case settings was proposed, as well as solutions that could be used in densely populated developing cities with signboard inconsistencies, background noise, and usefulness. They were spurred by a desire to improve establishment labeling in web-based services like Google Maps, Uber, and other navigation or mapping software that rely heavily on labeled physical signs to reliably classify locations of interest. To make model development and testing easier, the authors released Street View Signboard Objects (SVSO), a vast collection comprising 5,000 street photos from across the globe and 12,961 signboards annotated in 10 Bangladeshi cities. The dataset shows crowded visuals, occlusion, lighting patterns, and signboard structures typical to metropolitan environments in the Global South. The authors also created a second, very independent test sample of 200 images from six more developing countries, such as Malaysia, Bhutan, and Myanmar, and compared their model with the Billboard classification in the Open Images Dataset V6 (OIDv6) to make it even more globally representative. Their approach relies on R-CNN architecture with a Faster R-CNN model, which is accurate and can localize objects. However, taking into mind the drawbacks of popular approaches employed on non-homogeneous and visually chaotic datasets like the SVSO, the authors made many detection process alterations and enhancements. SB PreCNN, a domain-specific pre-training policy, used a secondary signboard vs. non-signboard-labeled image dataset to optimize the CNN backbone recognition process by processing significant characteristics; PreRCNN, which pretrained the Region Proposal Network (RPN) and Fast R-CNN modules to improve region processing and convergence; and ARAS (Anchor). These enhancements significantly improved performance in several assessment sets. With a mean Average Precision (mAP) of 0.91 in SVSO validation set, 0.90 in the independent test set, and 0.86 on OIDv6 benchmark, the system

outperformed top state-of-the-art detectors like YOLOv4 (0.88 mAP) and YOLOv5 (0.89 mAP) on the SVSO dataset. These findings confirm the model's accuracy and geographical and environmental generalization. In a trading experiment (low resource), they trained the model with 10% of the SVSO data and reported a competitive mAP of 0.84, demonstrating the system's durability and efficiency even with little training data.This work adds a lot to automatic urban marking, especially in situations with few resources and a lot of people. It also gives a system that can be used by many by using digital maps to help with unorganized data and poorly developed infrastructure.

Muhammad Aminur Rahaman in the direction of Prof. Dr. Md. Hasanuzzaman and Prof. Dr. Md. Haider Ali has developed a Bangla signs language recognition system (BdSLR) based on real-time computer vision in an attempt to fill in the communication gap between the hearing-impaired individuals and the normal population [8]. The thesis tried to convert the Bangla Sign Language (BdSL) into a text, including hand-sign division, classification, word identification, and sentence structure.To accomplish the same, Rahaman employed a large-scale dataset, consisting of 46 two-handed Bangla alphabet and numeral signs (Set-1), 38 one-handed alphabet signs (Set-2), and 30 elementary hand shapes (Set-3). The training set consisted of 11400 images (4600, 3800 and 3000 images in Set1, 2 and 3), whereas each set was evaluated with images of sixfolds under different environments, resulting in 68400 images. Other databases incorporated 3,600 photographs on 18 BdSL word-template and 31,200 pictures on 52 hand signs in a sentence-level recognition system.It was found that the thesis used Haar classifiers to detect region-of-interest (ROI) and a Fuzzy Rule-Based RGB (FRB-RGB) model to segment skin colour. To classify it had two feature descriptors, such as Normalised Outer Boundary Vector (NOBV) and Window-Grid Vector (WGV), and the matching criterion was Inter-Correlation Coefficient (ICC). An algorithm of Bangla Language Modelling (BLMA) was also introduced that transformed the sequence of hand signs into acceptable Bangla sentences. The reason these techniques were selected was their reliability to different lighting conditions, occlusion, and cluttered backgrounds as well as accuracy and computational costs, such as balancing parts.The great efficacy of the system was executed based on the performance evaluation. An average hand-sign accuracy of 95.67, 95.60 and 95.10 percent was obtained using Set-1, Set-2 and Set-3 datasets, respectively, with an average processing time of 8.01 ms/frame. As a BdSL word recognition trial, 90.11% accuracy at a word level was achieved with 26.063 ms/frame, whereas sentence-level recognition had 93.50% word recognition rate, 95.50% numeral recognition rate and 90.50% sentence recognition rate . ICC was the main measure to assess the performance in recognition across modules.Depending on the measuring criteria and the stable results in various test conditions, the current thesis introduces a highly efficient solution to the BdSL recognition problem. The system proposed here is predictable in real-time applications, and it is a major breakthrough in serving the hearing-impaired persons, which can be described as assistive technology for the Blind in Bangladesh.

In the domain of virtual assistants, early systems focused on specific interaction methods. A notable example is the digital assistant developed by Gupta et al., which aimed to create a hands-free personal assistant primarily for native Bangla

speakers, including those with physical disabilities [10]. The system was designed to recognize continuous Bangla voice commands and allow for mouse cursor control through facial movements. The primary goal was to provide an efficient and natural way to interact with a computer without relying on traditional input devices like a keyboard or a physical mouse. To achieve this, the researchers created their own dataset for training and testing the system. This dataset consisted of 240 audio files containing 12 distinct Bangla voice commands. The recordings were collected from five different speakers, whose ages ranged from 20 to 25, to capture some degree of natural variation in speech. To ensure the system's robustness, the voice commands were recorded in three different acoustic environments: a noiseless setting (a quiet room), a moderately noisy one (a classroom), and a highly noisy environment (a roadside). For each of the 12 commands, two reference signals were recorded from each person, resulting in a total of 24 reference signals per participant. The commands were simple, two to three-word phrases designed for common computer operations, such as "Open Notepad" or "Go to C-drive." For the voice recognition module, the core technology used was a cross-correlation technique. This method was chosen for its ability to measure the similarity between a live input voice command and the pre-recorded reference signals by comparing their energy levels. The system calculates the cross-correlation value, and if it surpasses a predefined threshold, the command is recognized and executed. For the face detection and mouse control feature, the system employed the Viola-Jones algorithm to initially detect the user's face, followed by the Kanade-Lucas-Tomasi (KLT) feature tracking algorithm to monitor head movements. These movements were then translated into on-screen cursor motion. The performance of the system was measured using accuracy and response time. The voice recognition component achieved an accuracy of 83% in both noiseless and moderate environments, and 75% in the noisy environment, with an average accuracy of 80.34% across all conditions. The system was able to respond to commands within 2-3 seconds. The face detection-based mouse control was active for 20-second intervals and performed well under consistent lighting, achieving a frame rate of up to 18.88 fps. The authors deemed the project successful, as it demonstrated a high accuracy rate and a reasonable response time, especially when compared to other models like a combined MFCC & DTW approach, which was slower and less accurate. However, the system is limited by its small set of 12 predefined commands and the short duration of the mouse control feature, indicating that while it is a successful proof-of-concept, its practical applicability is constrained.

Addressing a different user base, Sultan et al. developed 'Adrisya Sahayak', a desktop virtual assistant specifically designed for visually impaired, Bangla-speaking users [24]. The primary goal of this research was to mitigate the difficulties this demographic faces when using computers, peripheral devices, and common home appliances. The system was designed to provide a comprehensive suite of functionalities through Bangla voice commands, covering tasks such as sending and receiving emails, performing web and YouTube searches, voice typing in applications, navigating with maps, and controlling home automation devices like lights and fans. Unlike systems that rely on custom-built recognition models, 'Adrisya Sahayak' was built using the Google Web Speech API for both its speech-to-text and text-to-speech capabilities. The operational flow involved capturing the user's Bangla voice command, sending it to the Google API for transcription, and then matching

the resulting text against a local, customizable command database to recognize the user's intent. If a command was matched, the system executed the corresponding function. If no match was found, the query was automatically forwarded to a default web search engine. For its home automation feature, the system used a microcontroller connected to a Wi-Fi module, enabling wireless control over various household electronics. The system's performance was not evaluated on a static dataset but through a comprehensive user study. The evaluation involved 45 visually impaired participants who conducted a total of 1,185 trial cases across the system's seven main functional modules. The key performance metrics were system accuracy and response time. The results of this study were highly positive, demonstrating an average system accuracy of 93.3%. The accuracy for specific tasks varied, with YouTube searches performing best at 96% accuracy, while general search engine queries were the lowest at 88.5%. The average response time for the entire system was 2.28 seconds. Although the authors considered the project a success due to its high accuracy and broad functionality, this reliance on an external API makes the system entirely dependent on an active internet connection. Consequently, its performance is subject to degradation from environmental noise, internet speed, and any changes to the third-party API.

A more advanced, modular approach was taken by Ullah et al. for the 'Disha' system, a bilingual virtual assistant designed to operate in both Bengali and English, primarily for the banking sector in Bangladesh [48]. The central goal was to create a humanoid assistant capable of handling customer queries, conducting real-time financial transactions, and ultimately reducing the need for human agents and traditional IVR systems. This project also aimed to address the significant lack of sophisticated conversational AI systems for the Bengali language. The development of 'Disha' relied on a variety of diverse datasets for its different components. For the Automatic Speech Recognition (ASR) model, the team aggregated data from public sources like OpenSLR and a substantial, confidential dataset of 800 hours of call center audio from a telecommunications company. For the Text-to-Speech (TTS) component, they gathered data from Google's Bengali AI project and Bengali audio stories from YouTube. The Natural Language Understanding (NLU) model was trained on a large corpus of real-world user queries, including nearly 500,000 text queries from a major bank and additional data from the Land Ministry's call center, which was manually processed and annotated. The system's architecture was constructed as a multi-stage pipeline. For speech recognition, a Kaldi-based ASR model was employed. A significant challenge was the lack of a sophisticated phoneme dictionary for Bengali, which the researchers had to create manually. The core "brain" of the assistant was built using the open-source RASA framework, which was responsible for intent recognition and dialogue management. Due to the absence of advanced Bengali NLU models, the team used an English model from the spacyNLP library and fine-tuned it for their specific Bengali banking-related tasks. Finally, a Coqui-TTS model was used to synthesize the assistant's voice for output. The performance of the system was evaluated for each component. The Bengali ASR model achieved a Word Error Rate (WER) of 30.86% (equivalent to 69.14% accuracy), while the English model performed better with a 15.60% WER. The authors noted that the overall system represented a complete, end-to-end conversational agent. However, they also identified several limitations: the assistant

currently lacks emotional awareness in its responses, and the NLU model could be confused by inputs containing similar sequences of digits, such as PINs or account numbers. This architecture, while functional, highlights the challenges of building complex AI systems for low-resource languages, particularly the reliance on fine-tuning models from other languages

In the area of character recognition, researchers have tackled both scene text and handwritten characters. For scene text, the system proposed by Islam et al. provides an efficient method for extracting and recognizing Bangla text from natural images, with the goal of supporting applications like license plate reading, robot navigation, and aiding the visually impaired [28]. Their work addresses the unique challenges of the Bangla script, such as the connecting headline ('matra') and various character modifiers. To develop and validate their system, the researchers created a comprehensive dataset. This included 500 natural scene images captured from sources like posters, banners, and license plates, featuring diverse colors, writing styles, and orientations. From these images, they extracted and compiled a character database containing 31,000 individual character images, covering 62 different Bangla characters (digits, basic characters, and common conjuncts), with 500 examples for each. The system's methodology involved a multi-step pipeline. First, a novel algorithm was used to detect the Region of Interest (ROI) containing text. This process filtered non-text regions using morphological features like area and aspect ratio, along with horizontal and vertical projection profiles to remove areas with long, non-textual edges. Once words were segmented using a connected-component analysis, a second innovative algorithm was applied to extract individual characters. This algorithm was specifically designed to avoid the common problem of 'matra' removal, which can alter the identity of a character. Instead, it performed a vertical scan of the word, identifying the gaps between characters by locating columns with a minimum number of foreground pixels, effectively splitting the word without damaging the characters. For the final recognition step, a multi-class Support Vector Machine (SVM) classifier was trained using Histogram of Oriented Gradient (HOG) features. The performance of the system was rigorously evaluated. The ROI extraction algorithm achieved an accuracy of 92.70%, and the character extraction algorithm achieved 93.23%. The final character recognition stage was highly successful, reaching an average accuracy of 99.16%. The authors also compared their SVM-based approach to a Convolutional Neural Network (CNN), which only achieved 83.52% on the same task, validating their choice of model. The primary limitation acknowledged is the algorithm's failure to extract characters that are physically connected (other than by the matra) or written in a significantly curved or circular shape.

In a 2021 study, Ghosh et al. conducted a comprehensive performance analysis of state-of-the-art, pre-trained Convolutional Neural Network (CNN) architectures for the task of Bangla handwritten character recognition [20]. The primary goal of their research was to address the significant challenges posed by the Bangla script, particularly its vast number of complex and structurally similar compound characters, which had limited the efficacy of previous recognition systems. The authors aimed to identify which modern deep learning models could achieve the highest accuracy on a large-scale character set and to evaluate the practical trade-offs between performance and computational cost. To facilitate this large-scale evaluation,

the researchers utilized the CMATERdb dataset, a well-established repository for Bangla characters. They compiled a comprehensive dataset by merging its basic, numeric, and compound character databases, resulting in a total of 231 distinct character classes for their classification task. Before feeding the data into the models, the images underwent a preprocessing stage where they were converted into black and white with a white foreground and resized to a uniform 28x28 pixel format. The study investigated a wide array of prominent pre-trained CNNs, including InceptionV3, ResNet50, DenseNet121, VGG16, and InceptionResNetV2, among others. The rationale for using pre-trained models was to leverage the powerful feature-extraction capabilities they gained from being trained on massive datasets like ImageNet, thereby reducing training time and improving performance on the specialized task of character recognition. The performance of each model was rigorously measured using accuracy, average precision, and average recall as the primary metrics. After training for 50 epochs, the InceptionResNetV2 architecture emerged as the most successful single model, achieving a remarkable recognition accuracy of 96.99%. Other architectures like DenseNet121 and InceptionV3 also demonstrated strong performance, with accuracies of 96.55% and 96.20%, respectively. The authors further experimented with an ensemble approach, combining the top three models through a voting algorithm. This combined model yielded an even higher accuracy of 97.69%. However, it was ultimately deemed infeasible for practical applications due to its substantial computational and memory demands. This paper is significant because it provides a thorough benchmark for a complex script, concluding that InceptionResNetV2 offers the best compromise between high accuracy and operational feasibility for developing robust, real-world Bangla optical character recognition (OCR) systems.

In a similar vein, Das et al. (2021) sought to develop an effective method for Bangla handwritten character recognition by proposing a custom, extended Convolutional Neural Network (CNN) [19]. The primary goal of their work was to leverage a deep learning model that could automatically extract features from character images, thereby bypassing the challenges and complexities of traditional, explicit feature extraction techniques. The authors aimed to build a classifier that could accurately recognize the basic characters of the Bangla alphabet, including numerals, vowels, and consonants. For their experiment, the researchers utilized the "BanglaLekha-Isolated" dataset. From this collection, they selected a subset of 60 distinct character classes, comprising 10 digits, 11 vowels, and 39 consonants. It is important to note that this work explicitly excluded the more complex compound characters, which were identified as a subject for future research. The core of their methodology was a bespoke CNN architecture which they termed an "extended" model. This network consisted of three sequential convolutional blocks, each containing a convolutional layer, a batch normalization layer, a ReLU activation function, and a max-pooling layer. This structure was designed to process preprocessed 32x32 pixel images and feed the learned features into a fully connected network for classification. The choice of a CNN was driven by its proven ability to learn hierarchical patterns directly from pixel data, making it well-suited for the variations in shape, size, and style inherent in handwritten text. The model's performance was evaluated based on classification accuracy. After training for 10 epochs, the network achieved a very high validation accuracy of 99.50% for digits. For the alphabetical characters, the accuracy was

93.18% for vowels and 90.00% for consonants. The combined validation accuracy across all 60 character classes was 92.25%. While the proposed custom CNN demonstrated strong performance on the basic character set, its primary limitation lies in its scope. By intentionally omitting the compound characters, the study does not address one of the most significant hurdles in developing a comprehensive Bangla OCR system. Thus, the paper serves as a strong proof-of-concept for basic character recognition but underscores the remaining challenge of handling the full complexity of the script.

Providing a broad overview of the field, the survey by Sen et al. (2022) presents a comprehensive review of Bangla Natural Language Processing (BNLP) [30]. The primary goal of this extensive work was to systematically map the landscape of BNLP research by analyzing a significant body of literature to identify progress, prevalent methodologies, and persistent challenges. The authors aimed to fill a notable gap by creating a centralized resource that offers a holistic and structured view of the diverse tools and techniques developed for the Bangla language over two decades. To achieve this, the authors conducted a thorough analysis of 75 research papers sourced from major academic databases, published between 1999 and 2021. Their review was methodically organized, categorizing the surveyed works into eleven distinct sub-domains of BNLP. These included key areas such as Information Extraction, Machine Translation, Named Entity Recognition, Sentiment Analysis, Speech Processing, and Text Summarization. The paper meticulously examines the evolution of techniques within these areas, charting the clear transition from classical, rule-based, and statistical methods (like HMM and N-gram models) to the more recent adoption of machine learning and deep learning approaches (such as SVM, CNN, LSTM, and GRU models). This methodological analysis effectively illustrates the field's technological trajectory over time. A critical contribution of this survey is its cogent synthesis of the major challenges that continue to impede progress across the BNLP domain. The review identifies several recurring obstacles that provide essential context for understanding the efforts of other researchers. Key among these is the profound lack of large-scale, publicly accessible, and professionally annotated corpora. This scarcity often compels researchers to develop their own small-scale datasets, which hinders the reproducibility and comparative evaluation of new models. Furthermore, Sen et al. highlight the absence of standardized, modern linguistic tools (e.g., robust parsers and POS taggers) and the inherent grammatical and morphological complexity of the Bangla language itself as significant, persistent hurdles. By collating these challenges, the survey serves as an invaluable meta-analysis rather than a proposal of a new technique. This review is a crucial reference, contextualizing individual research efforts and clearly outlining the foundational issues that the BNLP community must collectively address to advance the field.

2.3 Summary of Key Findings

The literature surveyed spans multiple key areas that are essential for building an AI-powered voice assistant designed to read Bengali signboards aloud for visually impaired users. The central focus lies in Bengali OCR development, scene-text de-

tection, assistive applications, and voice synthesis.

Bengali OCR Technologies

The foundation of such assistive systems lies in the development of robust OCR engines capable of reading printed, handwritten, and scene text in Bengali. A number of studies explored models such as Tesseract, BLSTM-CTC, APSIS-Net, and CNN–RNN combinations to improve recognition under noisy and complex conditions. Among them:

- Paul & Chaudhuri [12] proposed a BLSTM-CTC model that performed significantly better than traditional engines like Tesseract and Google OCR. It achieved 99.32% character accuracy and 96.65% word-level accuracy, particularly effective as it did not require character segmentation.
- Safir et al. [23] focused on handwritten Bengali text, achieving state-of-the-art accuracy using a DenseNet121 + GRU model. Their system reached a CER of 0.091 and WER of 0.273, highlighting the effectiveness of deep learning for variable handwriting styles.

Scene-Text Recognition & Signboard Detection

Recognizing Bengali text in natural scenes is especially relevant for real-world use. Various detection techniques were used, including YOLO, Pseudo-Zernike Moments (PZM), Stroke-Width Transform (SWT), and SVMs. Key contributions:

- Guezouli (2022) [26] introduced the RSbS framework, using PZM and SWT for signboard segmentation, achieving an F1 score of 0.785, robust under poor lighting and background noise.
- Mazumder et al. [49] proposed an R-CNN-based signboard detector, reporting mean Average Precision (mAP) of 0.91, with high generalizability across different countries.
- Zulkarnain et al. [40] developed bbOCR, a flexible pipeline that significantly reduced both WET and CER (by 8–12%) compared to Tesseract, across various datasets.

Assistive Applications for Visually Impaired Users

Research in assistive technology demonstrates real-world deployments of OCR and TTS:

- Islam et al. [11] built a system to convert road signs into speech, achieving 90% real-world accuracy—helpful in preventing accidents and improving independent navigation.
- Sultan et al. [24] launched Adrisya Sahayak, a voice-based assistant for blind Bengali users, attaining 93.3% system accuracy, supporting natural interaction.

- Murad & Ali [39] used YOLOv4 + transformer models to extract address information from urban scenes in Bangla, showing high precision useful for smart city navigation.
- Mazumder et al. [49] also implemented a NER model to identify institutions from signboard texts, scoring 92.89% accuracy, proving NLP's potential in complex outdoor environments.

Text-to-Speech (TTS) Advances

TTS is vital for converting recognized text into meaningful speech. Two notable projects include:

- Adnan et al. [3] adapted the Festival framework using diphone synthesis. While functional, the system lacked naturalness in speech delivery.
- Ullah et al. [48] employed Coqui-TTS in a bilingual setup for the banking industry, demonstrating that conversational Bengali TTS is feasible in customer-facing environments.

Datasets and Benchmarking

Training and testing Bengali OCR models depends heavily on comprehensive datasets. Key datasets include:

- BanglaWriting: for handwritten recognition.
- SVSO & BaDLAD: scene-text recognition.
- SynthTIGER: for generating synthetic training data.

These datasets enable standardized evaluation of OCR engines across different domains—printed, handwritten, and scene-based.

Key Gaps and Unresolved Challenges

Despite meaningful progress, several limitations still prevent the deployment of real-time, assistive Bengali OCR-TTS systems:

- Lack of Real-Time Scene OCR: Most systems are too heavy for low-power or mobile devices, limiting usability in real-world scenarios.
- No Context-Aware Navigation: OCR systems generally stop at text recognition. They fail to translate visual information into actionable voice instructions (e.g., interpreting a "Caution" sign differently from a "Turn Left" sign).
- High Computational Demands: Many high-performing models are not optimized for speed or resource efficiency, making real-time application difficult.
- Fragmented System Design: Most current systems treat sign detection, OCR, and TTS separately. Lack of an integrated end-to-end pipeline hinders seamless user experience.

- TTS Naturalness: Synthesized speech often lacks expressiveness and clarity, especially in dynamic or noisy settings. Prosody modeling and expressiveness need improvement for effective communication.
- Insufficient Real-World Datasets: There's a serious shortage of datasets containing real-world Bengali signboards, especially those with low resolution, occlusion, and urban clutter.

Summary and Relevance to Proposed Work

The reviewed studies establish a strong technical groundwork for the development of assistive technologies in the Bengali language. Significant headway has been made in printed and handwritten OCR, and several models now demonstrate high accuracy in scene-text detection as well. However, no system has yet offered a lightweight, real-time, and fully integrated solution capable of detecting, reading, interpreting, and vocalizing Bengali text in real-world, mobile settings. This gap is exactly what the proposed thesis aims to address. By integrating models like bbOCR with real-time video input, and using a TTS framework like Coqui-TTS, the proposed system would deliver contextual, voice-guided feedback to blind users navigating urban areas. It doesn't merely aim to read signs—it seeks to interpret and vocalize them in a meaningful way, providing autonomy and safety for visually impaired users. In doing so, it aligns with the existing research trajectory while addressing the most urgent unresolved challenges, particularly speed, integration, and contextual interpretation. The goal is not just a better OCR engine—but a complete, user-centric assistive solution.

Chapter 3

Requirements, Impacts and Constraints

3.1 Final Specifications and Requirements

This section describes the technical and resource needs related to the implemented part of the pipeline which includes two-phase signboard detection, OCR-based text extraction, post-processing with a large language model and text-to-speech conversion.

3.1.1 Functional Requirements

The system must identify Bengali signboards in images of the real world street scenes and then extract the name and address areas of the identified signboards. The text fragments that have been extracted have to be put through an OCR engine that can read Bengali script and the raw OCR output then corrected to eliminate errors in recognition. Lastly, the edited text should be translated into readable Bengali audio data that can be consumed by the visually impaired users.

3.1.2 Hardware Requirements

Training on the models was done in two hardware settings, using a personal laptop with an Intel Core i7-14650HX processor with 16 GB of RAM and a dedicated research laboratory workstation at BRAC University, equipped with an Intel Core i9-14900K processor (24 cores, 32 threads), an NVIDIA RTX 4090 that has 24 GB of VRAM, 64 GB of DDR5 RAM, a 1 TB SSD, and a 1200W power supply unit. The research lap pc was primarily used for training while the personal laptop was for overall implementation.

3.1.3 Software and Library Requirements

The pipeline has been written in Python. The YOLO26n and YOLO26s object detector models were installed and trained on the Ultralytics framework, which was based on the PyTorch deep learning backend. Image loading, preprocessing, and cropping were performed on the pipeline stages using OpenCV. EasyOCR has been used as the main optical character recognition engine to extract Bengali text using

cropped region images. It was integrated with the Gemini API, which post-processes the output of the OCR and a well-calculated prompt is issued to the model to fix diacritical errors, spacing anomalies and misrecognized characters in the raw output of OCR. Lastly, the gTTS library was utilized in order to synthesize Bengali audio with the corrected text.

3.1.4 Dataset Requirements

The training data of both detection models were taken by the publicly available Bengali signboard dataset proposed by Mazumder et al. (2025) [49], available at the https://doi.org/10.7910/DVN/UOT0RE. This dataset includes 16,402 annotated Bengali signboard images that were taken in a variety of real-world scenarios in Bangladesh such as night, poor signboard, images with wire blocks, and different lighting situations. Phase 1 training was done using the entire street-scene image subset whereas Phase 2 training was done using the pre-cropped signboard region subset. No other data collection was conducted in this work.

3.2 Societal Impact

The main social advantage of such a system is the ability to enhance the autonomy and mobility of the visually impaired people in Bengali-speaking urban areas where signboards are the key navigation and informational markers which are not available to them otherwise without the help of sighted people. The pipeline makes the process of the detected signboard text transformed into the spoken Bengali audio and, therefore, less dependent on human intervention, which encourages independent involvement in the life of the community. In addition to access, the work also touches on a larger problem of digital equity since Bengali is grossly underrepresented in commercial offerings of OCR and assistive technology as a language, yet it is one of the most commonly spoken languages worldwide. Secondary uses of the system are also in urban navigation, smart city infrastructure, and language education, and could have a positive outcome on safety by allowing users to self-locate essential facilities, like hospitals and pharmacies, in time-sensitive scenarios.

3.3 Environmental Impact

The environmental impact of this project is mostly linked to the energy used to train the YOLO26n and YOLO26s detection models, especially when using the NVIDIA RTX 4090 in the workstation of the university research laboratory during the training of the models. The training of the deep learning models is a computationally intensive task that requires substantial electrical power, and the repetitive training cycles to tune the hyperparameters and compare models add up to this usage. It is, however, important to note that the training was only done up until the satisfactory performance was attained and since a pre-existing publicly available dataset was used, no field data collection activities were carried out, which otherwise would have meant travelling and related emissions. At the inference side, the pipeline is

made lightweight so that it can be executed on a personal laptop without the need of a GPU that is extremely restrictive to the energy cost of real implementation and daily utilization. The dependency of the use of cloud-based APIs, namely the Gemini API to perform post-processing and the gTTS to provide speech synthesis, creates a level of indirect environmental cost due to the use of remote servers, but which is insignificant at the magnitude of the present project. All in all, the environmental footprint of this work is not very big in comparison with its possible use in society, and the next versions of the system may even lessen their footprint by considering the quantized or distilled versions of the models that can preserve the performance with a reduced computational cost.

3.4 Ethical Issues

A number of ethical issues come up during the formulation and implementation of a system that captures and processes real world street images to extract and voice out textual information.

3.4.1 Privacy

The pipeline uses images of street scenes and street scenes were known to be at risk of capturing personally identifiable information that is not limited to text on the signboards, but also faces, vehicle registration plates and other incidental information that can be seen in the image. Although the current implementation is limited to detecting signboards and does not do any identity detection in addition to the captured images of stores, the possibility of abuse in a deployed mobile application should be approached with due attention. When this system is deployed again in the future, it should be rigorously built around data minimization, with frames captured being processed on the device and discarded after inference directly without ever being sent to a remote server or stored on the server.

3.4.2 Bias and Fairness

Detection and OCR modules of the pipeline were trained and tested only on Bengali signboard images obtained in the urban settings in Bangladesh, which mostly represented the visual properties of the street infrastructure in Dhaka. Its performance on signboards in rural, or other Bengali-speaking such as West Bengal, or stylized, handwritten, or severely damaged text that is not overrepresented in the training data, may therefore be less reliable. This is one kind of algorithmic bias which may restrict the fair usefulness of the system to the entire range of its envisioned users.

3.4.3 Responsible API Usage

The addition of a Gemini API to do the post-processing creates a reliance on the use of a third-party commercial service, which casts doubt on the ability to handle data, the terms of service adherence, and the accuracy of AI-generated corrections. Data sent to the Gemini API to be corrected can be done on external servers, and those users of any application deployed on this pipeline should be informed of this data

flow. The timely design had been done with care so that the scope of intervention of Gemini was limited to correcting the typography but not to generative rewriting, which could change the semantic meaning of the original signboard text.

3.5 Project Management Plan

Phase & Timeline	**Key Activities**	**Methodological Focus**	**Deliverables**
Phase 1: Data Preparation (Weeks 1–3)	Download and inspect dataset. Remove corrupted images. Fix bounding box labels via Roboflow and custom script. Apply 70:20:10 split.	Data cleaning, label validation, dataset partitioning.	Cleaned Phase 1 dataset (590 images). Cleaned Phase 2 dataset (3,213 images). Verified annotation files.
Phase 2: Model Training (Weeks 4–8)	Train YOLO26n and YOLO26s on Phase 2 dataset. Evaluate and select best model. Train YOLO26s on Phase 1 dataset. Apply fine-tuning on best checkpoint.	Architecture comparison, hyperparameter tuning, transfer learning, early stopping.	Trained Phase 2 weights (YOLO26s, YOLO26n). Fine-tuned Phase 1 weights (YOLO26s). Training logs and plots.
Phase 3: Pipeline Integration (Weeks 9–11)	Integrate Phase 1 and Phase 2 models. Connect EasyOCR to Phase 2 output. Integrate Gemini API for post-processing. Connect gTTS for audio output.	End-to-end pipeline design, API integration, prompt engineering.	Functional end-to-end pipeline script. Gemini correction prompt. Audio output from real signboard images.
Phase 4: Evaluation & Reporting (Weeks 12–14)	Evaluate all models on test sets. Generate confusion matrices and training curves. Qualitative OCR evaluation. Write thesis report.	Quantitative benchmarking, qualitative analysis, comparative analysis with Mazumder et al. (2025).	Final performance metrics. Confusion matrices and plots. Completed thesis report.

Table 3.1: Project Schedule and Deliverables

Resource Category	Specifications / Requirements	Budget / Allocation Strategy
Computational Hardware	Personal laptop: Intel Core i7-14650HX, 16GB RAM. University research lab: Intel Core i9-14900K, NVIDIA RTX 4090 24GB, 64GB DDR5 RAM, 1TB SSD.	Personal laptop used at no cost. Lab workstation accessed through university research infrastructure at no personal expense.
Dataset	Bengali signboard dataset by Mazumder et al. (2025). Publicly available at https://doi.org/10.7910/DVN/UOT0RE.	No cost — freely available open-access dataset.
Software Stack	Python, PyTorch, Ultralytics, OpenCV, EasyOCR, gTTS, Roboflow, VS Code.	All tools are open-source and free of charge. No licensing fees incurred.
API Services	Gemini 2.5 Flash API (Google AI Studio). gTTS (Google Text-to-Speech).	Gemini API used under free tier — no cost incurred during development. gTTS is free and open-source.
Human Resources	Primary researcher (pipeline design, training, integration, evaluation). Supervisor (periodic feedback and guidance).	Conducted as primary research activity. No external labor cost.
Total Estimated Cost	**Approximately $0 in direct expenditure**	

Table 3.2: Resource and Budget Management

3.6 Risk Management

This section determines the major risks that are experienced and expected during the development of the pipeline and its possible deployment as well as the mitigation measures that are taken or proposed.

3.6.1 OCR Accuracy on Bengali Script

Although EasyOCR could process Bengali, it was not customized to the visual information requirements of real-world Bengali signboards, which often have stylized fonts, mixed scripts, ornaments and damaged surfaces. The raw OCR results were not complete or had diacritical mistakes that made the content extracted to be incorrect semantically. This was avoided by adding the Gemini API as an after-processing layer, which was requested to rectify typographical and diacritical mistakes, but retain the original meaning of the content on the signboard. This two-step method was very effective in enhancing the quality of the final text output without the need

of retraining the OCR engine.

3.6.2 Model Performance on Degraded Images

The two detection stages have the risk of poor performance in the case of signboard images highly occluded, at sharp angles or in low light conditions. The dataset in the training has a wide variety of such conditions, but still, there are edge cases. The latter threat was partly addressed by training both YOLO26n and YOLO26s variants and comparing their relative performance, which allowed choosing between them the more suitable model based on the deployment environment.

3.6.3 API Dependency and Availability

The post-processing step of the pipeline requires the Gemini API and the TTS step of the pipeline requires the use of gTTS, both of which need an active internet connection and subject to outside service availability, rate limits, and possible modifications in pricing and access policies. Any interruption of one or the other of these services would rupture the pipeline at these steps. This risk is known to be a limitation that is tolerated in the present research prototype. This dependency should be removed by looking into locally deployable options of both correction and speech synthesis in future iterations of the technology.

3.6.4 Dataset Bias and Generalization

As mentioned in the ethical issues section, the training data used is only urban Bengali signboards in Bangladesh and this may restrict the generalizability of the detection models to other settings. The inherent diversity of the Mazumder et al. (2025) data, comprising nighttime scenes, wire occlusions, fading text, and different resolutions partially benefited to mitigate this risk. Nevertheless, this risk would need additional data gathering in rural and non-Dhaka settings to completely avert in a production deployment.

3.6.5 Hardware Dependency During Training

The slow training of YOLO26 models on a CPU-only machine was impractically slow, making it dependent on the availability of the university research laboratory GPU workstation. The loss of access to that resource at any point within the training period would have grossly set back the project schedule. This was alleviated by performing all the exploratory and debugging tasks locally on the personal laptop and only using thegpu resources on full training jobs.

3.7 Economic Analysis

This pipeline was developed with a low direct financial cost since most of the tools and frameworks utilized are open-source and free. Python, PyTorch, Ultralytics, OpenCV, EasyOCR and gTTS do not have any license fees. The training dataset was also made publicly available by Mazumder et al. (2025) without any fee. Hardware expenses were also not incurred with training being performed on a university

research laboratory workstation supplied at no personal cost with additional prototyping performed on a personal laptop.

The only element that has a possible cost implication is the Gemini API, which is applied in OCR post-processing. In this project, the free version of Gemini 2.5 Flash was employed and this version offers free input and output tokens with no financial fee as per the free version. Nevertheless, the free version has significant restrictions: the usage statistics can be used to enhance the products of Google, rate limits are much more restrictive compared to the paid version, and such advanced features as context caching and batch processing are not available. These limitations were not a practical issue particularly in the case of a research prototype that was going to process a limited number of images. The bottleneck would however arise in case the pipeline were to be scaled to a production application with real users in the pipeline at all times. To upgrade to the paid version of Gemini 2.5 Flash would use 0.30 per million input tokens and 2.50 per million output tokens. Since signboard text is relatively short, the number of tokens spent on each API call is very small, which implies that the cost per request at scale would be low. It has been estimated roughly that it would take less than 100 cents at the paid tier rates to process 10,000 signboard correction requests, thus it would be economical even at a moderate level of deployment.

Overall, this project had a development cost that was essentially free of time, and the estimated cost of a deployed version is insignificant at small to medium scale, so this pipeline is a cost-effective solution to assistive technology development in low-resource environments.

Chapter 4

Proposed Methodology

4.1 Design Process or Methodology Overview

This paper describes the design, implementation and evaluation of a multi-stage real-time pipeline to Bengali scene text recognition which was created specifically to be an assistive technology to help visually impaired persons navigate a complex urban environment. The system is meant to scan the textual information contained on the physical sign board, then isolate the written text and deliver it to the user in the form of synthesized Bengali speech, allowing the user to understand the textual information surrounding him or her without depending on the visual information and sighted help.
It has a modular, sequential pipeline architecture, whereby each of the stages carries out a specific task with its output being piped to the next stage. The pipeline works in the following manner:

Stage 1: Signboard Detection

Signboard Detection The initial step involves detection and localization of signboards on an image or video frame that is captured. A special object detection model is used to scan the entire scene and generate bounding boxes around parts of the scene that contain Bengali signboards thus eliminating irrelevant image content and directing computational resources to areas of interest.

Stage 2: Region of Text Interest (RTI) Detection

Stage 1 images are cropped signboards are sent to a second detection model that recognizes and isolates particular sub-regions with the establishment name and address. The two-stage detection strategy helps to make sure that the signboard only presents the most informative elements into the OCR stage, enhancing the accuracy of the recognition and minimizing noise.

Stage 3: Optical Character Recognition (OCR)

The images of the cropped text area are fed into an OCR engine that converts the visual representation of the Bengali text into machine-readable character strings. This is the most challenging stage technically because the Bengali script has the most morphological complexity as it has conjunct characters, diacritical marks, and variable spacing.

Stage 4: Post-Processing

Raw OCR output is then improved with a large language model to fix diacritical mistakes, space errors, and characters that are misrecognized or atypical artifacts of OCR on real-world Bengali scene text. This step enhances the semantic correctness and readability of the extracted text to a great extent, and then it is sent to speech synthesis.

Stage 5: Text-to-Speech Conversion
The edited text will be translated into natural Bengali audio output by the use of a text-to-speech engine which will be the main channel of communication with the visually impaired user.

The evaluation of performance is conducted based on common quantitative measures such as precision, recall, F1-score and mean average precision (mAP) of the detection phases, as well as a qualitative measure of the final audio output intelligibility and naturalness.

4.2 Preliminary Design or Design (Model) Specification

Here the preliminary design and modeling approach for the proposed artificial-intelligence enabled signboard recognition system, which was designed as an aid for the visually impaired by converting the signboard text to a speech mode, is presented. The designed system consists of three major subsystems, namely, the detection of signboard, the text extraction and the text-to-speech conversion, and utilizes a mixture of state-of-the-art computer vision and natural language processing techniques. In the following route of optimization, a number of design options were developed and analyzed both by theory and by selective experiments on its realization.

The study uses a pipeline assessment approach and considers three increasingly advanced designs, which share a common detection architecture, but which vary in terms of text-recognition scheme. All implementations start with the YOLOv9s detector, trained to detect signboards in unconstrained environments, which is followed by different implementations of optical character recognition (OCR) and text-to-speech synthesis.

4.2.1 Pipeline Design Variations and Evaluation

Two pipeline configurations were discussed and compared prior to deciding on a final implementation; one being an already existing approach that was found in the literature which addressed the same problem but with the same data set and the other the approach that was finally adopted in this work. Mazumder et al. (2025) offered a good point of reference, having developed an end-to-end Bengali signboard recognition system that was also trained on the same publicly available dataset as in this case [49]. They adopted a two-phase structure of detection with a YOLOv9s

in both phases to detect signboards and localize them with RTI, where the Phase 1 mAP was 99.1 percent and Phase 2 mAP was 92.89 percent. They used Tesseract OCR with a custom binarization strategy, which uses pixel histograms to determine the threshold (75th percentile of the pixel grayscale distribution) to make cleaner binary images than normal CV2 binarization, for text extraction. This produced 91 percent accuracy of OCR over 1,600 test images. These are good numbers in general, although the OCR accuracy value points to the actual weakness of Tesseract, as, although it can be helped with better preprocessing, the visual complexity of the real-world Bengali signboards poses a serious challenge to it, especially when the fonts are changing, stylized, and partially damaged. This work is based on this foundation with the pipeline implemented overcoming some of its limitations. Two-phase detection architecture is also maintained, except that YOLO26n and YOLO26s are now used in place of YOLOv9s, where the new generation of the YOLO family has been improved in architectural terms in January 2026. In the case of OCR, EasyOCR was selected instead of Tesseract due to two main factors: it directly recognizes Bengali script without any manual binarization pre-processing, and is therefore easier to use with a wide range of image types. The most notable addition, however, is the addition of a Gemini API based post-processing step, where a structured prompt is applied to fix diacritical errors, spacing errors and misrecognized characters in the raw OCR output, before it goes to speech synthesis. The given step of the pipeline, which is driven by an LLM and is absent in the Mazumder et al. one, is the first methodological contribution of this work to the OCR phase. The assistive functionality is completed with the gTTS that provides audio output in Bengali.

4.2.2 Integrated Subsystem Design

The adopted pipeline is composed of five closely coupled steps with their output transferred straight to the next one in the following manner:

1. **Signboard Detection (Phase 1)** A video frame or input image is inputted into a YOLO26 model that was trained on the entire street-scene subset of the Mazumder et al. (2025) dataset. The model generates bounding boxes of all the Bengali signboards that are detected in the scene. Detection areas are clipped and sent to Phase 2 one at a time.

2. **RTI Detection (Phase 2)** The image of every cropped signboard is forwarded to a second YOLO26 model that is trained on the pre-cropped signboard subset of the same dataset. In this model, the name and address subregions in the signboard are identified and isolated. These are cropped RTI images which are sent to OCR step.

3. **OCR with EasyOCR** The RTI images that are cropped are fed through Easy OCR that extracts Bengali text of the visual input and gives out a raw character string. There is no need to perform any manual binarization

preprocessing, where EasyOCR does internal image normalization. The unrefined output at this point can be filled with recognition errors common to complicated text of Bengali scenes.

4. **Post-Processing with Gemini-API** The raw OCR text is sent to the Gemini API and a structured prompt is sent to the model to fix diacritical marks, spacing and misrecognized characters and maintain the original intent of the text. The fixed string comes out as the input into the last stage.

5. **Text-to-Speech with gTTS** The fixed Bengali text is submitted to Google Text-to-Speech library which synthesizes natural Bengali audio output. This sound is sent to the user as the last pipeline output allowing one to understand the content of the signboard without visual aid.

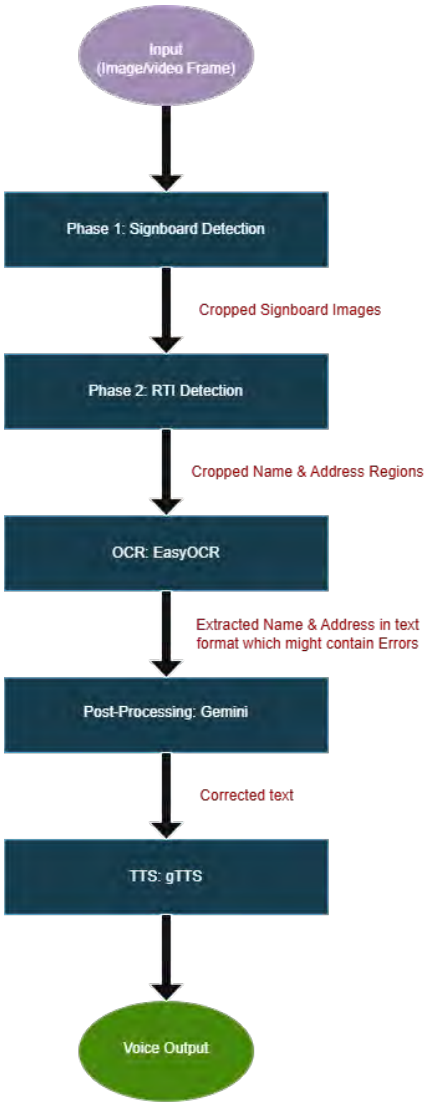

Figure 4.1: End-to-end pipeline architecture of the proposed Bengali signboard recognition system, illustrating the sequential flow from raw image input through signboard detection, RTI detection, OCR, LLM-based post-processing, and text-to-speech conversion to final voice output.

4.2.3 Predicted System Behavior and Simulation Framework

Considering the design of the implemented pipeline and the performance indicators that Mazumder et al. (2025) [49] set with the same data set, the following behaviors of the system are expected:

Detection Stage: YOLO26n and YOLO26s will be able to meet or even surpass the detection performance of the YOLOv9s due to the architectural enhancements provided in the YOLO26 generation. Phase 1 detection mAP should be close to or even higher than 99 and Phase 2 RTI detection mAP should match the same as well. Both of the models are anticipated to have real-time inference rates applicable to run on hardware with GPUs.

OCR Stage: EasyOCR will be able to generate competitive raw recognition on Bengali scene text. Although direct comparison to Tesseract on the same conditions cannot be made in previous literature on this particular dataset, removal of the manual binarization dependency is likely to enhance robustness to the changing image conditions. The post-processing step of the Gemini should be able to significantly lower both the character and word error rate of the final output as compared to the raw OCR results.

Speech Stage: gTTS is expected to produce clear and intelligible Bengali audio output with low latency, consistent with its established performance characteristics for Bengali text synthesis.

4.3 Data Collection

The training data of both detection stages was obtained by the publicly available Bengali signboard dataset provided by Mazumder et al. (2025) and available on the web at https://doi.org/10.7910/DVN/UOT0RE. This dataset consists of 8,366 full street-scene images of Phase 1 and 8,036 pre-cropped signboard region images of Phase 2, which in total amount to 16,402 annotated Bangladesh-collected signboard images under a variety of real-world conditions. But after downloading the data, the number of images obtained in practice was much less than the ones mentioned in the article and in Phase 1 and Phase 2, 601 and 3218 images were obtained respectively. This discrepancy is not clear and can be related to the incompleteness of hosting, unavailability at the time of download, or access limitations at a platform level. No further data gathering was conducted as we evaluate the available dataset's performance first.

4.3.1 Data Cleaning

After downloading, the dataset was checked on the presence of corrupted files that were not openable or could not be properly read. Few of such images were detected and eliminated out of the two subsets. The end result of the cleaning was 590 and 3213 images used in Phase 1 and Phase 2 respectively. In addition to the cleaning

of images, the annotation files in the YOLO format were also inspected and some of the bounding box annotations were discovered to have positional errors. These were manually fixed by Roboflow which offered a convenient interface to view and adjust bounding box coordinates in relation to relevant images.

4.3.2 Data Transformation

The dataset was not subjected to any image-level transformation or preprocessing before being trained. Particularly, no auto-orientation correction or resizing was done, because the aim was to test the detection models in conditions that best represent the variability of real-world inputs, when images can be presented in arbitrary orientations, resolutions and aspect ratios. This was intentional to enable the YOLO26 models to be tested with the entire diversity of inputs that exist in the dataset without normalization that may obscure performance attributes that are pertinent in the actual deployment contexts.

4.3.3 Data Integration

The two subsets of the data had different purposes in the pipeline. Full street-scene images in the Phase 1 subset were only used to train the initial detection model which had the duty of localizing signboards in complex urban scenes. The second detection model that was trained on the Phase 2 subset, which comprised of pre-cropped signboard images, was tasked with the responsibility to detect name and address regions within individual signboards. There was no cross-contamination of phases because the two subsets were completely separated during training.

4.3.4 Data Reduction

There was no post-processing of data other than the elimination of corrupt images as outlined in Section 4.3.1. Since the downloaded subset was already significantly smaller than the entire dataset that Mazumder et al. (2025) reported [49], no additional reduction was deemed necessary and could have a negative impact on model performance. The 590 and 3,213 images were split into cleaned subsets, which were used as a whole to be trained and validated.

4.4 Implementation of Selected Design

It was implemented completely in Python on Visual Studio Code as the development platform. It was constructed as a chain of modular scripts, one script per system stage, and connected into one end-to-end inference workflow. In this section the implementation decisions are described at each stage.

1. **Dataset Preparation and Label Fixing:** Before training, the two dataset subsets were subjected to a label validation step that was not corrected in

Roboflow. A custom program was developed to search all YOLO-format annotation files of the training, validation and test splits and to determine any bounding box labels whose class index fell outside of the valid range as stated in the dataset configuration file. In the case of Phase 1, a single-class detection task, only the index of classes 0 is valid. The labels that had a greater index of class were corrected automatically to class 0. This was to make sure that no defective annotations would disrupt the training process. The two subsets of data were divided into training, validation and test sets respectively in the 70:20:10 ratio.

2. **Phase 1 Training — Signboard Detection:** The training of phase 1 was performed with YOLO26n and YOLO26s, which was loaded through the Ultralytics framework. The original training was conducted on 400 epochs at 1024x1024 image resolution, AdamW optimizer, initial learning rate of 0.003 and cosine learning rate schedule (to final rate of 0.00005). The early stopping patience was set at 50 epochs. The batch size was set to automatic whereby Ultralytics can make decisions on the best batch size, which in this case is determined by the available memory of the RTX 4090. In order to mitigate the CUDA memory fragmentation when training, the PYTORCHCUDAALLOCCONF environment variable was configured to expandablesegments:True. Mixed precision training (AMP) was switched off to prevent the instability problems that had been observed with RTX 4090 on this setup. Data augmentation was HSV jitter, rotation up to 5 degrees, translation, scaling, mosaic augmentation and a small mixup ratio of 0.1. VRAM spikes were observed and led to the copy-paste augmentation being disabled. The loss weights had a box loss gain of 8.0, classification loss gain of 0.3 and DFL loss gain of 1.5 and were tuned to a single-class detection task. The checkpoints were periodically saved in 5 epochs during training. After the first training, a fine-tuning phase was done on the best checkpoint of each model. The further 100 epochs fine-tuning was performed with a low image resolution of 640x640 with a exceedingly low initial learning rate of 0.0001 declining to a final rate of 0.01, and mosaic augmentation was removed during the last 20 epochs to enhance localization stability. To further optimize the regression of bounding boxes, box loss weight was raised to 10.0 and a mixup ratio of 0.15 was used to enhance resistance to overlapping signboards.

3. **Phase 2 Training — RTI Detection:** The phase 2 training was set up similarly with configuration options based partially on the reported hyperparameter settings of Mazumder et al. (2025) [49] on the same detection task. YOLO26n and YOLO26s were trained during 200 epochs using 640x640 images. An optimizer was AdamW replaced with the SGD optimizer, with an initial learning rate of 0.01, a momentum of 0.937, and a weight decay of 0.0005, which is in line with the conventional YOLO training. Warmup 3 epochs of warmup momentum 0.8 was used. Box loss, classification loss and DFL loss were to be set at 7.5, 0.5 and 1.5 respectively. The augmentation settings consisted of 3 degrees rotation, translation, scaling, mosaic, mixup. Early stopping was done with a patience of 50 epochs and training could be resumed at the last checkpoint to take into consideration possible interruptions.

4. **OCR and Post-Processing:** The OCR and post-processing steps were made up of a single inference script. Based on an input image, the Phase 2 YOLO26 model first detects with a confidence threshold of 0.5 and yields bounding box categories of either name regions or address regions. The OpenCV is used to crop each detected region off the image and send them to EasyOCR which was configured to support Bengali and English languages. EasyOCR works on each crop and provides the extracted text that is extracted individually under the name and address classes. The combined raw OCR result is next passed to the Gemini API with the gemini-2.5-flash model. A guided prompt asks Gemini to be knowledgeable on Bengali language and Dhaka geography, to fix typical OCR mistakes like character replacements in Bengali diacritics, and to make the address geographically consistent. The prompt is also very strict on the form of the output to two lines a corrected name and a corrected address, and it can avoid any form of generative rewriting except typographical correction. The modified output of Gemini is the output that represents the signboard content in the last text form.

5. **Text-to-Speech:** The corrected text string returned by Gemini is sent to gTTS which synthesizes Bengali audio output. The audio output is stored as an audio file and can be reproduced as the overall output of the pipeline providing the user with the signboard information in spoken Bengali.

Chapter 5

Result Analysis

In this part, the quantitative findings of every detection step of the introduced pipeline are given, while the evaluation of the OCR and post-processing steps is conducted qualitatively.

5.1 Performance Evaluation

Phase 2 was prioritized over Phase 1 since, the dataset of Phase 2 had more images, which gave more training stability and due to the fact that the reference paper of Mazumder et al. (2025) [49] had already a strong Phase 1 baseline of 99.1% mAP50. YOLO26s and YOLO26n were both trained and tested on Phase 2. Table 5.1 shows results on the validation set.

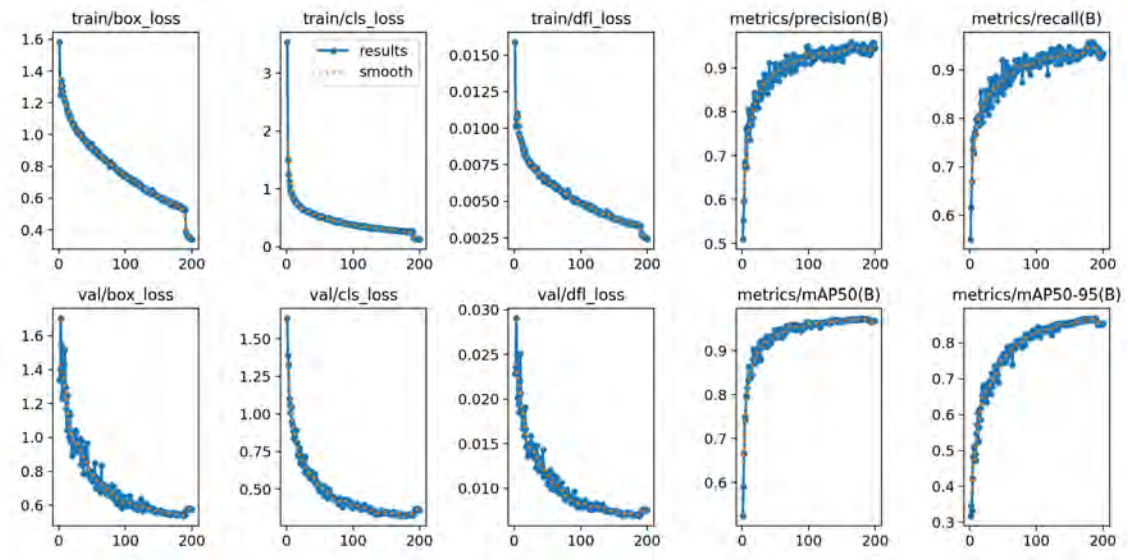

Figure 5.1: Training and validation loss and mAP curves for YOLO26s Phase 2 RTI detection over 200 training epochs, demonstrating stable convergence across both the *name* and *address* detection classes.

Class	Model	Images	Instances	P	R	mAP50	mAP50-95
All	YOLO26s	643	1272	0.941	0.947	0.971	0.863
Name	YOLO26s	627	633	0.955	0.973	0.977	0.906
Address	YOLO26s	637	639	0.927	0.920	0.965	0.820
All	YOLO26n	643	1272	0.938	0.913	0.958	0.821
Name	YOLO26n	627	633	0.963	0.947	0.971	0.877
Address	YOLO26n	637	639	0.912	0.879	0.944	0.765

Table 5.1: Phase 2 Validation Results — YOLO26s vs YOLO26n

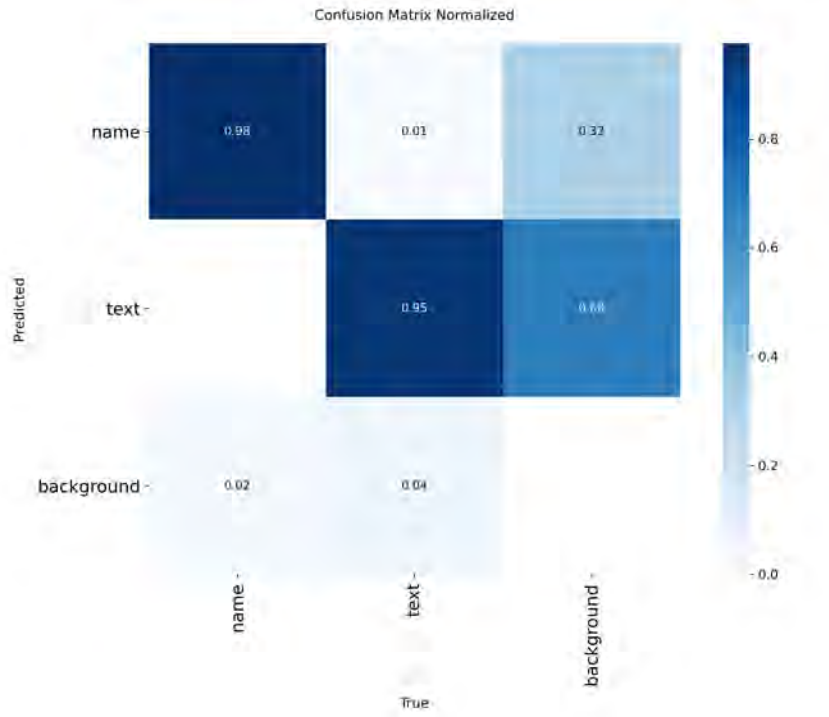

Figure 5.2: Confusion matrix for YOLO26s Phase 2 RTI detection, showing classification performance across the *name* and *address* region classes on the validation set.

Phase 2 showed that YOLO26s outperforms YOLO26n on all metrics, with the largest differences in overall recall (0.947 vs 0.913) and mAP50-95 (0.863 vs 0.821) being the indicators that it generalizes better to smaller IoU thresholds. The address class was under all models lower than the name class by expected margin as there should be more visual complexity and typographical variation in address regions in real world Bengali signboards. These findings led to the choice of the model of interest to use in Phase 1 training; YOLO26s.
Phase 1 was trained with YOLO26s alone, as it was more effective in Phase 2. It was trained in two phases: the first phase of training consisting of 200 epochs and the second phase of fine-tuning consisting of 100 extra epochs based on the most successful checkpoint. Table 5.2 and Table 5.3 present results on the validation set and test set respectively.

A consistent increase was obtained in all metrics, and precision rose to 0.926, mAP50

Figure 5.3: Sample validation batch predictions from the YOLO26s Phase 2 RTI detection model, showing detected bounding boxes for *name* and *text* (address) regions across a diverse set of real-world Bengali signboard images. Confidence scores are displayed alongside each predicted class label.

Metric	Initial Training	After Fine-Tuning
Images	118	118
Instances	121	121
P	0.890	0.926
R	0.940	0.925
mAP50	0.966	0.973
mAP50-95	0.894	0.916
Inference (ms)	1.4	1.3

Table 5.2: Phase 1 YOLO26s — Validation Results Before and After Fine-Tuning

to 0.973 and mAP50-95 to 0.894 to 0.916, which proves that the extra refinement step was helpful.

Metric	Value
Images	59
Instances	60
P	0.9139
R	0.9500
mAP50	0.9714
mAP50-95	0.9492
Inference (ms)	6.8

Table 5.3: Phase 1 YOLO26s — Final Test Set Results (After Fine-Tuning)

At Phase 1, signboard detection with fine-tuned YOLO26s model, the test set mAP50 and mAP50-95 were 97.14% and 94.92%, respectively, which is a high level of generalization to unseen data.

Phase	Model	mAP50	mAP50-95	Precision	Recall
Phase 1	YOLO26s (fine-tuned)	0.9714	0.9492	0.9139	0.9500
Phase 2	YOLO26s	0.9710	0.8630	0.9410	0.9470
Phase 2	YOLO26n	0.9580	0.8210	0.9380	0.9130

Table 5.4: Cross-Phase and Cross-Model Detection Performance Summary

5.2 Analysis of Design Solutions

The two stages both reached mAP50 of over 97 percent with YOLO26s, which indicates that the two-phase detection architecture can be applied in the two signboard-level and region-level detection tasks and that it is effective and consistent. The larger difference between mAP50 and mAP50-95 in Phase 2 compared to Phase 1 indicates that although the model is performing well in localizing name and address regions in most cases, the precision of the bounding box at stricter IoU values is a little lower than in the more visually varied RTI regions.

Metric	Mazumder et al.	YOLO26s	YOLO26n
Model Architecture	YOLOv9s	YOLO26s	YOLO26n
mAP50	0.977	0.971	0.958
Precision	N/A	0.941	0.938
Recall	N/A	0.947	0.913
Inference Time	∼21.4 ms (GPU)	∼34.7 ms (CPU)	∼12.5 ms (CPU)
Training Dataset Size	8,036 images	3,213 images	3,213 images

Table 5.5: Phase 2 Comparison with Mazumder et al. (2025)

One of the observations made in Table 5.5 is that the models used in this work produced competitive scores of mAP50 with only 3,213 training images (approximately 40% of the 8,036 images of Mazumder et al. (2025) [49]. This implies that YOLO26 is able to generalize effectively when the data available is significantly low which is a significant finding considering that the download of the entire dataset is not complete as stated in Section 4.3.

5.3 Final Design Adjustments

The fine-tuning step implemented on the Phase 1 model was a valuable addition that resulted in better results in all validation metrics. The choice to raise the box loss weight to 10.0 in the fine-tuning step and to deactivate mosaic augmentation in the last 20 epochs also helped in achieving closer bounding box regression and more reliable convergence. In Phase 2, YOLO26s was chosen instead of YOLO26n as the final model due to its higher recall and mAP50-95, which are of greater concern to an assistive application where the cost of a text region being missed is significant compared to a slightly slower inference time.

Figure 5.4: Sample validation batch predictions from the fine-tuned YOLO26s Phase 1 model, showing detected signboard bounding boxes overlaid on real-world street scene images from the validation set.

5.4 Statistical Analysis

The F1-score, which is a balance between precision and recall, is a rather informative measure in this task due to the asymmetry in the cost of false negatives in an assistive setting. The F1-scores of the models and phases are computed and presented in Table 5.6.

Phase	Model	Precision	Recall	F1-Score
Phase 1	YOLO26s (fine-tuned)	0.9139	0.9500	0.9316
Phase 2	YOLO26s	0.9410	0.9470	0.9440
Phase 2	YOLO26n	0.9380	0.9130	0.9253

Table 5.6: F1-Score Summary Across Models and Phases

Phase 2, Phase 2 has the best F1-score of 0.9440, indicating an optimal precision-recall tradeoff. The Phase 1 fine-tuning model has an F1-score of 0.9316, slightly lower because the precision dip on the test set is slightly larger than that on the validation set, which is as expected with a small test split of 59 images.

5.5 Comparisons and Relationships

The findings between the two phases provide a similar trend: YOLO26s is more reliable in performance than YOLO26n on mAP50-95 and recall whereas the latter provides faster inference on the CPU at 12.5ms than the former at 34.7ms. The practical implications of this tradeoff are clear in that YOLO26s is the better choice to use in a deployment that requires accuracy to use the GPU and that it is possible to consider using YOLO26n in a deployment that is constrained by inference speed.

5.6 Discussions

Figure 5.5 illustrates an actual example of pipeline output, a Bengali pharmacy signboard, which was run in Phase 2 detection, EasyOCR and Gemini post-processing. The raw output of the EasyOCR on this image was the following:

Figure 5.5: Real-world Bengali pharmacy signboard used for qualitative evaluation of the OCR and post-processing pipeline stages.

Field	Raw EasyOCR Output	Gemini Corrected Output
Name	সিকদীার ফেমসী	সিকদার ফার্মেসী
Address	৯৭ ২, পশ্চিম জাফরাবাদ, পুলপার, সাদেক খাঁণ রোড 9 মৌঃপূর, ঢাকা-১২০৭	৯৭/২, পশ্চিম জাফরাবাদ, পুলপার, সাদেক খান রোড, মোহাম্মদপুর, ঢাকা-১২০৭

Table 5.7: Qualitative OCR Comparison — Raw EasyOCR vs Gemini Post-Processing

In the comparison between the corrected output and the real content of the signboard as it appears in Figure 5.5, the Gemini correction was able to correct the diacritical error in the name of the shop (phmesii - phaarmesii), correct the address separator (97 2 - 97/2), correct the name of the road (khaaNnn - khaan), and correct the abbreviated name of the area (mauHpuur - mohaammdpur). Such corrections are semantically useful and geographically correct, and prove that the post-processing stage based on LLM can add considerable value to the one offered by EasyOCR itself to real-world Bengali scene text.

The fixed text output generated by the Gemini post-processing step was fed to the Google Text-to-Speech library (gTTS), which generated the final output of Bengali

audio. In the pharmacy signboard case, the corrected name সিকদার ফার্মেসী and the corrected address ৯৭/২, পশ্চিম জাফরাবাদ, পুলপার, সাদেক খান রোড, মোহাম্মদপুর, ঢাকা-১২০৭ were both converted to the intelligible speech of the Bengali. The sound was recorded as an audio file, which was an mp3 file and listened to as the last pipeline output. Subjectively, the synthesized speech was fluent and the corrected text was much more natural-sounding than the output the raw EasyOCR would have given, especially the address field with several corrections made. This proves that the integration of Gemini-corrected text and gTTS synthesis provides a workable and convenient final-to-end result stage to the suggested assistive pipeline.

Chapter 6

Conclusion

6.1 Summary of Findings

The work demonstrated the application of a two-stage Bengali signboard-detecting pipeline as the basic building block of a larger assistant technology platform to support the visually impaired consumer. On the publicly accessible dataset that Mazumder et al. (2025) introduced, Phase 2 RTI was trained on and evaluated on both YOLO26s and YOLO26n, with the former being able to achieve a higher mAP50 of 97.1% and F1-score of 0.9440. Phase 1 was then trained using YOLO26s with a two stage training and fine-tuning step, and produced a final test mAP50 of 97.14% and mAP50-95 of 94.92%. It is important to note that these results were obtained with only about 40 percent of the entire dataset used by the reference paper, indicating high data efficiency with the YOLO26 architecture. The post-processing stage implemented on the Gemini platform showed significant enhancement on raw EasyOCR output, diacritical errors, address formatting errors, and abbreviated place names were all addressed correctly in actual Bengali signboard text.

6.2 Contributions to the Field

The key contributions of this paper are the benchmarking of the YOLO26 architecture on Bengali signboard detection with a known public dataset and the introduction of an LLM-based post-processing layer with the help of the Gemini API to correct Bengali OCR output, to the best of my knowledge, this specific combination of EasyOCR and a large language model for Bengali signboard text correction has not been reported in prior literature on this dataset. The paper also shows that it is possible to achieve the performance of competitive detection using a much smaller training set, which can be applied in future low-resource research in Bengali computer vision.

6.3 Recommendations for Future Work

The present pipeline is limited to signboard only and its extension to other types of real-world text information including posters, banners, restaurant menus, and road signs would

greatly expand its practical use as a tool of assistance. More importantly, the pipeline is yet to be experimented in the real-life conditions outside of isolated image inputs, and field testing in a real-life urban setting is a critical step. Although the post-processing step is qualitatively attractive, it does not have formal quantitative indicators to objectively assess the extent of the improvement offered by Gemini correction over raw EasyOCR output - a benchmark of this step based on CER and WER on a collection of annotated images would seriously enhance the assessment of the entire pipeline. Lastly, practical utility, intelligibility, and responsiveness of the system should be evaluated with end-to-end user tests on visually impaired individuals to evaluate the system in assistive use conditions.

Bibliography

[1] A. W. Black, P. Taylor, and R. Caley, *The Festival Speech Synthesis System*, University of Edinburgh, [Online]. Available: https://www.cstr.ed.ac.uk/projects/festival/, 1998.

[2] C. M. Shi, C. Yao, and C. L. Liu, "Scene text detection and recognition: Recent advances and future trends," *Frontiers of Computer Science*, vol. 10, no. 1, pp. 19–36, 2016. DOI: 10.1007/s11704-015-4488-0.

[3] M. J. Adnan, A. A. Quraishi, and A. Rahman, "Text to speech synthesis for bangla language," 2017, B.Sc. Thesis, Dept. of Computer Science and Engineering, BRAC University, Dhaka, Bangladesh.

[4] A. K. M. R. Hoque, P. M. Pandit, N. Nasreen, and H. Raihan, "Bangla text extraction by digital image processing," Dec. 2017, B.Sc. Thesis, BRAC University, Dhaka, Bangladesh.

[5] A. Howard et al., "MobileNets: Efficient convolutional neural networks for mobile vision applications," *arXiv preprint arXiv:1704.04861*, 2017.

[6] K. R. Jumani, "Increasing the position precision of a navigation device by a camera-based landmark detection approach," M.Sc. thesis, Chair of Computer Engineering, TU Chemnitz, Chemnitz, Germany, 2017.

[7] A. D. Mohammed, "Camera-phone obstacle detection and emergency exit-sign recognition to support autonomous navigation," M.Sc. thesis, School of Electrical & Electronic Engineering, University of Manchester, Manchester, U.K., 2017.

[8] M. A. Rahaman, "Computer vision based bangla sign language recognition," Ph.D. dissertation, Dept. of Computer Science and Engineering, University of Dhaka, 2017.

[9] C. Baek, B. Lee, D. Han, S. Yun, and H. Kim, "Character region awareness for text detection," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 9365–9374. DOI: 10.1109/CVPR.2019.00959.

[10] D. Gupta, E. Hossain, M. S. Hossain, K. Andersson, and S. Hossain, "A digital personal assistant using bangla voice command recognition and face detection," in *2019 IEEE Int. Conf. on Robotics, Automation, Artificial-intelligence and Internet-of-Things (RAAICON)*, 2019, pp. 110–114. DOI: 10.1109/RAAICON48939.2019.9087492.

[11] K. A. Islam, N. F. Mubin, and S. Zaman, "Road sign detection and translation in bangla using image processing and machine learning," Aug. 2019, B.Sc. Thesis, Dept. of Computer Science and Engineering, BRAC University, Dhaka, Bangladesh.

[12] D. Paul and B. B. Chaudhuri, "A blstm network for printed bengali ocr system with high accuracy," *arXiv preprint arXiv:1908.08674*, 2019. [Online]. Available: https://arxiv.org/abs/1908.08674.

[13] B. Shi, M. Yang, X. Wang, P. Lyu, C. Yao, and X. Bai, "ASTER: An attentional scene text recognizer with flexible rectification," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 41, no. 9, pp. 2035–2048, Sep. 2019. DOI: 10.1109/TPAMI.2018.2848939.

[14] World Health Organization (WHO), *World report on vision*, World Health Organization, Geneva, [Online]. Available: https://www.who.int/publications/i/item/world-report-on-vision, 2019.

[15] F. Hasan, S. N. Shuvo, S. Abujar, M. Mohibullah, and S. A. Hossain, "Bangla continuous handwriting character and digit recognition using cnn," in *Innovations in Computer Science and Engineering*, Springer, 2020, pp. 555–563. DOI: 10.1007/978-981-15-5113-0_55. [Online]. Available: https://doi.org/10.1007/978-981-15-5113-0_55.

[16] R. Pramanik and S. Bag, "Segmentation-based recognition system for handwritten bangla and devanagari words using conventional classification and transfer learning," *IET Image Processing*, vol. 14, no. 5, pp. 959–972, 2020. DOI: 10.1049/iet-ipr.2019.1473. [Online]. Available: https://doi.org/10.1049/iet-ipr.2019.1473.

[17] R. Tapu, B. Mocanu, and T. Zaharia, "Wearable assistive devices for visually impaired: A state of the art survey," *Pattern Recognition Letters*, vol. 137, pp. 37–52, 2020. DOI: 10.1016/j.patrec.2020.06.007.

[18] T. M. K. Anik and M. S. Islam, "Bangla speech synthesis using deep neural networks," in *2021 IEEE Region 10 Symposium (TENSYMP)*, 2021, pp. 1–6. DOI: 10.1109/TENSYMP52854.2021.9550868.

[19] T. R. Das, S. Hasan, M. R. Jani, F. Tabassum, and M. I. Islam, "Bangla handwritten character recognition using extended convolutional neural network," *Journal of Computer and Communications*, vol. 9, pp. 158–171, 2021. DOI: 10.4236/jcc.2021.94009.

[20] T. Ghosh, M. H. Z. Abedin, H. A. Banna, N. Mumenin, and M. A. Yousuf, "Performance analysis of state of the art convolutional neural network architectures in bangla handwritten character recognition," *Pattern Recognition and Image Analysis*, vol. 31, no. 1, pp. 60–71, 2021. DOI: 10.1134/S1054661821010089.

[21] S. Kumar, V. K. Singh, and A. Ranjan, "Bengali handwritten text recognition using encoder-decoder and lstm-ctc models," in *Proc. 2021 12th Int. Conf. on Computing, Communication and Networking Technologies (ICCCNT)*, 2021. [Online]. Available: https://ieeexplore.ieee.org/document/9579690.

[22] S. P. Mohanty, D. P. Hughes, and M. Salathé, "Using artificial intelligence to improve accessibility for the visually impaired," *IEEE Access*, vol. 9, pp. 12 145–12 158, 2021.

[23] F. B. Safir, A. Q. Ohi, M. F. Mridha, M. M. Monowar, and M. A. Hamid, "End-to-end optical character recognition for bengali handwritten words," *arXiv preprint arXiv:2105.04020*, 2021. [Online]. Available: https://arxiv.org/abs/2105.04020.

[24] M. R. Sultan, M. M. Hoque, F. U. Heeya, I. Ahmed, M. R. Ferdouse, and S. M. A. Mubin, "Adrisya sahayak: A bangla virtual assistant for visually impaired," in *2021 2nd Int. Conf. on Robotics, Electrical and Signal Processing Techniques (ICREST)*, 2021, pp. 493–497. DOI: 10.1109/ICREST51555.2021.9331080.

[25] Coqui.ai, *Coqui TTS: A deep learning toolkit for text-to-speech*, GitHub Repository, [Online]. Available: https://github.com/coqui-ai/TTS, 2022. [Online]. Available: https://github.com/coqui-ai/TTS.

[26] L. Guezouli, "Reading signboards for the visually impaired using pseudo-zernike moments," *Advances in Engineering Software*, vol. 169, 2022. [Online]. Available: https://www.sciencedirect.com/science/article/abs/pii/S0965997822000382.

[27] K. V. Hegde, M. T. Marathe, and S. R. Kamath, "Real-time assistive text reading system for visually impaired using deep learning," *IEEE Access*, vol. 10, pp. 85 471–85 482, 2022. DOI: 10.1109/ACCESS.2022.3198840.

[28] R. Islam, M. R. Islam, and K. H. Talukder, "An efficient roi detection algorithm for bangla text extraction and recognition from natural scene images," *Journal of King Saud University - Computer and Information Sciences*, vol. 34, no. 9, pp. 6150–6164, 2022. DOI: 10.1016/j.jksuci.2022.01.026.

[29] A. Rahman, M. S. Islam, and S. Haque, "Assistive technologies for visually impaired people in developing countries: A review," *IEEE Access*, vol. 10, pp. 41 218–41 235, 2022. DOI: 10.1109/ACCESS.2022.3166725.

[30] O. Sen et al., "Bangla natural language processing: A comprehensive analysis of classical, machine learning, and deep learning-based methods," *IEEE Access*, vol. 10, pp. 38 999–39 046, 2022. DOI: 10.1109/ACCESS.2022.3164937.

[31] M. S. I. Toaha et al., "Automatic signboard detection and localization in densely populated developing cities," *arXiv preprint arXiv:2003.01936v4*, Aug. 2022. [Online]. Available: https://arxiv.org/abs/2003.01936v4.

[32] T. L. S. Vidhya and A. R. Thangadurai, "Lightweight deep learning model deployment on edge devices using tensorflow lite and raspberry pi," *IEEE Access*, vol. 10, pp. 74 729–74 739, 2022. DOI: 10.1109/ACCESS.2022.3189658.

[33] Z. Zhang, C. Zhang, and W. Liu, "Real-time scene text detection and recognition in natural environments," *IEEE Transactions on Image Processing*, vol. 31, pp. 2021–2035, 2022.

[34] R. B. Adams and K. K. Ellis, "User-centered AI: Designing human-assistive systems for accessibility," *IEEE Transactions on Human-Machine Systems*, vol. 53, no. 3, pp. 412–424, Jun. 2023. DOI: 10.1109/THMS.2023.3245671.

[35] H. Ali et al., "Gold standard bangla ocr dataset: An in-depth look at data preprocessing and annotation processes," in *Proc. of the 2023 Conf. on Empirical Methods in Natural Language Processing: Industry Track*, Dec. 2023, pp. 460–470.

[36] M. N. Hridoy, S. Chakraborty, and M. S. Uddin, "bbOCR: A benchmark dataset for bangla text detection and recognition in natural images," *IEEE Access*, vol. 11, pp. 60 354–60 366, 2023. DOI: 10.1109/ACCESS.2023.3280123.

[37] R. Kundu et al., "IndicSTR: A benchmark dataset for scene text recognition in 12 Indian languages," in *Proc. International Conference on Document Analysis and Recognition (ICDAR)*, IEEE, 2023, pp. 150–159. DOI: 10.1109/ICDAR56305.2023.00032.

[38] H. Lunia, A. Mondal, and C. V. Jawahar, "Indicstr12: A dataset for indic scene text recognition," in *ICDAR 2023 Workshops*, ser. Lecture Notes in Computer Science, vol. 14193, Springer, 2023, pp. 233–250. DOI: 10.1007/978-3-031-41498-5_17.

[39] H. Murad and M. E. Ali, "Towards detecting, recognizing, and parsing the address information from bangla signboard: A deep learning-based approach," *arXiv preprint arXiv:2311.13222*, 2023.

[40] I. M. Zulkarnain et al., "Bbocr: An open-source multi-domain ocr pipeline for bengali documents," *arXiv preprint arXiv:2308.10647*, 2023. [Online]. Available: https://arxiv.org/pdf/2308.10647.

[41] P. M. Bansal, P. Khurana, and R. R. Shah, "Indic-STR12: A large-scale multilingual dataset for scene text recognition in 12 indic languages," in *IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, 2024, pp. 1208–1217. DOI: 10.1109/WACV57701.2024.00125.

[42] Ethnologue, *Most spoken languages in the world*, Ethnologue: Languages of the World, [Online]. Available: https://www.ethnologue.com, 2024. [Online]. Available: https://www.ethnologue.com.

[43] Ethnologue, *What are the top 200 most spoken languages?* SIL International, [Online]. Available: https://www.ethnologue.com/guides/ethnologue200, 2024. [Online]. Available: https://www.ethnologue.com/guides/ethnologue200.

[44] Google Cloud, *Cloud text-to-speech API documentation*, Google Developers, [Online]. Available: https://cloud.google.com/text-to-speech, 2024. [Online]. Available: https://cloud.google.com/text-to-speech.

[45] A. S. A. Rabby, H. Ali, M. M. Islam, S. Abujar, and F. Rahman, "Enhancement of bengali ocr by specialized models and advanced techniques for diverse document types," in *Proc. IEEE/CVF Winter Conf. Applications of Computer Vision Workshops (WACVW)*, 2024, pp. 1102–1109.

[46] Raspberry Pi Foundation, *Raspberry pi documentation*, Raspberry Pi Foundation, [Online]. Available: https://www.raspberrypi.org/documentation/, 2024. [Online]. Available: https://www.raspberrypi.org/documentation/.

[47] S. Sharmin, T. Mim, and M. M. Rahman, "Bangla optical character recognition for mobile platforms: A comprehensive cross-platform approach," *Am. J. Electr. Comput. Eng.*, vol. 8, no. 2, pp. 31–42, Sep. 2024. DOI: 10.11648/j.ajece.20240802.12.

[48] M. R. Ullah, M. N. Mahbub, M. A. Hakim, Y. Sultana, and I. Alam, "Disha: A bilingual humanoid virtual assistant," in *2024 6th Int. Conf. on Electrical Engineering and Information & Communication Technology (ICEEICT)*, 2024, pp. 838–843. DOI: 10.1109/ICEEICT61959.2024.10534321.

[49] T. Mazumder, F. Nusrat, A. B. S. Mahi, J. B. Brishty, R. Rahman, and T. Helaly, "Automated and efficient bangla signboard detection, text extraction, and novel categorization method for underrepresented languages in smart cities," *Results Eng.*, vol. 26, p. 105 156, May 2025. DOI: 10.1016/j.rineng.2025.105156.