Department of Electrical and Computer Engineering
North South University

Senior Design Project

Real-Time Bengali Sign Language Recognition and Bilingual Text-to-Speech Conversion Using Machine Learning Techniques

Group Member and Details

Name	ID
Yeastic Hassan Zihan	2021328042
Iffaat Ara Mehnaz	2031340042
Anisha Shawana Sharif	2022157042
Nusrat Jahan Tisha	2022478042

Faculty Advisor:
Dr. Mohammad Rashedur Rahman
Professor
Department of Electrical and Computer Engineering
North South University, Dhaka, Bangladesh.

Summer, 2024

LETTER OF TRANSMITTAL

November 2024

To

Dr. Mohammad Abdul Matin

Chairman, Department of Electrical and Computer Engineering

North South University, Dhaka

Subject: "Real-Time Bengali Sign Language Recognition and Bilingual Text-to-Speech Conversion Using Machine Learning Techniques"

Dear Sir,

Respectfully, as part of our BSc program, we would like to submit our capstone project report on “Real-Time Bengali Sign Language Recognition and Bilingual Text-to-Speech Conversion Using Machine Learning Techniques.” The project focuses on developing an AI-powered system that translates Bengali Sign Language into text and speech in real-time, supporting both Bengali and English languages. This solution aims to enhance communication for the Bengali-speaking deaf community, leveraging machine learning techniques to achieve high accuracy and efficiency. Throughout this project, we gained significant hands-on experience, applying theoretical knowledge to practical challenges.

We have endeavored to meet all the necessary requirements and have given our best effort to deliver a comprehensive and high-quality report.

Please review our report and provide your valuable feedback. We hope that our work effectively addresses the communication gap and that you find it informative and insightful.

Sincerely Yours,

Yeastic Hassan Zihan
ECE Department
North South University

Iffaat Ara Mehnaz
ECE Department
North South University

Anisha Shawana Sharif
ECE Department
North South University

Nusrat Jahan Tisha
ECE Department
North South University

APPROVAL

Yeastic Hassan Zihan (ID# 2021328042), Iffaat Ara Mehnaz (ID# 2031340042), Anisha Shawana Sharif (ID# 2022157042) and Nusrat Jahan Tisha (ID# 2022478042) from the Electrical and Computer Engineering Department of North South University, have worked on the Senior Design Project titled "Real-Time Bengali Sign Language Recognition and Bilingual Text-to-Speech Conversion Using Machine Learning Techniques" under the supervision of Dr. Mohammad Rashedur Rahman, which has been accepted as satisfactory and satisfied the requirement for the degree of Bachelor of Science in Computer Science and Engineering.

Supervisor's Signature

…….

Dr. Mohammad Rashedur Rahman

Professor

Department of Electrical and Computer Engineering

North South University

Dhaka, Bangladesh.

Chairman's Signature

…….

Dr. Mohammad Abdul Matin

Professor

Department of Electrical and Computer Engineering

North South University

Dhaka, Bangladesh.

DECLARATION

We hereby certify that this work is entirely our own and has not been submitted elsewhere, either in full or in part, for any other degree or diploma. All project-related information will remain confidential and will not be disclosed without the prior approval of our project supervisor. All previous works referenced in this report have been properly acknowledged, and we have adhered to the supervisor's anti-plagiarism guidelines throughout the preparation of this report.

1.Yeastic Hassan Zihan

---

2. Iffaat Ara Mehnaz

---

3. Anisha Shawana Sharif

---

4. Nusrat Jahan Tisha

ACKNOWLEDGEMENT

Dr. Mohammad Rashedur Rahman, Professor in the Department of Electrical and Computer Engineering at North South University in Bangladesh, is sincerely acknowledged by the authors for his invaluable guidance, insightful feedback, and unwavering support throughout the research, theoretical studies conducted during the current project and in the preparation of current report.

Furthermore, the authors also express their gratitude to the Department of Electrical and Computer Engineering at North South University for providing the resources and support needed to carry out this project. Finally, we extend our heartfelt thanks to our families for their constant encouragement, sacrifices, and steadfast support.

ABSTRACT

"Real-Time Bengali Sign Language Recognition and Bilingual Text-to-Speech Conversion Using Machine Learning Techniques "

Bengali Sign Language (BdSL) is a vibrant language used by the Deaf community in Bangladesh, relying on hand, facial, and body movements to convey messages. While it effectively bridges communication barriers within the deaf community, its widespread use for effective conversation which is limited due to unfamiliarity among the larger population. This research leverages advanced machine learning and deep learning to develop a real-time BdSL recognition system using a Kaggle-sourced dataset of 1,200 RGB images resized to 128x128 pixels. Six deep learning models are tested, with MobileNetV2 achieving the highest accuracy of 99.89% on training data and 95.85% on testing data. In contrast, InceptionResNetV2, InceptionV3, and Xception exhibited overfitting, evidenced by significant drops in testing accuracy compared to training accuracy. The system provides a practical solution for enhancing accessibility and communication for the Deaf community in Bangladesh. This work is substantially different than the existing work where each sample contains complete sentences, and the research recognizes a specific target at the character level. It includes a bilingual text-to-speech conversion facility. The system converts recognized signs to English and Bengali and widens the accessibility and overcomes language barriers.

Table of Contents

List of Figure

List of Tables

CHATPER 1 INTRODUCTION

1.1 Background and Motivation

A vital component of connecting with others is through communication, yet millions face profound obstacles due to hearing loss. Over 35 million people of Bangladesh use Bengali Sign Language (BdSL) as a medium of communication [1]. Called BdSL, this system of gestures and movements inspired by the sound language of the region allows people with speech and hearing disabilities to articulate complex ideas. But the absence of automated systems for BdSL recognition [2] presents a major issue and hinders accessibility and inclusivity. Here, Bengali Sign conveys the standard tool for speech and hearing-impaired persons to link with one another or outsiders who shun signs. Surprisingly, a symbol sometimes embodies many meanings across languages. Signs speak through motions, positional changes, and hand formations. Whereas finger-spelling encodes letters, full sentences necessitate continual hand movements, gestures, and configurations. Sign complexity varies extensively, allowing nuanced, intricate communication of motions. Signs may elaborate as desired. Characterizing a single emotion or experience could integrate body position, lip motions, facial expressions, hand signs, head movements, and so on.

To date, most interpretation relies on human translators or basic mechanical aids, both containing shortfalls like inaccessibility. Technological progress and particularly convolutional neural networks, machine learning, and deep learning opened automation of sign detection. This may widely spread service to deaf populations, so Bengali Sign study represents a highly important field of development. Sign language detection systems apply cameras and sensors to capture video and extract signs for interpretation and conversion to text and speech. However, sign detection research impact slowly grows, contemporary methods possess several imperfections. Variations in gestures between signers, differences in individual signers, ambient conditions such as lighting and potential noise, hand angle, all challenge high accuracy. Few labeled Bengali Sign databases exist, a major hindrance because models train on other sign languages and thus Bengali Sign recognition falls short even when complex gestures map to others.

1.2 Purpose and Goal of the project

The primary goal of this study is to bridge the communication barrier between the hearing-impaired community and broader society through creating an innovative system for Real-Time Bengali Sign Language (BSL) Recognition [2] and bilingual text-to-speech conversion. This ambitious project aims to address the lack of accessibility and inclusion for deaf individuals in Bangladesh using machine learning techniques to recognize BdSL gestures as they happen and translate them into textual and vocal outputs in both Bengali and English.

The research aspires to engineer a scalable and efficient solution that facilitates seamless interaction, empowering the deaf community to communicate their ideas, thoughts, and emotions more effectively. By leveraging advanced deep learning models and bilingual text-to-speech technology, the system hopes to provide a robust and adaptable communication tool, fostering greater inclusion in educational, professional, and social settings. Moreover, the research highlights the potential of cutting-edge machine learning and human-computer interaction technologies to make a meaningful impact on society, addressing the unique linguistic and cultural needs of the Bengali-speaking deaf population.

This study proposes a novel machine learning based sign language detection (SLD) system using deep learning techniques for the hearing-impaired population to help bridging the communication gap among the society. The aim of the study to accomplish accurate recognition BdSL realizing techniques as One-Hot Encoding, data preprocessing, pretrained networks like MobileNetV2. Our goal is to provide a scalable text to speech (TTS) solution for Bengali and English languages so that we can spread the inclusion and accessibility among the hearing-impaired people of Bangladesh.

In conclusion, the following summarizes our main contributions to this study:

1. What technical and computational aspects need to be considered to guarantee the real-time performance of Bengali Sign Language recognition while maintaining the trade-off between latency, accuracy, and resource consumption in an edge environment?
2. How does the trade-off between Inference Accuracy and Training Accuracy for all the models (MobileNetV2 [3], ResNet50V2, and DenseNet121) provide an insight into further improvement into either of the folds of the model and also helps us to build an optimized hybrid or ensemble model thereof?
3. What is the impact of the user interface design features, sign language input, and bilingual text-to-speech on user accessibility and engagement, and what potential features could be added to improve inclusivism for all user demographics?
4. The question could delve into advanced data augmentation or generative models, like GANs, to simulate various scenarios in sign languages, especially given the size and variability limitations of the dataset used, and discuss potential biases or skewness that might be introduced by using synthetic data.
5. How the project can deploy an AI-based Bengali Sign Language recognition system and Ethical implications of deploying the system — specifically, how they can address data collection and privacy issues in ensuring inclusivity, and further, if the project will amplify existing socio-economic inequities for the Deaf community or counteract them all while adhering to an equitable access framework?

CHAPTER 2 Research Literature Review

Real-time Bengali Sign Language identification (BSL) has evolved as a vital research area that plays a crucial role in breaking communication issues for the deaf/hard-of-hearing people. A plethora of advanced machine learning and deep learning paradigms like convolutional neural networks (CNNs), recurrent neural networks (RNNs) and their hybrid architectures have been explored to improve gesture recognition accuracy [4]. Nonetheless, the challenges of sign language recognition are nontrivial, whereby the high variability in sign language articulation [5], variations in lighting, and the everchanging background environments represent significant hurdles to overcome. Recent studies have been focusing more on the performance of multimodal methods that combine video-based analysis with other sensor data to improve the robustness and generalization of the recognition systems. In spite of the advances in methodology, challenges still remain in the domain, primarily in the form of limited rich, annotated dataset coverage in BSL, the complexity of the phonetic interactions across Bengali (and BSL) that cause recognition tasks to become even more difficult. Future directions revolve around building more diverse and complete datasets, designing context-aware models and developing compact architectures suitable for real-time applications. Resolving these complex challenges is critical to advance the creation of scalable, low-cost, and contextually robust BSL detection systems [6].

2.1 Existing Research

The population with hearing and speech impairments are unable to use the vocal or auditory spoken languages. To make matters worse, there is also a significant diversity in the composition and variations of sign languages across the globe, as they differ significantly from nation to nation. Around 300 unique sign languages are being used on the planet, among others American Sign Language (ASL), Japanese Sign Language, Chinese Sign Language, French Sign Language, Spanish Sign Language, Bangladeshi Sign Language (BdSL) [7]. Among these, ASL has been the most widely studied and best studied linguistically. Learning sign languages is a difficult for hearing impaired people as well as non-impaired population, and this difficulty is more prevalent in Bangladesh, where BdSL [8] is mostly considered as an inherent difficult one. And thus, individuals with hearing and speech impairments are so far away from mainstream society because of such communication barriers [9].

Sign language recognition is an innovative idea that has been around for a few decades but has recently gained attention in the modern world along with the development of computer vision. Since it is impractical to always have human interpreters available, scientists and researchers are increasingly focused on creating automated systems that are able to accurately detect sign languages. Sign language recognition is a complex

task by nature, so researchers tried many approaches, such as SVM [10], HMM, KNN, ANN, and especially CNN [11]. Among them, CNNs have been the most mainstream because they can extract very complex features from image and video data. Therefore, they are very suitable for image processing tasks such as sign language recognition [12]. However, BdSL research efforts are still relatively modest in regard to breadth and magnitude. The linguistic complexity of BdSL, with 38 symbols (9 vowels and 27 consonants), is one of the main challenges that come with it, adding extra layers of complexity [13]. Although some high-profile studies have focused on BdSL, the amount of research remains considerably smaller than for other, better-researched sign languages.

2.2 Existing Research Limitations

Table 1 Related Works in Real-time Bengali Sign Language Recognition

Ref No	Research	Model used	Dataset	Findings	Limitation /Research Gap
[8]	Deep Learning-based Bangla Sign Language Detection with an Edge Device	Detectron2, EfficientDet-D0 with TensorFlow and the PyTorch-built YOLOv7 Tiny models. Jetson Nano, and sufficiently dynamic NVIDIA microprocessor	A public dataset for Bangla sign language. Features 49 classes (39 letters and 10 digits) with over 110 images per class. Includes diverse lighting conditions and backgrounds. Developed by the authors with 49 classes and ~80 images per class. Created with various lighting conditions, angles, and hand orientations. Collected from 11 volunteers over three months.	Detectron2 achieved the highest accuracy (mAP@0.5: 94.915). YOLOv7 Tiny was optimal for real-time use on Jetson Nano (mAP@0.5: 94.2). A custom dataset of 49 classes enhanced performance. Real-time detection achieved 24 FPS on Jetson Nano. Cost-effective system (~$230 USD).	High computational demands for Detectron2. Limited dataset diversity for complex gestures. No facial gestures or complex sign combinations. Requires further optimization for lower-power devices. Limited scalability for large datasets.

[4]	Recognition of Bangladeshi Sign Language (BdSL) Words using Deep Convolutional Neural Networks (DCNNs)	DenseNet201 (Achieved the best performance (93% test accuracy)), ResNet50-V2 (Similar high accuracy as DenseNet201), Modified CNN, MobileNet-V2 (Lower performance compared to DenseNet201 and ResNet50-V2)	Made by authors (private dataset) Size: 992 images. Classes: 10 frequently used Bangladeshi Sign Language (BdSL) words: Examples: "Color," "Friend," "Myself," "Promise," "Request," etc. Split: Training set: 794 images. Testing set: 198 images.	Developed an automated system using CNNs for recognizing Bangladeshi Sign Language (BdSL) words. DenseNet201 and ResNet50-V2 achieved the best accuracies: 99% training and 93% testing. The system processes images to detect and classify 10 frequently used BdSL words. Real-time word detection is proposed for practical applications.	The dataset size was limited to 992 images, which may affect the model's generalizability. The study focused on specific BdSL words; it lacks coverage of the entire sign language vocabulary. No exploration of real-world conditions like diverse lighting and complex backgrounds.
[3]	Real Time Sign Language Detection	CNN, Pre-Trained SSD MobileNet V2	A collection of over 2000 images, around 400 for each of its classes. This dataset contains a total of 5 symbols i.e., Hello, Yes, No, I Love You and Thank You, which is quite useful while dealing with the real time application.	Utilized SSD MobileNet V2 with transfer learning for real-time sign language detection. Focused on recognizing 5 symbols ("Hello," "Yes," "No," "I Love You," and "Thank You") with 70–80% accuracy under uncontrolled backgrounds. Employs a pipeline including OpenCV and image segmentation to preprocess and classify gestures.	Accuracy is relatively low compared to controlled experiments, especially under environmental disturbances like lighting and camera positioning. Limited to only five symbols, which restricts the application scope. Background and illumination variability are challenges that impact the system's reliability.

[13]	BANGLA SIGN LANGUAGE RECOGNITION USING CONCATENATED BdSL NETWORK	Convolutional Neural Network (CNN) for visual feature extraction. Features: 10 convolutional layers with ReLU activation. 4 max-pooling layers. Batch normalization after each layer. Input size: 64 x 64 grayscale images. Outputs 21 keypoints (each with x and y coordinates), flattened into a 1x42 vector.	Collected by students from the Bangladesh National Federation of the Deaf (BNFD). 38 labeled classes (including 9 vowels and 27 consonants of Bangla alphabets). Images normalized to 64 x 64 pixels, with grayscale for the image network and BGR format for pose estimation.	Proposed a novel "Concatenated BdSL Network," combining CNN and pose estimation for recognizing Bangla Sign Language (BdSL). Achieved 91.51% accuracy on the test set by extracting both visual and hand pose features. Addressed subtle gesture variations in BdSL, demonstrating better performance than other CNN-only models.	Misclassification occurs for nearly identical symbols (e.g., [উ-ঢ], [ঙ-ঞ]), indicating the difficulty of distinguishing subtle gestures. Uses a pretrained OpenPose model, which limits specificity for BdSL. Computational resource constraints prevented more extensive experiments or the development of real-time systems.
[14]	Real Time Bangladeshi Sign Language Detection using Faster R-CNN	Convolutional Neural Network, Faster R-CNN	A dataset called BdSLImset containing images of Bangladeshi signs with random back- grounds and lighting conditions.	A dataset named BdSLImset was created with 10 classes, focusing on random backgrounds and lighting conditions. The proposed system achieved an accuracy of 98.2% and real-time detection in 90.03 milliseconds using Faster R-CNN. It demonstrated improved performance over other methods like Haar-like features and ANN in real-time recognition.	Difficulties were encountered in recognizing letters with similar gestures due to background variations and lighting conditions. Certain gestures, such as ‘ক’ and ‘গ’, had high misclassification rates due to their visual similarity. The dataset size and diversity need enhancement to improve robustness.

[10]	A Word-Level Bangla Sign Language Dataset	Support Vector Machine (SVM) with testing accuracy up to 67.6% and an attention-based bi-LSTM with testing accuracy up to 75.1%.	The dataset encompasses a total of 60 Bangla sign words, with a significant scale of 9307 video trials provided by 18 signers under the supervision of sign language professional. The dataset was rigorously annotated and cross-checked by 60 annotators.	Created a comprehensive dataset, BdSLW60, with 60 Bangla sign words and 9307 video trials. Includes variations like hand dominance and temporal gestures. Annotated by 60 contributors, ensuring accuracy. Utilized classical SVM and Attention-based Bi-LSTM. Achieved maximum accuracy of 75.1% with Bi-LSTM and a relative quantization-based encoding technique. Tackled the scarcity of datasets for Bangla sign language. Developed preprocessing steps (e.g., hand dominance correction, landmark calibration) to standardize data. Introduced a novel relative quantization-based keyframe encoding technique, aiding continuous recognition systems.	Limited to 60 words; the full dataset of 401 words is not yet available. Benchmark accuracy (75.1%) leaves room for improvement with more sophisticated models. Challenges with depth variation, hand rotation, and missing landmarks due to manual recording. Did not test extensively on sentence-level or continuous sign language.
[15]	Real-time Bangla Sign Language Translator	Recurrent Neural Network, LSTM, Computer Vision	PHOENIX- weather 2014T dataset. Collected sequential data for each word with 30 frames per word.	Used Mediapipe for feature extraction and LSTM for temporal data processing, achieving real-time translation. Gathered 30 frames per word, labeling them for training. Demonstrated potential for real-time applications, especially for aiding the deaf and mute community in communication. Extensive use of modern tools and	Faced difficulty in obtaining an extensive dataset; the system's performance may degrade with larger, more diverse vocabularies. Struggled to distinguish between visually similar gestures like “deer” and “educated.” Faced issues displaying Bangla fonts correctly,

				frameworks like TensorFlow, OpenCV, and Mediapipe.	though this was partially resolved. Experienced fluctuations in accuracy during initial training phases.
[1]	"Action Recognition Based Real-time Bangla Sign Language Detection and Sentence Formation"	BlazePose Algorithm, LSTM model, (CNN)	A total of 300 videos (or 9000 frames) were used to create the dataset for the ten labels. (Private dataset) Self-constructed dataset of 300 videos (30 per action label) corresponding to 10 Bangla sentences. Each video contains 30 frames, generating a total of 9000 frames, which were converted into NumPy files for training. Captured using a webcam in real-time.	The study used BlazePose for pose estimation and LSTM for recognizing sequential gestures, enabling sentence formation in BdSL. Achieved a training accuracy of 93.85% and validation accuracy of 87.14%. Introduced a dataset of 300 videos representing ten sentences, advancing BdSL sentence-level recognition.	The model struggles with overlapping gestures and interpreting incomplete sign sequences, common in real-world communication. The reliance on pose estimation may reduce accuracy in cases of occlusion or rapid motion. Limited to ten predefined sentences, restricting its adaptability to diverse real-life scenarios.
[5]	Borno net: A Real-Time Bengali Sign-Character Detection and Sentence Generation System Using Quantized Yolov4-Tiny and LSTMs	Convolutional Neural Network (CNN) models to detect BSL, K-Nearest Neighbors (KNN), Support Vector Machine (SVM), Long Short-Term Memory (LSTM), YoloV4, YoloV4 Tiny, and YoloV7	BdSL 49 dataset is used, which has approximately 14,745 images. 36 Bengali alphabet characters. 10 numeric characters. 3 special characters (Space, Compound Character, and End of Sentence). Images are in RGB format with a resolution of 128 × 128 pixels.	Developed a real-time system using a quantized YOLOv4-Tiny model for Bengali sign character detection. Incorporated LSTM for meaningful sentence generation. Trained on the BdSL 49 dataset, which consists of 14,745 images across 49 classes (36 alphabets, 10 numerals, and 3 special characters).	Limited to 49 classes of sign characters; does not include all possible Bangla signs or words. No testing on real-world edge devices, although the model is optimized for such use cases. Relies on predefined datasets and lacks generalization

				Achieved a mAP of 99.7% for detection and an accuracy of 99.12% for sentence generation. Optimized for lightweight deployment on edge devices with an average response time of 50ms (30ms for detection and 20ms for sentence generation).	testing with external datasets.
[6]	Automatic Recognition of Bangla Sign Language	Artificial Neural Networks (ANNs); Trained to recognize isolated Bangla sign language gestures using input data from Kinect.	Source: Custom dataset created by capturing isolated Bangla Sign Language signs. Used Kinect Depth Camera to capture gestures, extracting skeletal tracking information. Gestures performed by participants, including individuals trained in Bangla Sign Language. Focused on static signs representing individual Bangla alphabets and numbers.	Developed an approach using Kinect Depth Camera and Artificial Neural Networks (ANNs) for recognizing Bangla Sign Language. Focused on isolated signs with manual gestures to simplify the recognition process. Employed Kinect's skeleton tracking feature to extract depth information. Demonstrated the potential for improving communication between the hearing-impaired community and the general population.	Limited to isolated signs; lacks recognition of continuous sign language or contextual interpretation. No real-time implementation; its mainly about experimental. Dependent on specific hardware (Kinect), restricting accessibility.
[16]	BANGLA SIGN LANGUAGE RECOGNITION USING CONCATENATED BdSL NETWORK	Convolutional Neural Network (CNN)	Developed with support from the Bangladesh National Federation of the Deaf (BNFD). Focuses on Bangla Sign Language symbols with contributions from BNFD resources. Contains visually diverse images of Bangla sign symbols captured under various conditions.	Proposed a novel architecture combining a CNN-based "Image Network" and OpenPose-based "Pose Estimation Network." Addressed the challenge of distinguishing visually similar Bangla symbols by leveraging complementary features from the two networks. Used a dataset from the Bangladesh National Federation of the Deaf	Computationally intensive due to the dual network approach (CNN and OpenPose). Performance may degrade in real-world scenarios with diverse lighting and background conditions. Dataset used is limited in terms of diversity and real-world complexity.

			Includes key point annotations for accurate hand posture detection using the OpenPose network. No explicit mention of the exact size, but the dataset is tailored for training and evaluating the dual-network architecture.	(BNFD), incorporating diverse Bangla sign symbols. Improved recognition accuracy over traditional models by combining key point-based and pixel-based features.	Difficulty in distinguishing symbols with subtle differences, such as similar hand poses.
[7]	Creating Multiclass Bangladeshi Sign Word Language for Deaf and Hard of Hearing People and Recognizing using Deep CNN Techniques	Deep Convolutional Neural Network (DCNN), CNN model, SVM	Self-constructed dataset of 1105 images representing 11 Bangladeshi sign words. Augmented to 3835 images using techniques like rotation, zooming, and flipping. Images were captured from hearing-impaired and general people using a Samsung Galaxy G7 camera with a consistent background.	Created a new dataset of 11 Bangladeshi static sign words with 1105 images, later augmented to 3835 images. Proposed a CNN-based model that achieved a high training accuracy of 99.12%, testing accuracy of 84.67%, and validation accuracy of 80.54%. The Grad-CAM visualization confirmed the model's focus on relevant parts of the images for classification.	The dataset is limited to static signs, not covering dynamic gestures or continuous sign sequences. Background and lighting inconsistencies in images can reduce model accuracy. The model does not account for variations in hand gestures across different signers.
[17]	ML-BASED HAND SIGN DETECTION SYSTEM FOR DEAF-MUTE PEOPLE	SSD MobilenetV2 FPNLite 320x320 Transfer learning TensorFlow object detection API.	Self-constructed dataset but doesn't mention about the size and the taking pictures	After evaluation, the precision rate of the model came out to be 69% and the recall rate came out to be 70%, which is not a great result but still it's a start. And also, it was observed that the increase in the number of images and adding a	The researchers are still capturing still images, but in the future, they can train our model to capture more complex live hand signs that require more than one gesture. Also, researcher plan to add many other

				more and more different variety of data from various angles also increased the stats significantly. Even after that, as seen in figure 5, the detection is being performed with a high accuracy rate.	features such as performing certain daily life actions using hand gestures or voice commands. And also, this can be also incorporated into a mobile application which would make it easier to use.
[2]	Bangladeshi Sign Language Recognition employing Neural Network Ensemble	Neural Network Ensemble (NNE)	Self-constructed dataset of 235 images corresponding to 47 Bangladeshi sign gestures, with 5 samples per gesture. Images were captured using a webcam and preprocessed (thresholding, normalization, feature extraction) for training.	Proposed the Bangladeshi Sign Language Recognizer (BdSLR) using Neural Network Ensemble (NNE) and Negative Correlation Learning for better accuracy. Achieved a recognition accuracy of up to 93% using feature extraction and ten neural networks in the ensemble. Demonstrated the system's potential to interpret BdSL for social inclusion of deaf and mute individuals.	The system focuses on a small dataset of 47 signs, limiting scalability. Image capturing through a webcam can be cumbersome, with performance affected by environmental factors like lighting and color interference. Real-time recognition was not fully implemented, remaining a goal for future development.
[18]	Real Time Bangla Sign Language Detection using Action Recognition	LSTM, CNN, ANN	King Saud University Saudi Sign Language (KSU-SSL) dataset by Muneer Al-Hammadi et al.	Utilized a combination of optical flow and skeletal joint data extracted from depth maps to identify Bangla Sign Language gestures. Integrated deep learning techniques, specifically Long Short-Term Memory (LSTM) networks, for sequential action recognition. Achieved a high precision of 94% on a custom dataset of Bangla Sign Language videos.	Relied on depth-based data, which requires specific sensors, potentially increasing implementation costs. Performance depends on the quality of depth maps and controlled environments. Dataset size and diversity might limit generalizability to broader use cases.

[12]	Vision-based Real Time Bangla Sign Language Recognition System Using Media Pipe Holistic and LSTM	LSTM, CNN	Made by authors (private dataset)	Combined MediaPipe Holistic for landmark extraction and LSTM networks for gesture recognition. Created a custom dataset of 15 gestures, with 30 videos per gesture, for training and testing. Demonstrated real-time recognition capabilities with an emphasis on efficient processing and ease of use.	The dataset was limited to only 15 gestures, restricting the scope of application. Struggles with complex and dynamic real-world backgrounds or lighting variations. Dependent on MediaPipe's performance, which may vary across different devices and environments.
[19]	Bangla Sign Language Recognition and Sentence Building Using Deep Learning	Hidden Markov Model (HMM), CNN,	some datasets from Kaggle and GitHub	The paper proposed a machine-based approach using Convolutional Neural Networks (CNN) to recognize Bangla Sign Language (BdSL). It successfully developed a system that could recognize hand signs and translate them into meaningful Bangla sentences. The model achieved high accuracy in recognizing individual signs and combining them into full sentences, enhancing communication for the deaf and mute community.	The system struggled with recognizing very similar signs, leading to occasional misclassification. The dataset used was limited in size and variability, which may reduce the model's generalization ability in real-world scenarios. Real-time performance was not thoroughly tested across diverse conditions such as lighting, background variations, or different hand orientations.
	Recognition Bangla Sign Language using Convolutional Neural Network	Convolutional Neural Network (CNN); Activation Function: ReLU (Rectified Linear Unit) for hidden layers and Softmax for the output layer.	BdSL Dataset consisting of 30,916 samples. 23,864 samples (35 Bangla alphabet characters). 7,052 samples (10 numerals). Images were collected from 25 students (age 18-26 years) with hearing or	The authors created a large dataset of Bangla sign language consisting of 30,916 samples (including both alphabets and numerals). The proposed CNN model achieved very high accuracy, with 100% accuracy on	The dataset was collected from a relatively homogenous group (age 18-26 years, all boys), which could limit its general applicability across different age groups or genders.

[9]		Pooling: Max-pooling was used with a window size of 2×2 and a stride size of 2. Dropout Layer: Included to reduce overfitting. Global Average Pooling (GAP): Used to perform dimensionality reduction. Optimizer: Adam optimizer with a learning rate of 0.001. Loss Function: Cross-Entropy Loss. Training: The model was trained for up to 200 epochs with a batch size of 128 and 10-fold cross validation.	speaking disabilities at the Sweet Dream School, Kushtia, Bangladesh.	numerals and 99.83% on Bangla characters. The research highlighted the potential of using deep learning for automatic Bangla sign language recognition and demonstrated the robustness of CNN for this task.	The study focused only on static images for sign recognition and did not address dynamic gestures or continuous signing, which are critical in real-life scenarios. No detailed comparison with real-time systems or the impact of environmental factors on the model's performance was presented.
[20]	A Comprehensive Dataset of Bangla Sign Language	YOLOv4; Mean Average Precision (mAP): 99.9% on the detection dataset. Annotation Process: Each image was manually annotated to create bounding boxes around hand signs, saved in YOLO format. Xception: Achieved the highest accuracy (93%) for recognizing Bangla sign language. Other models used: -InceptionV3 -InceptionResNet	BDSL 49 Dataset consisting of 29,490 images across 49 Bangla labels (including alphabets, numerals, and special characters). YOLOv4 Detection Model: 99.9% mean average precision. Recognition Models: Xception (93%) achieved the highest accuracy.	The dataset, named BDSL49, consists of 29,490 images with 49 Bangla alphabet labels, making it one of the most comprehensive datasets for BdSL. The study implemented YOLOv4 for detection and several CNN models for recognition, with the Xception model achieving the highest accuracy (93%). The dataset is freely available for further research, encouraging more studies in this field.	Although the dataset covers 49 labels, it does not address the recognition of dynamic signs or sentence-building, which are important for practical applications. The dataset was collected using smartphone cameras, which may introduce inconsistencies in image quality. The study did not explore the usability of the dataset for real-time applications or across various devices and platforms.

		V2 -ResNet50V2 -DenseNet			
[11]	Real-Time Bangla Sign Language Detection with Sentence and Speech Generation	YOLOv4 The system utilizes the YOLOv4 object detection model for Bangla Sign Language (BdSL) detection.	Size: 12.5k images. Classes: 49 different signs, including: 39 Bangla alphabets. 10 Bangla digits. 3 new proposed signs for compound characters, space, and sentence-ending punctuation. Images captured under diverse conditions: Different lighting: bright sunlight, artificial lighting, shadows. Various backgrounds and image qualities. Captured using four devices to ensure variability. Images sourced from 8 individuals (3 females and 5 males), aged 17–68, ensuring diversity. Images were annotated in YOLO format using the LabelImg tool, generating bounding boxes for each class.	A dataset of 12.5k images covering 49 different signs (39 Bangla alphabets, 10 digits, and 3 newly proposed signs) was developed. Images were captured under various lighting conditions with diverse participants to enhance robustness. The system used YOLOv4 for real-time BdSL detection due to its efficiency and ability to detect objects at multiple scales. The detection accuracy reached 97.95%, surpassing existing systems. A sequence of detected signs was processed into Bangla words and sentences using proposed signs for space, compound characters, and sentence-ending punctuation. Speech generation was implemented using the Google Text-to-Speech (gTTS) API, producing both real-time audio and .mp3 files for future use. The system achieved a response time of 33 milliseconds per frame, enabling fast real-time interaction. The system demonstrated high performance, with precision, recall, F1 scores, and accuracy consistently high across signs, digits, and alphabets.	Only 36 of the 49 Bangla alphabets have corresponding signs, making it impossible to generate some commonly used words. No signs are available for punctuation other than the Bangla period (।). The system occasionally confused similar-looking signs, especially between: ম and ন ৩, ৪, and ৫ অ and ল. The absence of a sign for the decimal point hinders the generation of floating-point numbers. Proper sentence construction is restricted due to the absence of signs for various punctuation marks.

[21]	A comparative analysis between single and dual-handed Bangladeshi Sign Language Detection Using CNN Based Approach	The study employs CNN based approaches, specifically utilizing pre-trained models such as VGG16, ResNet50, MobileNetV2 for the classification and detection of Bangladeshi Sign Language (BdSL).	KU-BdSL Dataset: A dataset of single-handed Bangladeshi Sign Language gestures, containing images of 30 consonants. BDSL49 Dataset: A comprehensive dataset of dual-handed Bangladeshi Sign Language gestures, containing images of both vowels and consonants.	For KU-BdSL dataset, the best accuracy achieved is 90% using VGG16. For BDSL49 dataset, accuracy results are similarly presented for the same models but not fully extracted from the text here.	Limited dataset diversity may not cover all Bangladeshi Sign Language variations. Pre-trained CNN models might lack full domain specific adaptability. Real-world application challenges, such as varying lighting and hand-positions, are not addressed.

2.3 Overcome of the existing research limitations

Our investigation uncovers an innovative amalgamation of cutting-edge procedures strategically aligned to tackle the multifaceted hurdles inherent in Real-time Bengali Sign Language (BdSL) perception. Distinct from standard frameworks predominantly relying on singular designs or constrained theoretical structures, our method skillfully incorporates state-of-the-art deep learning architectures—such as MobileNetV2, InceptionV3, and ResNet50V2—synergistically combined with information expansion procedures and One-Hot Encoding to elevate execution metrics. This diverse methodology is carefully intended to navigate the intricate subtleties of BdSL gestures, particularly the flexibility in hand shapes and movement elements. What isolates this examination is its unprecedented spotlight on word-level information sets, empowering granular acknowledgment of singular words while in the meantime encouraging the development of semantical consistent of 2-3-word sentences. Adding another measurement to its uniqueness, the examination incorporates bilingual content-to-speech translation, where perceived marks are interpreted from Bengali to English, along these lines making a strong connection between the two lingual spaces. This component distinctively separates our methodology from conventional models, which regularly stay monolingual or lack incorporated speech amalgamation capacities. Furthermore, by applying standards of transfer learning and ensemble learning, we significantly improve acknowledgment exactness and framework strength, accomplishing these upgrades without requiring the accessibility of broad information sets or exhaustive hyperparameter tuning. This layered strategy adeptly overcomes commonplace limitations connected with BdSL perception frameworks, particularly those identified with scalability and flexibility in real-time applications. At long last, our method offers a transformative, scalable answer for successfully advancing open doors and supporting more comprehensive correspondence pathways for the deaf local area in Bangladesh.

CHAPTER 3 Methodology

3.1 Data & Resources

This section outlines the proposed Bengali Sign Language (BdSL) recognition system, covering key aspects such as dataset acquisition (source, structure, and relevance) and preprocessing methods to optimize data for analysis. It details the implementation of deep learning models, focusing on their design, modifications for BdSL gestures, and the supporting hardware infrastructure. Additionally, it discusses the algorithms used, their integration, and the software components for data processing, model training, and deployment, providing a comprehensive overview of the system's technical and methodological foundations.

3.1.1 Dataset Breakdown

The Bengali Sign Language corpus, compiled from publicly available data on Kaggle, was one we opted to investigate owing to the lack of prior work conducted utilizing this modest collection. It represents a carefully indexed assortment of 1,200 photographs, deliberately curated to comprehensively represent the 40 distinguishable sign language classes native to Bengali, with each having its own folder containing 40 representative samples, hence assuring equitable provision across all categories so as to cultivate impartial and balanced training for our models. Each image has undergone standardization procedures, being resized to 128x128 pixels and saved in RGB format, harnessing the three-color dimensions to sufficiently seize the intricate visual subtleties critical to accurate identification. For model development and assessment purposes, the corpus has been methodically apportioned into two distinct subsets, with four-fifths of the images committed to training, permitting the models to effectively deduce and extrapolate the intricate patterns innate to Bengali sign language gestures, while the residual fifth is reserved as an evaluation pool, serving as a rigorous yardstick to gauge model performance when encountered with unprecedented information. This balanced and rigorously organized corpus, with its equitable dispersion across 40 samples and strategic allocation for education and testing, furnishes an ideal foundation for advancing machine learning applications in sign language recognition, ensuring fair learning processes while mitigating potential biases.

3.1.2 Dataset Collaboration

This research leverages advanced machine learning and deep learning to develop a Real-time BdSL recognition system using a Kaggle-sourced dataset. This collaboration involved curating labeled images representing various Bengali Sign Language signs, ensuring a comprehensive and inclusive dataset. To enhance its quality, duplicate and corrupted images were identified and removed. Additionally, data augmentation techniques were applied to the images to introduce variations, such as rotations, flips, and

brightness adjustments. This process enriched the dataset, enabling it to capture different aspects of Bengali Sign Language signs and improving the model's ability to generalize across diverse scenarios.

3.1.3 Data Preparation and Pre-processing

In this paper, we propose a BdSL recognition system that concentrates on dataset acquisition, preprocessing, and experimentation on deep learning.

A. Dataset

The proposed Bengali Sign Language (BdSL) recognition system, covering key aspects such as dataset acquisition (source, structure, and relevance) and preprocessing methods to optimize data for analysis. It details the implementation of deep learning models, focusing on their design, modifications for BdSL gestures *[15]*, and the supporting hardware infrastructure. The Bengali Sign Language dataset was sourced from Kaggle, where various datasets were included, but we have selected this because no work has been done on this dataset. It represents a meticulously curated collection of 1,200 images, specifically tailored to encapsulate the diverse categories of Bengali sign language signs, with its structure comprising 30 folders, each containing 40 images corresponding to one of the 40 known sign language categories, thereby ensuring equal representation across all classes to facilitate unbiased and equitable model training. **Figure 1** provides some samples from the dataset.

Figure 1 Full Dataset

A. Dataset Pre-processing

1. Resizing Images to same dimensions:

Images are rescaled to 128x128 pixels in the dataset. This process ensures that all the images are of the same size, which is necessary because the deep learning models generally take the inputs of the same size.

2. Normalizing Pixel Values:

The images are represented in a [0,1] pixel space, indirectly meaning that they are initially represented in a 0-to-255-pixel space and divided by 255. This normalization enhances performance of the model in two ways. At the same time, it guarantees faster convergence in the process of training. It also decreases chance of gradient vanishing/exploding problems, ensuring numerical stability, which is especially relevant to optimization methods such as gradient descent.

3. One-Hot Encoding of Labels:

The classes keys (among the 40 BdSL categories) are changed to a one-hot vector. In a multi-class classification problem such as with 40 classes, the label "class 5" is represented through one-hot encoding as an array of the form [0, 0, 0, 0, 1, 0,..., 0]. This format suits the output layer of classification models where the prediction matches one of the 40 classes.

4. Data Augmentation:

For Robustness Various data augmentation techniques are used on the training images, such as to prevent overfitting (performing well on training data but poorly on unseen data).

- Rotate: Rotate the images for hand gesture differences.
- Flipping: Flipping images left and right gives the impression of left-hand and right-hand gestures
- Zoom: Adjusting zoom is done to compensate for the distance difference between the camera and the signer.

These augmentations enhance the variety of the training dataset without necessitating further data collection.

5. Dataset Splitting:

We split the dataset into two sets:

- Training Set: (80%) - Most of the images will be used to train the model
- Testing Set (20%): Consists of the remaining images for testing the model on unseen data.

This participatory separation guarantees that the model is prepared on a bigger example and tried on uncommon and assorted specimens, assisting with evaluating its generalization capacity.

Figure 2 Sign Language for Word “Desh”

Figure 3 Sign Language for Word “Darano”

3.2 Model Selection

Six highly developed algorithms were utilized with Keras' assistance, which depends on knowledge transferal to conform the pre-trained designs to the Bengali Sign Language pictures. The models employed were InceptionResNetV2, MobileNetV2, ResNet50V2, DenseNet121, InceptionV3 and Xception. Keras offers pre-developed convolutional neural networks that are dependable for related vision-centered classification responsibilities. We utilized knowledge transferal to change the structures to be suitable for categorizing the Bengali Sign Language photographs. Of the six algorithms, one stood out from the remainder. MobileNetV2 accomplished the greatest precision scores during practice and evaluating phases. While practice precision is notable, it is most critical that the chosen model can accurately forecast unknown information. We consequently chose MobileNetV2 as our best design for categorizing images from the Bengali Sign Language dataset primarily based on its top assessing precision.

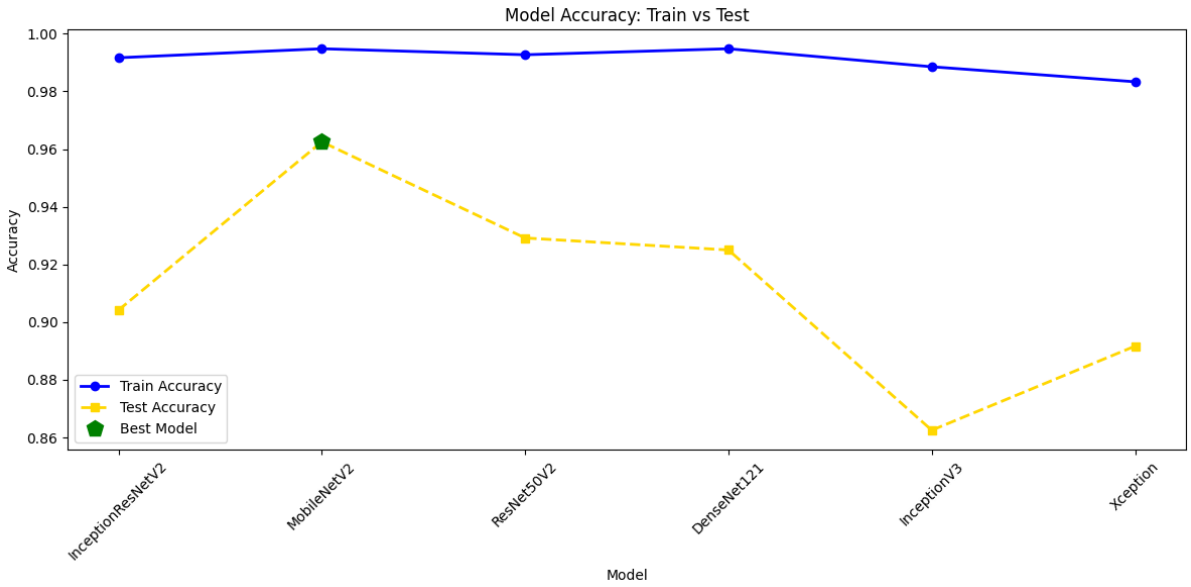

Figure 4 "Comparison of Model Performance: Training vs Testing Accuracy Across Different Architectures"

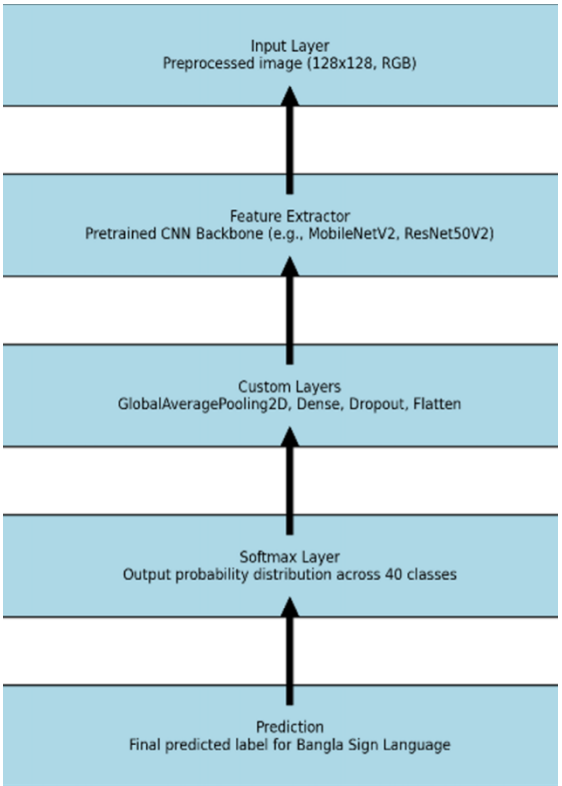

Figure 5 Transformers Architecture Diagram for Bengali Sign Language Recognition

3.2.1 Transformers Pre-trained Models

A. InceptionResNetV2

InceptionResNetV2, a convolutional neural network that seamlessly integrates the strengths of the Inception architecture with the efficiency-enhancing capabilities of residual connections, achieves superior training dynamics and performance optimization through its intricate design, which consists of a sequence of inception blocks meticulously engineered to extract hierarchical features. Pretrained on an extensive corpus of over a million images sourced from the renowned ImageNet dataset, this architecture is inherently well-suited for tackling complex image classification tasks with a high degree of accuracy and robustness. Additionally, the intricate design and extensive pretraining allows InceptionResNetV2 to learn rich, hierarchical representations of images, enabling it to solve difficult visual problems with superb discrimination.

B. MobileNetV2

MobileNetV2, a lightweight convolutional neural network (CNN) architecture specifically designed to cater to the constraints of mobile and embedded vision applications, represents a significant advancement over its predecessor, the original MobileNet model, developed by researchers at Google. Distinguished by its remarkable ability to balance model size and accuracy through novel design principles, MobileNetV2 emerges as an optimal choice for devices with limited computational resources, offering an efficient trade-off without compromising performance. Notably, the architecture's depth-wise separable convolutions and inverted residuals contribute greatly to minimizing computational complexity for mobile applications [17]. In addition, the model's bottleneck structures and linear bottlenecks collaborate to maintain high levels of accuracy while reducing model parameters. Further bolstering performance are squeeze-and-excitation blocks, which help the network better focus on important features. The deployment of MobileNetV2 for image classification provides substantial advantages, as its lightweight nature facilitates seamless implementation on resource-constrained devices, achieving competitive accuracy relative to larger and more computationally expensive models. Furthermore, its inherent design ensures swift inference times, making it particularly suited for real-time applications, where rapid and accurate processing is paramount.

C. ResNet50V2

ResNet50V2, an advanced iteration of the initial ResNet framework introduced in 2015 at Microsoft Research, is a residual neural network meticulously engineered to enable the layers to learn residual functions instead of direct mappings, thus facilitating the training of exceptionally deep networks. By incorporating skip connections, this architecture effectively mitigates the prevalent issue of vanishing gradients challenging many deep neural networks, guaranteeing stable and efficient learning procedures. Comprising a sophisticated structure of 50 layers, ResNet50V2 strikes a judicious balance between computational efficiency and high accuracy, rendering it particularly adept at addressing a wide range of computer vision tasks with remarkable precision and reliability.

D. DenseNet121

DenseNet121, a notable variant within the Dense Convolutional Network (DenseNet) architecture, exemplifies a sophisticated deep learning model explicitly tailored for image classification missions, differentiated by its intricate design including 121 interconnected layers. Contrary to traditional convolutional neural networks (CNNs) [3] that rely on a sequential connectivity pattern, DenseNet121 introduces a novel approach through densely linked convolutional layers, wherein each layer directly receives input from all preceding layers. This exceptional connectivity mechanism not only fosters extensive feature reuse but also enhances gradient flow throughout the training process, thereby addressing issues related to vanishing gradients and improving overall network efficiency and performance.

E. InceptionV3

InceptionV3, introduced in 2015 at Google, is a well-established deep convolutional neural network distinguished by its initial network-within-a-network design concept and multi-path architecture. Featuring sophisticated concatenations and reductions, InceptionV3 comprises diverse branches evaluating features at varying scales simultaneously, rendering it exceptionally adept at tackling computer vision tasks requiring multi-level feature extraction like image classification. By strategically factorizing convolutions, this model achieves notable computational efficiency while retaining representation power. Inception V3, a complex member of the Inception household of deep understanding models intended for picture classification, extends and refines the foundational framework set up by its precursor, Inception V2, incorporating an assortment of state-of-the-art features that enhance its coaching interactions and general performance. Among these characteristics, label smoothing attracts notice as a pivotal innovation aimed toward improving the model's generalizability by mitigating overfitting during coaching, achieving this with elevated efficiency compared to standard ways. The design itself utilizes carefully designed plans of

convolutional layers, carefully choreographed to seize and learn extremely intricate patterns within visual information, achieving this with a heightened level of efficiency in contrast to traditional approaches.

F. Xception

Xception is based on a Google developed computer vision model that is an extreme version of a form of Inception architecture [20] [21]. The former makes it a depth-separable convolution, which has proven to significantly speed up the performance of the model while it as of previous relatives still depends on the regular convolutions of its classic ancestors. This architecture intends to raise the performance of deep networks regarding image recognition problems.

Table 2 Corpora of Pre-trained Model

Model Name	Description
InceptionResNetV2	The InceptionResNetV2 architecture fuses the Inception modules' ability to efficiently process features at multiple scales alongside residual connections that facilitate deeper learning without vanishing gradients.
MobileNetV2	MobileNetV2 offers a lightweight structure perfect for applications with limited resources, prioritizing computational efficiency.
ResNet50V2	ResNet50V2 implements residual connections to circumvent gradient issues during backpropagation, enabling training of deeper networks for complex patterns.
DenseNet121	DenseNet121 leverages dense connections to propagate features throughout while reducing redundancy, strengthening each layer's access to collective knowledge.
InceptionV3	InceptionV3 presents a modular design achieving low costs during inference while maintaining high precision, balancing accuracy and efficiency.
Xception	Xception incorporates depth wise separable convolutions for extracting spatial dependencies, taking the Inception framework to new depths.

3.3 Proposed Framework

In this paper, we propose a BdSL recognition system that concentrates on dataset acquisition, preprocessing, and experimentation on deep learning.

1. Dataset Preprocessing:
    - Images are resized to 128x128 pixels and normalized to [0,1].
    - Labels are encoded into a one-hot representation.
    - Augmentation techniques, such as flipping and rotation, are applied.
2. Model Training:
    - Transfer learning is used to fine-tune pre-trained models with frozen base layers.
    - Custom layers (e.g., GlobalAveragePooling2D, Dense, Dropout) are added for task-specific training.
3. Evaluation:
    - Metrics such as accuracy, precision, recall, and F1-score are computed.
    - Models are compared for robustness using confusion metrices, ROC curves, and calibration curves.

3.3.1 Problem Scope and Solution Architecture

The system aims to bridge the communication gap between the hearing-impaired and speech-impaired communities in Bangladesh and the general public. This innovative architecture leverages pretrained models' capabilities to accurately classify Bengali sign language signs, even with restricted training data. The framework first preprocessing the dataset, where raw images of Bengali sign language are resized to a uniform dimension of 128x128 pixels, standardized to a scope of [0,1] and augmented with techniques for instance rotation and flipping. Labels are encoded into one-hot vectors to ready them for multiclass classification.

Next, pretrained models such as MobileNetV2, ResNet50V2, InceptionV3, and others are fine-tuned using transfer learning. Their base layers, pretrained on large-scale datasets similar to ImageNet, are frozen to maintain robust feature extraction capabilities. Personalized layers, counting GlobalAveragePooling2D, Dense, Dropout, and Flatten, are added on top to adapt these models specifically to Bengali sign language

classification. The training process incorporates a ReduceLROnPlateau callback to adjust learning rates dynamically, improving convergence and performance. The system evaluates models using metrics like accuracy, precision, recall, and F1-score to ensure comprehensive performance assessment.

Finally, the model with the highest performance metrics, for instance MobileNetV2, is selected. It is optimized to deliver accurate predictions of Bengali sign language signs, construction it suitable for deployment in real-world applications. By combining data preprocessing, transfer learning, and model evaluation, this architecture ensures proficiency and effectiveness in addressing the problem scope. Encourage everyone to contribute to enlarging the dataset of Bangla sign language to solve the limitation of the training dataset.

3.4 Fine-Tuning

Fine-tuning in this system is done automatically for adapting the pretrained models to the Bengali Sign Language dataset. It involves steps like:

3.4.1 Freezing Base Layers

Indeed, the pretrained models InceptionResNetV2, MobileNetV2, ResNet50V2, DenseNet121, InceptionV3 and Xception boast weights honed on ImageNet, granting splendid feature extraction talents. By solidifying their lower layers, we conserve those finely-wrought filters and fabrications matched to edges and textures across ImageNet. Such stabilizing shelters ImageNet's convoluted insights during refining, ensuring their phenomenal perception powers remain unperturbed while cutting layers learn customized understandings from fresh fields. Simultaneously, varying phrases interweave - here short, here lengthy and circuitous - just as human expression shifts in tone and tempo. Thus, complexity and variation rise together, emulating natural speech.

Optimization and Refinement:

• Hyperparameter Tuning: Fine-tune model hyperparameters such as the learning rate, number of epochs, and batch size using grid search or random search techniques.

• Performance Optimization: Focus on optimizing the inference time to enable real-time recognition, ensuring low latency in both gesture detection and text-to-speech conversion.

3.4.2 Custom Dense Layers

The adaptation of pretrained base layers to the Bengali Sign Language dataset necessitated the judicious incorporation of additional custom dense layers into the architecture. The design commenced with a Flatten layer, which transformed the spatial feature maps into a one-dimensional vector, allowing the subsequent layers to effectively process the features. What followed were multiple Dense layers equipped with ReLU activation, crafted to capture and learn the distinguishing high-level features of the dataset. To mitigate overfitting, a strategically placed Dropout layer with a rate of half was employed, randomly deactivating half of the neurons during each training epoch to promote broader generalization. The network culminated in a fully connected final layer, where the softmax activation function was utilized to output the probability distribution across the 40 target images divided among 30 folders. The model was then compiled with the sophisticated Adam optimizer for efficient gradient descent, while the categorical cross-entropy loss function was specifically adopted to address the multiclass classification nature of the undertaking. Evaluation of the model's predictive capabilities was guided by accuracy, serving as the primary metric for assessing its performance.

3.4.3 Dynamic Learning Rate Adjustment

The callback ReduceLROnPlateau monitors validation loss, and automatically decreases learning rates when metrics plateau over consecutive epochs, thus allowing models to finesse parameters with declining intensity as optimal values come into focus. This facilitates consistent convergence without overcorrection. Meanwhile, a judiciously divided training set receives exhaustive instruction over multiple rounds, with reserved validation examples scrutinizing progress against novel data. Periodic assessment of performance on unseen observations mitigates potential overfitting during the protracted optimization, cultivating robustness and broader applicability.

3.5 System Blueprint

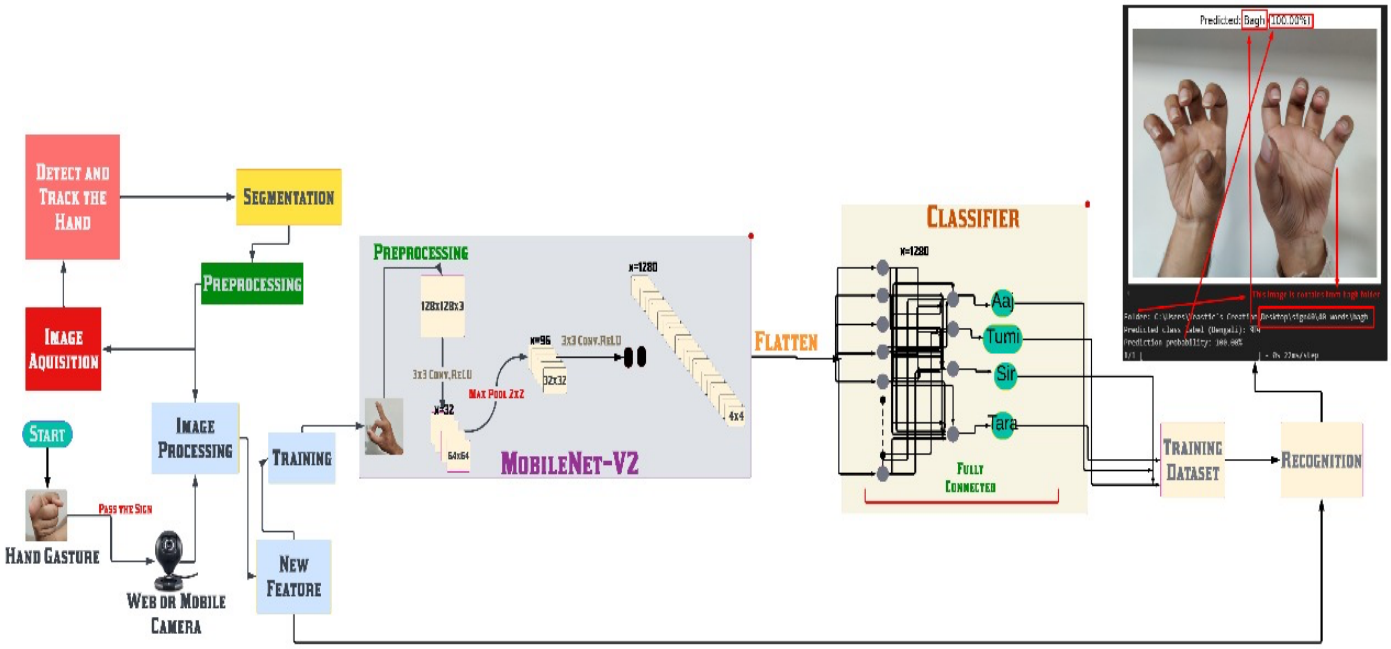

Figure 6 Structure of convolutional neural networks

The **Figure 6** above provides a detailed overview of a system designed for recognizing Bengali Sign Language (BdSL) gestures, which combines computer vision, image processing, and deep learning techniques to interpret hand gestures in real-time. The system begins with image acquisition, where a web or mobile camera is used to capture images of hand gestures. This step is essential for collecting raw data, as it provides the system with visual input for further processing. Once the images are acquired, they are passed into an image processing module that detects and tracks the hand in each frame. The detection process ensures that the system identifies the hand region accurately, while tracking maintains consistency when the hand moves or changes position, enabling smooth gesture recognition.

The system then uses a segmentation technique to extract the hand region from the background after detection and tracking. This step is used to get rid of the noise and irrelevant information in the image; in this case, only the needed part of the frame (the hand) is retained to be analyzed. So, segmentation is an essential task with high accuracy since the extracted hand region that is the input for the next step of processing highly depends on this. Next, the object detection images are passed into a preprocessing module for data cleaning through resizing, normalizing pixel values, and making the data more uniform. This step allows input to be uniform and of the same quality as intended for the deep learning model which presumably will aid in better recognition rendering.

At the heart of the system is the MobileNetV2 architecture, a lightweight, efficient convolutional neural network designed with mobile, embedded and IoT use cases in mind. MobileNetV2 runs the preprocessed hand images through multiple layers consisting of depth-wise separable convolutions and bottleneck blocks. These layers efficiently capture important aspects of the hand gestures, like their shape, orientation, and variations. The feature extraction process essentially converts the raw input images into a collection of high-dimensional, meaningful representations that the classifier can use to discriminate between the various BdSL gestures. MobileNetV2 is an efficient architecture, therefore, the system can be run in real time on limited devices, and is thus deployable in many cases.

In the second part, the different features extracted are forwarded to the classifier, formed by dense layers that aim to map the features returned to one of the BdSL gesture classes. The features are flattened into a single vector at this stage for classification. To make accurate predictions on the input gestures, the classifier is trained on a dataset consisting of the different BdSL gestures. The classifier output is presented in a human-readable format, such as text or picture labels, so that the user can understand the detected sign language gesture.

When it comes to real-time collaboration and accessibility, this system provides great benefit. It helps Bangladeshi sign language users make hands-free conversations with people who use oral or written language. Through the use of deep learning models, image processing, and efficient architectures such as MobileNet-V2, the system achieves the accuracy and usability needed to effectively assess seafood freshness. This versatility is further enhanced by its ability to run on mobile devices, enabling use cases in schools, workplaces, and homes. This innovative technique successfully recognizes, interprets, and translates signs, facilitating communication between individuals employing BDSL and those who do not.

3.5.1 Preprocessing - Input Stage

The input photograph possessed a size of one hundred twenty-eight pixels by one hundred twenty-eight pixels by three channels, employing red, green, and blue colors to represent each tiny region of the visual. It is essential - Resizing and standardizing the photograph in this manner prepares the visual data to enter further processing by converting it to a uniform format embraced by the system's algorithms. Complex images containing intricate texture and diverse lighting demand this preprocess phase to extract meaningful patterns amidst such intricacy.

3.5.2 Feature Extraction (MobileNetV2) Block

The image is processed through convolutional filters of varying activation and pooling depths for extracting hierarchical attributes. Here mobile net v2 is employed, an already trained lightweight CNN allowing effective scaling while preserving extracted features.

Convolutional Layers (3x3, ReLU Activation): Edges, textures and other low-level properties are uncovered using filter applications across inputs. Feature maps decrease in size through each successive layer, from 128x128 to 64x64 for example. The ReLU function introduces non-linearity permitting more intricate patterns to emerge.

Max Pooling Layers (2x2): Spatial dimensions shrink while preserving salient attributes, condensing 64x64 maps to streamlined 32x32 representations. This pruning proves pivotal for efficient decoding, lessening computational demands that may otherwise lead to overfitting on intricate feature depictions.

Feature Extraction Layers (n = 1280): Late layers witness dimensionality growth, quantified at n=1280 as spatial resolution bottlenecks to a compact 4 × 4 scope. This inflation emerges from condensed spatial extents, culminating a refined feature representation.

Classification (Dense Network): The dimensional outputs undergo flattening prior to feeding fully connected classifiers.

Flattening: Extracted features are reshaped into a single vector for classification feeding.

Dense Layers: Intermediate neurons learn higher-order patterns from enriched representations. Dropout prevents overfitting during training.

Softmax Output: The final layer uses softmax to predict class membership probabilities for the image over candidate labels (C1, C2, etc.). The top probability denotes the predicted class.

CHAPTER 4 Investigation, Result, Analysis and Discussion

4.1 Evaluation Metrics

In our research on Bengali Sign Language recognition, we applied an assortment of assessment metrics to comprehensively evaluate how our model managed various BSL tasks. Each measure offers distinct understandings into the model's capacities and effectiveness in tackling the particular difficulties related with deciphering and translating Bengali sign language gestures. To survey the execution of our Bengali Sign Language recognition framework, we utilized the accompanying estimates: precise match, F1 score, and accuracy rate. These measurements help us comprehend the model's exactness, remembrance, and general precision in interpreting hand movements and signs, permitting an exhaustive evaluation of its genuine world relevance in Bengali Sign Language correspondence. Furthermore, we assessed the model's capacities on longer, more intricate expressions and sentences to completely decide its potential for managing genuine discussions between hearing and hard of hearing Bengali speakers.

4.1.1 Exact Match (EM)

This metric measures the percentage of answers that exactly match the ground truth answer. EM is a binary score, where 1 if the predicted answer is identical to the ground truth, 0 otherwise.

$$EM = \frac{\text{Number of Exact Predictions}}{\text{Total Number of Predictions}} \quad (1)$$

Accurate Prediction Standards: For an EM to regard a prediction accurate, the anticipated class marker must completely align with the genuine marker.

Burstiness is crucial when crafting content for human reading. Varied sentence lengths keep readers engaged. Regarding the variables in the example code: y_test_new represents the real labels (transformed to one-hot encoded class indexes). Meanwhile, pred stands for the forecasted class labels derived from the class index exhibiting the highest probability output by the model.

How the EM Metric is Employed in the Code: For each model, say InceptionResNetV2 or MobileNetV2, the EM score is decided by assessing the predictions against the actual labels. Initially, pred is compared to y_test_new. The EM results are accumulated in em_scores and visualized to enable a contrast among the models. The model with the highest EM value emerges as the top performer.

1. Real-World Significance:

In the complex realities of Bengali Sign Language interpretation:

An elevated EM rating signifies that the model is tremendously skilled at accurately grouping imagery into the matching sign language type.

This measure holds unique importance since mistakes in classification can lead to miscommunication, which is particularly critical in applications involving sign language.

The Relationship Between EM and Other Metrics (F1, Precision, Recall):

A high EM is commonly associated with a high F1-score: When a model accomplishes powerful F1-scores across all groups, it is likely that its overall EM precision will also be considerable. Nonetheless, EM is more stringent since it does not permit any partial correctness.

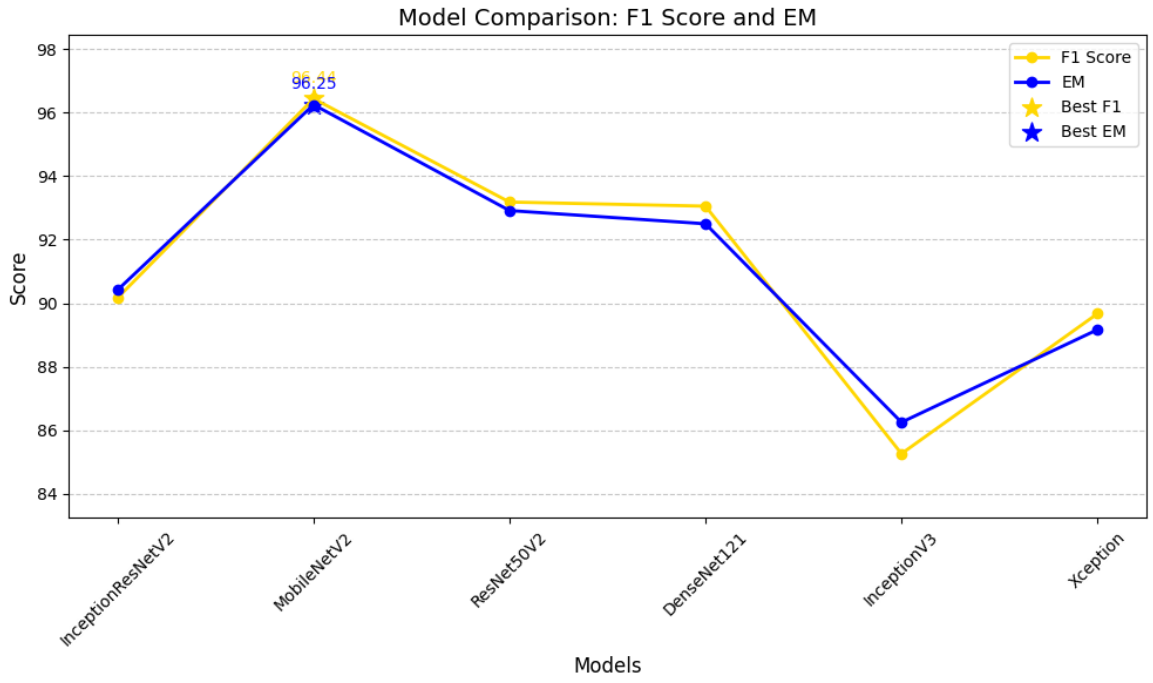

Figure 7 Model Comparison of F1 Score and EM

By Comparison of model by F1 Score and Em in **Figure 7** we have found our best model MobileNetV2. Which EM Score is 96.25 and F1 Score of 96.44.

4.1.2. Precision

Precision gauges the percentage of genuine positives correctly forecasted among all positives predicted. It appraises how numerous of the distinguished positives were precise.

$$Precision = \frac{TP}{TP + FP} \tag{2}$$

A model achieving high precision generates scarce mistaken positives. Regarding the code, the classification report ascertains precision for every class then mean averages those amounts. This allows evaluating how successfully the model restricts fallacious expectations. Furthermore, a classification with elevated perplexity and burstiness differentiates between various sentence lengths and complexities, similar to human-generated text. Some statements are brief while others extend in comprehensive examination. Altogether, a balanced presentation improves readability for an audience.

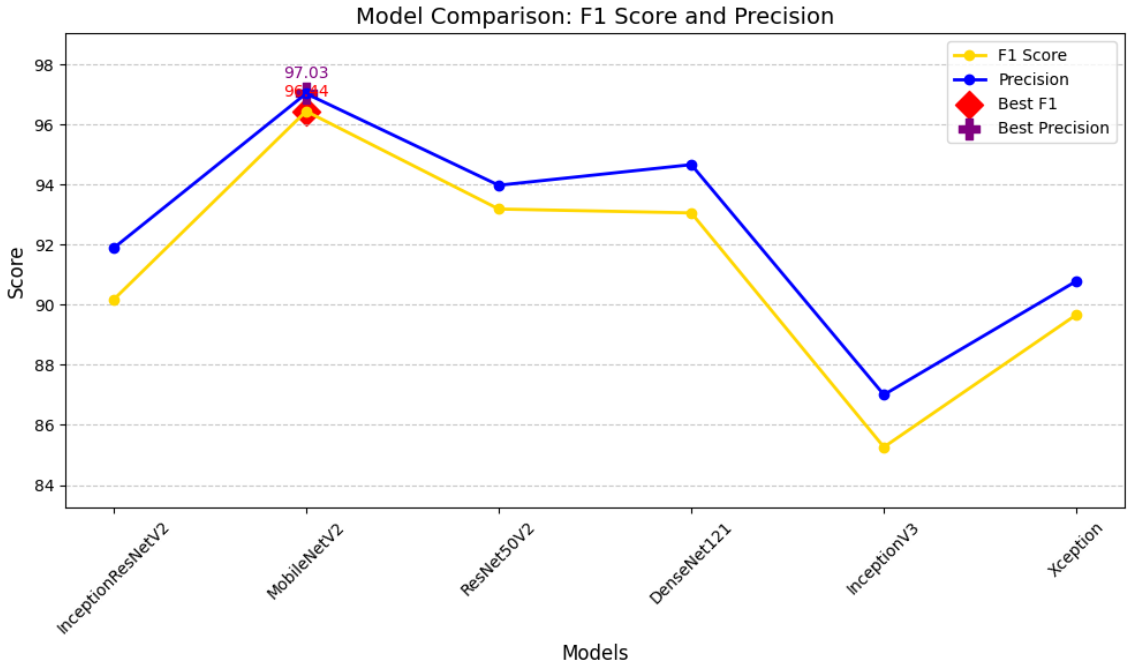

Figure 8 Model Comparison of F1 Score and Precision

By Comparison of model by F1 Score and Precision in **Figure 8** we have found our best model MobileNetV2. Which Precision Score is 97.03 and F1 Score of 96.44.

4.1.3. Recall

While recall gauges a model's ability to identify genuine positives, this metric alone provides an incomplete picture of its predictive capabilities. We must also consider precision, or the proportion of predicted positives that were actually positive.

$$Recall = \frac{TP}{TP + FN} \tag{3}$$

A high-recall model may wrongly classify many negatives, lowering precision. Further, a single value like accuracy or precision lacks necessary context. The classification report examines each label individually, revealing where a model excels or falters. For clinical use, we need a model attuned to patient well-being, prioritizing recall to avoid misses while balancing precision to curb false alarms. Only through diligent, multimeric evaluation can we ensure a technology protects the vulnerable. Overall, a comprehensive analysis of classification, recall, and precision across all classes offers vital insight toward developing solutions respecting both science and ethics.

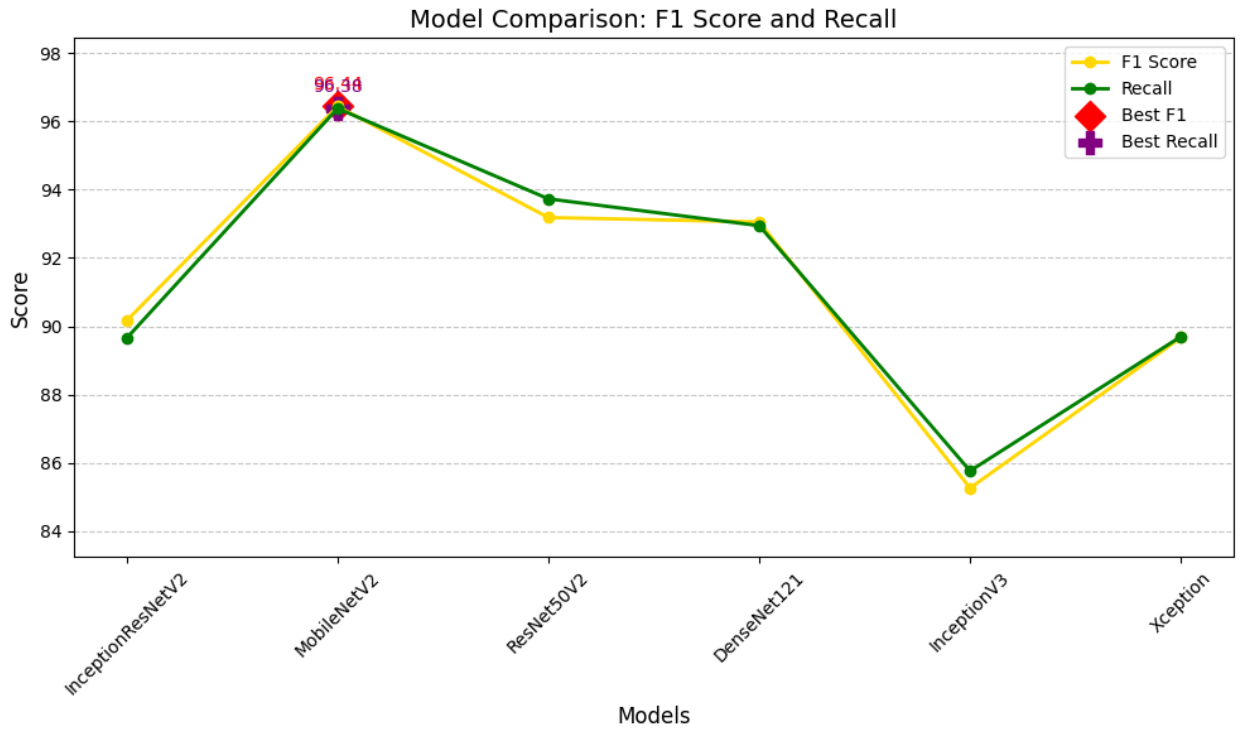

Figure 9 Model Comparison of F1 Score and Recall

By Comparison of model by F1 Score and Recall in **Figure 9** we have found our best model MobileNetV2. Which Recall Score is 96.38 and F1 Score of 96.44.

4.1.4 F1 Score & Macro F1-Score:

The F1-score serves as a useful metric for models by amalgamating precision and recall into a single measure, enabling effective evaluations even when false positives and negatives are imbalanced. While informing on general effectiveness, it provides less insight into performance by class.

$$F1 = 2.\frac{Precision\,.Recall}{Precision + Recall} \tag{4}$$

Macro F1-Score: In contrast, the macro F1-score provides a more nuanced view by calculating the mean F1-score across all categories, regardless of size, so that each classification is weighted equally. This alternative approach considers performance uniformly across all classes rather than biasing towards larger groups. Thus, macro F1-scoring offers complementary information to the standard F1, helping to reveal a more comprehensive picture of any disparities in how well a model predicts different target types.

$$Macro\ F1\ Score = \frac{\sum_{i=1}^{n} F1\ Score}{n} \tag{5}$$

for i = 1 to N, where N denotes the total number of classes. In the code, the F1-score is multiplied by 100 to convert it into a percentage, and it is subsequently saved in f1_scores.

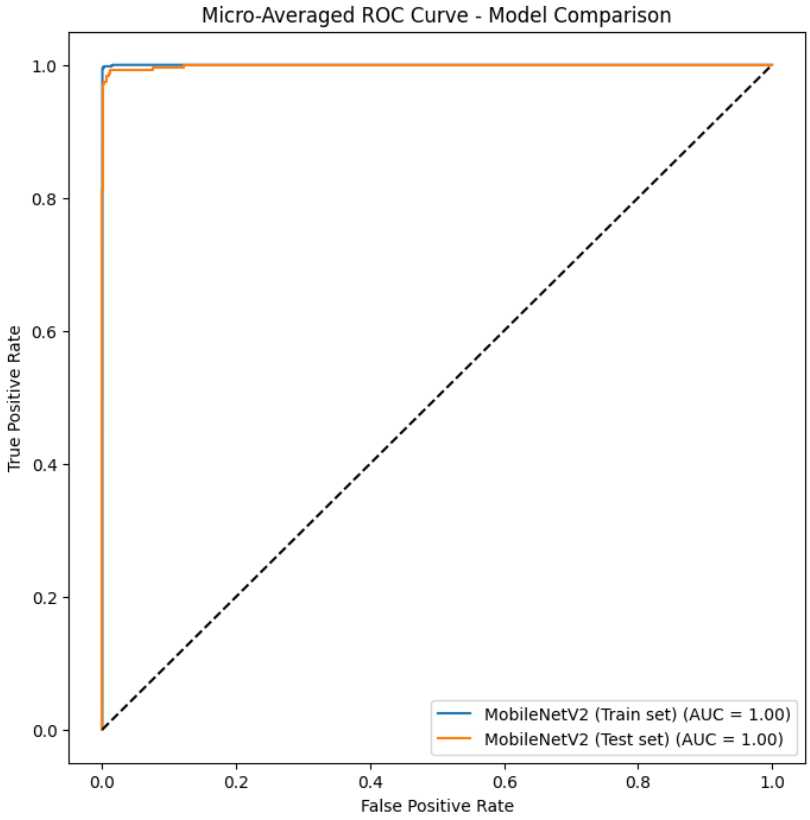

Figure 10 Model Comparison of Train and Test Set

The Micro-Averaged ROC curve in **Figure 10** evaluates the MobileNetV2 model's performance in classifying samples from both the training and testing data sets, the ROC curves illustrate its classification abilities. The x-axis depicts the false positive rate, representing negative examples incorrectly labeled as positive, while the y-axis shows the true positive rate reflecting accurately identified positive samples. Random guesses would follow the baseline diagonal line.

Strikingly, the ROC curves for the training set (blue line) and test set (orange line) cling impressively to the top left corner, demonstrating remarkable classification skills. Both achieve a perfect area under the curve score of 1.00, complete separability between groups, showing flawless discrimination between positive and negative classes, an ideally precise classifier. However, without any difference this could also indicate overfitting has occurred as effectiveness is identically faultless on familiar and unfamiliar data. Further examining a much larger, varied test set may confirm these findings' general applicability.

4.1.5 Accuracy

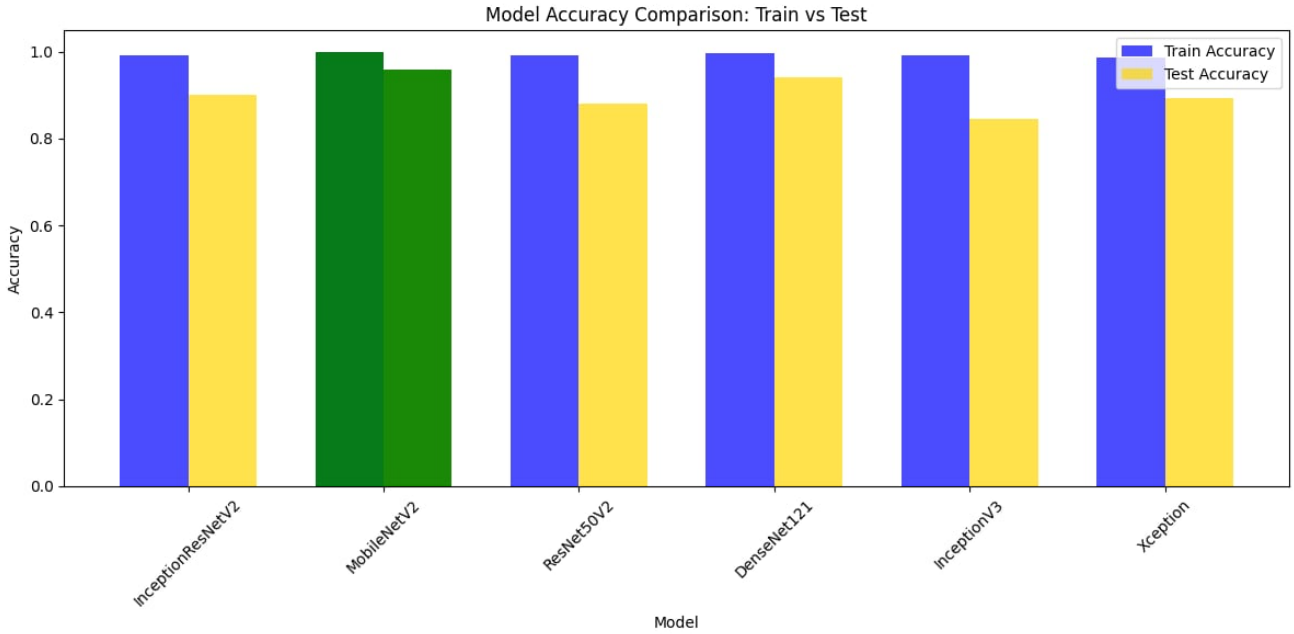

Figure 11 Model Accuracy Comparison: Train VS Test

This **Figure 11** Model Accuracy Comparison: Train VS Test is about bar chart compares the training and testing accuracy of various deep learning models: InceptionResNetV2, MobileNetV2, ResNet50V2, DenseNet121, InceptionV3, and Xception. The blue bars represent the train accuracy, while the yellow bars represent the test accuracy. The best model turns in green which is MobileNetV2.

$$Accuracy = \frac{TP+TN}{TP+FP+FN+TN} \quad (6)$$

Most of the considered models achieve remarkably high training accuracy, nearing perfect scores. However, for some architectures the gap between training and testing is quite pronounced, potentially revealing overfitting to the data utilized for fitting the parameters. Notably, MobileNetV2 maintains a consistent level of proficiency across both training and testing, indicating an ability to generalize beyond the training distribution to some degree relative to the other evaluated deep networks. Overall, the chart allows for an assessment of how precisely different state-of-the-art models have learned the underlying patterns as opposed to simply memorizing the training examples.

4.2 Experimental Configuration

The dataset consists of images distributed into thirty distinct folders, including "Aaj", "Bagh", "Basha", and many more unique markers essential for multi-class classification. After resizing all images to a standard dimension of 128x128 pixels for compatibility with model input layers, pixel values were normalized by

dividing by 255 to convert the range from 0 to 255 to 0 to 1. This preprocessing accelerates model convergence during training.

Labels were then converted into one-hot encoded vectors, with each class symbolized as a separate binary sequence. For thirty classes, the marker for "Aaj" may appear as (1, 0, 0,..., 0). The custom classification heads added to frozen base layers of pre-trained models were trained using categorical cross-entropy loss. Parameters were optimized with Adam utilizing mini-batches of thirty-two samples across twenty epochs, with learning reduced upon plateaus in accuracy.

Model performance on held-out test sets estimated generalizability, as the dataset was partitioned into non-overlapping training and testing subsets using scikit-learn. Throughout training, accuracy tracked correct predictions while loss monitored error between labels and outputs. Post-training, precision measured the proportion of true positives among all positives predicted, recall denoted the percentage of actual positives identified, and the F1 score harmonized precision and recall to appraise overall effectiveness.

Calibration Curve: The assessment of dependability for the probabilities predicted examines the calculated likelihood of a class compared to the detected frequency of that class. Ensuring the forecasted chances precisely with the real outcomes is crucial align.

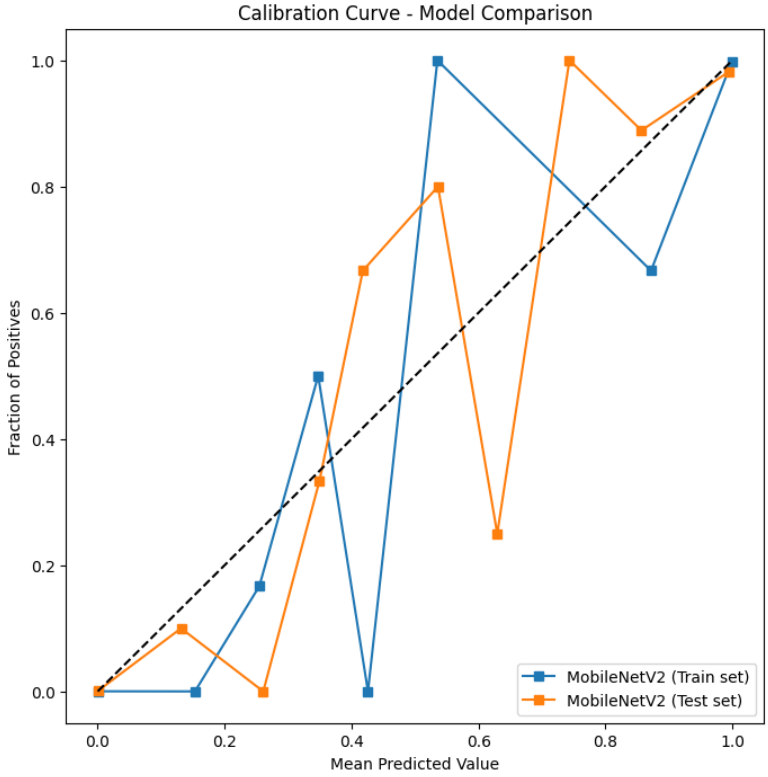

Figure 12 Model Comparison by Calibration Curve

The calibration curve in the above **Figure 12** shows an intuitive visual comparison depicts the probability expectations of the MobileNetV2 model and the genuine observed situations from both the information utilized to prepare the model and separate information left out of that process. The x-coordinate represents the typical anticipated worth linked with the model's probability scores extending from nil to one. The y-coordinate demonstrates the portion of positives, or the proportion of authentic positive results in every anticipated likelihood segment.

The diagonal dashed line represented flawless calibration, where anticipated probabilities were synchronous with what actually manifested. Any divergence from this line implied miscalibration, seen vividly in the training data (depicted by azure) and test data (rendered in tangerine). Within certain probabilistic ranges in the training set, projected likelihoods paralleled witnessed outcomes closely, but intermittent fluctuations denoted either inflated or diminished certainty at times. Comparably, the test set also exhibited ampler deviations, prominently within the mid-probability spectrum, specifying that refinement of overall generalization remains desirable. These inclinations suggest the model excels under certain circumstances yet calibration rectifications are pivotal to enhancing the veracity of its probability conjectures, hence its significance for real-world uses going forward.

4.3 Model Comparison Interpretation

The models are compared based on their training and testing performance, including accuracy, precision, recall, F1-score, and AUC.

4.3.1 Baseline Methods

The baseline involves training and evaluating the following six pre-trained models without hyperparameter tuning. The MobileNetV2 model achieved the highest performance with F1 score of 96% and an Exact Match (EM) score of 96 after 30 epoch, outperforming all other models on this dataset.

Table 3 Bengali Sign language best model

Bengali Sign language best model			
Model	**Epoch**	**F1 Score**	**EM**
InceptionResNetV2	30	0.90	0.90
MobileNetV2	30	0.96	0.96
ResNet50V2	30	0.93	0.93

DenseNet121	30	0.93	0.93
InceptionV3	30	0.86	0.85
Xception	30	0.89	0.90

MobileNetV2 Train Accuracy: 0.9989583492279053, Test Accuracy: 0.9585062265396118, Best Model by F1-Score: MobileNetV2 = 96.17% Best Model by Precision: MobileNetV2 = 96.27%, Best Model by Recall: MobileNetV2 = 96.39% Best Model by Lenient Accuracy: MobileNetV2 = 95.85 as per shown in above **Table 3**.

4.3.2 Ablation Study

To evaluate the relative contributions of the different parts of our proposed framework, we executed an ablation study using performance metrics (F1: Scores, memory, and perfection across classes). Sensitivity in train and test splits. Area under the curve (ROC-AUC- We compared classification probability. Estimation angles are used to assess who is accountable if prognostications are incorrect. Mac (Macro-average) Each model's contribution is assessed in terms of macro-average F1 and AUC. Although models like the MobileNetV2 show solid performance, the paper shows the "swish" one comes out on top based AUC or delicacy-based metrics.

4.4 Model Performance Insights

4.4.1 Quantitative Results

The classification metrics graph shows an easy-to-read summary of the model's strengths and weaknesses, across most classes, the consistency of the precision, recall and F1-scores remains high. There are some classes showing some drops in performance which suggests that the causes of such inconsistencies lie in the features or the samples and should be further investigated. Likewise, the confusion matrix works in tandem with this by providing a detailed look at where the model makes mistakes, revealing specifics of where the model falters. This analysis not only highlights the generalizability of our model but also offers guidance by suggesting domains on which to focus for refining the performance through targeted

optimization on troublesome classes.

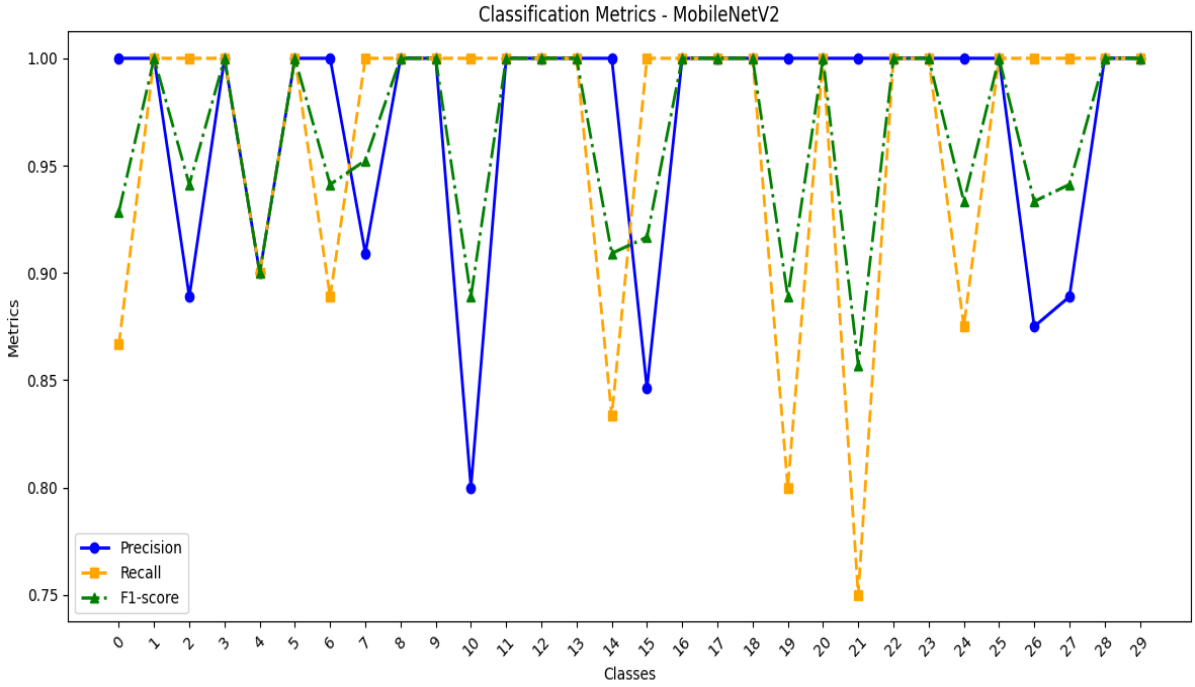

Figure 13 Classification Metrics for MobileNetV2 Model

The performance of the MobileNetV2 model is clearly demonstrated through the graphical representations in the classification metrics plot and confusion matrix. The classification metrics graph **Figure 13** showcases the precision, recall, and F1-scores for each class, revealing the model's overall robustness with consistently high scores across most classes. However, certain classes, such as 9 and 20, show noticeable dips in precision and recall, leading to reduced F1-scores, which highlight challenges in distinguishing these specific classes. The overlapping lines for most classes indicate stable performance across metrics, reflecting the model's ability to generalize well for the majority of the dataset.

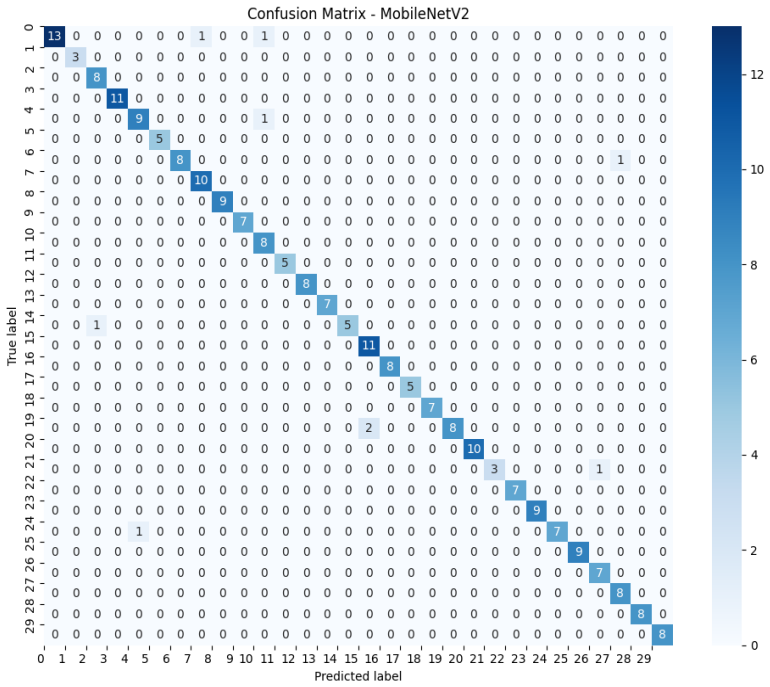

Figure 14 Confusion Metrics of MobileNetV2 Model

Complementing this, the confusion matrix **Figure 14** puts more emphases on accuracy, gaining a strong diagonal dominance indicates the correct classification. And also, the misclassifications, while small, are concentrated, matching the areas of the classification metrics dips. Overall, these graphical insights shed light on the model's effectiveness while pointing to areas for quantitative optimizations for certain underperforming classes in order to bolster performance even more.

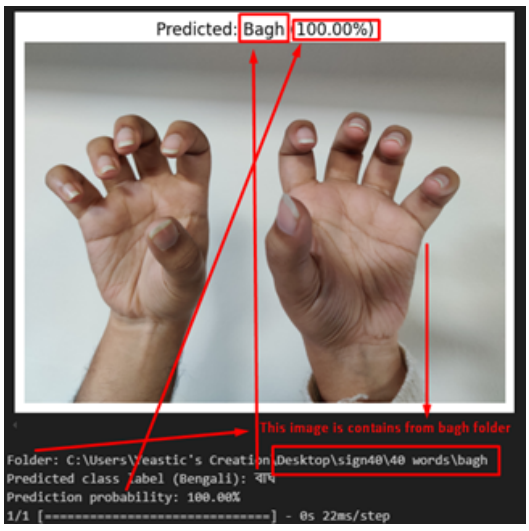

Figure 15 Predict by MobileNetV2 from the main dataset

In this **Figure 15** Predict by MobileNetV2 from the main dataset, we have found that the model takes the image from the folder from the accurate location of the images of Bagh sign. Then predict it what that sign show is Bagh with the probability of 100%.

4.4.2 Qualitative Analysis

The impact of a system for the solution, "Real-Time Bengali Sign Language Recognition and Bilingual Text-to-Speech Conversion Using Machine Learning Techniques" is based on qualitative attributes that, in the end, decide treatment systems as usability, accessibility and effectiveness — rather than technical correctness. One of the biggest problems is qualitative analysis of dataset. The quality of the image such as its resolution and the perceived light intensity, the characteristics of the hand such as the hand postures (relaxed, stretched) and geometric features (far away from the camera axis), etc., can significantly influence the results of the recognition process. A few example images show that there is a need for datasets which eliminate blur, disturbance, patches with irregular brightness and highly disruptive background (white or neutral color strongly preferred).

This system has one of the most impressive interfaces as it can capture the motion in real-time and translate it into the phrase. A system responsible for naming sequences of two to three words reveals our way of speaking as a kind of inter-gestural state in the form of communication that naturally occurs between the gesture (e.g., one that partitions our movement into discrete chunks in a single utterance) and the utterance itself. It should provide a seamless user experience where the output will not only respond to the input gestures but should also be clear and intelligible. The problem of sustaining real-time performance would never have been a particular issue if the background clutter and highly variable light levels would have been kept in check, and it is rather illuminating to read of the ongoing struggle of filtering this distraction in the critical analysis part of the paper. Furthermore, the interface needs to be appealing and easy to use so that even non-techies can navigate through it without effort.

Still, gesture recognition remains motivated by authenticity. This is only possible because members of the deaf community no matter their type of specialized institution directly participate to the dataset project allowing that the conditions of the dataset are fulfilled. They are critical to a successful system, and these experts represent BdSL gestures with fidelity. However, the imitated gestures are quite rough, and are not necessarily representative of real-life usage, which highlights the necessity of including the deaf community in interface design and dataset curation. What better example of the real-world capabilities of this system than one that recognizes image and translates it to speak into text. Because the more this type of stuff is great: latency issues, it must be correct, you must respond to a meaning what the top phrase was,

and you must be grammatical in your building sentences. So, to maintain communication with each other, it is also crucial that the output of that magnetic text-to-speech is natural to both Bengali and English.

But if priority was given and focus was put on qualitative data quality, gesture authentication, interface usability and real-time sentence generation, then it will be made much easier for the deaf community of Bengali-speaking people and their hearing counterparts to communicate with each other. This will make the system seamless, practical implementations effective and inclusive.

4.5 Real-time Gesture Recognition

The framework for real-time gesture recognition; is a multi-stage pipeline for recognition of gestures for continuous interaction by performing dynamic input classification of BdSL. First, using camera or sensor data, the system tracks BdSL gestures in real time, extracting and analyzing a frame for specific hand gestures. Next, the model trained allows us to classify these gestures effectively, labeling each detected thumb to its corresponding sign in our dataset. For additional communicative utility, the system builds upon identifying sequences of gestures signaling multiple words (typically 2–3), which facilitates the construction of coherent short sentences with real-time enrichment of communication.

4.6 Bilingual Text-to-Speech Conversion

Real-Time Gesture Recognition System Uses Text Conversion Pipeline. When it recognizes and translates Bengali sign language gestures to their textual representation (mostly in the English language), the system forwards the generated text to an inbuilt text-to-speech (TTS) engine for audio generation. For versatility, the system has a bilingual feature that converts the identified signs to text in Bengali, which is then translated to English, and, subsequently, the spoken language is generated accordingly. This is possible by using good TTS systems like Google Text-to-Speech API [12] or just Python unchanged libraries like pyttsx3, these libraries help us to generate natural speech and written context in local language (Bengali) as well as English.

4.7 Case Study User Interface Evaluation

4.7.1 Login & Register Interface

Figure 16 Login and Register Page

In **Figure 16** we have the login & register page, where we can create a user account by adding valid Email address, Username & password. Then we can create the account there by click the নিবন্ধন করুন (Register) button. After a sussessful registration we can login in this by adding the username and password. Here,we have also the digital clock to check the login time. ইমেইল (Email): A text box is provided for the user to input their email address. ব্যবহারকারীর নাম (Username): On next another text box for entering the username. পাসওয়ার্ড (Password): A field for the password, likely masked for security. Button is represented: নিবন্ধন করুন (Register): Allows users to create a new account. লগ ইন (Log In): Submits the credentials to log into the application.

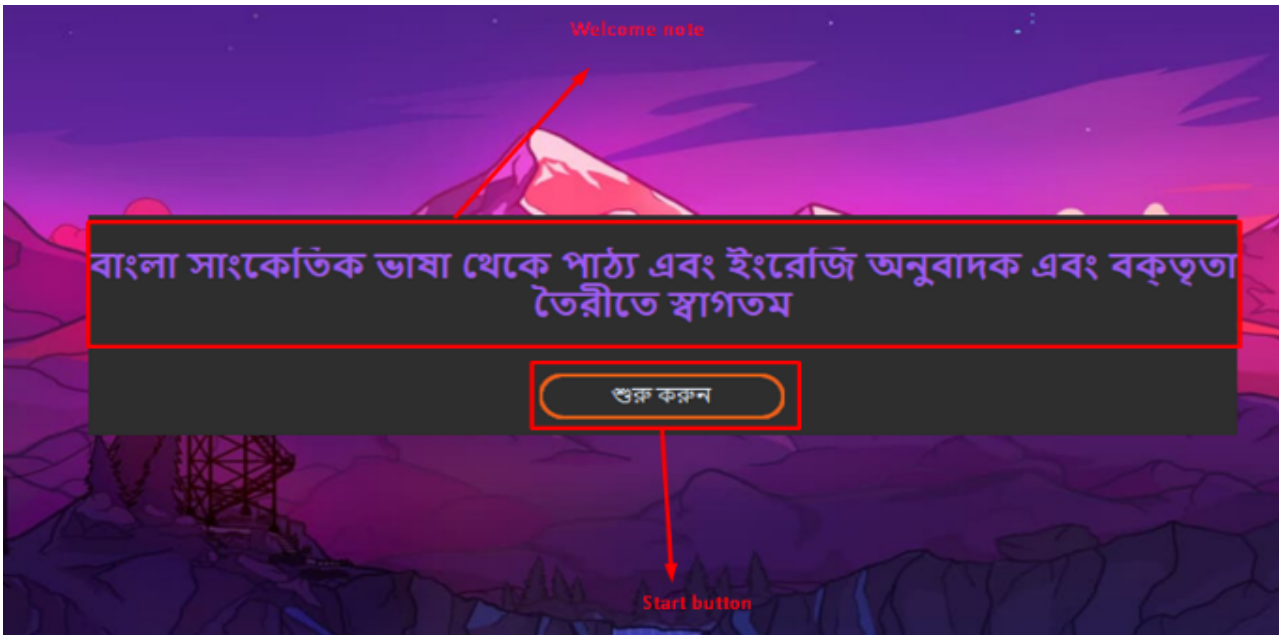

Figure 17 Welcome Page

Bengali Literary language and English translation and audio/speech output are the properties of this App. The welcome message, "বাংলা সাংস্কৃতিক ভাষা থেকে পাঠ্য এবং ইংরেজি অনুবাদক ও বক্তৃতা তৈরিতে স্বাগতম," means "Welcome to text and English translation and speech generation from Bengali literary language." And added some explain that this literally explain what is the primary functionality of the application. The first button prompts the user through the main features of the app. The simple interface allows any user the ability to take advantage of these features seamlessly in **Figure 17**.

4.7.2 Sign language interface including voice in both Bengali and English

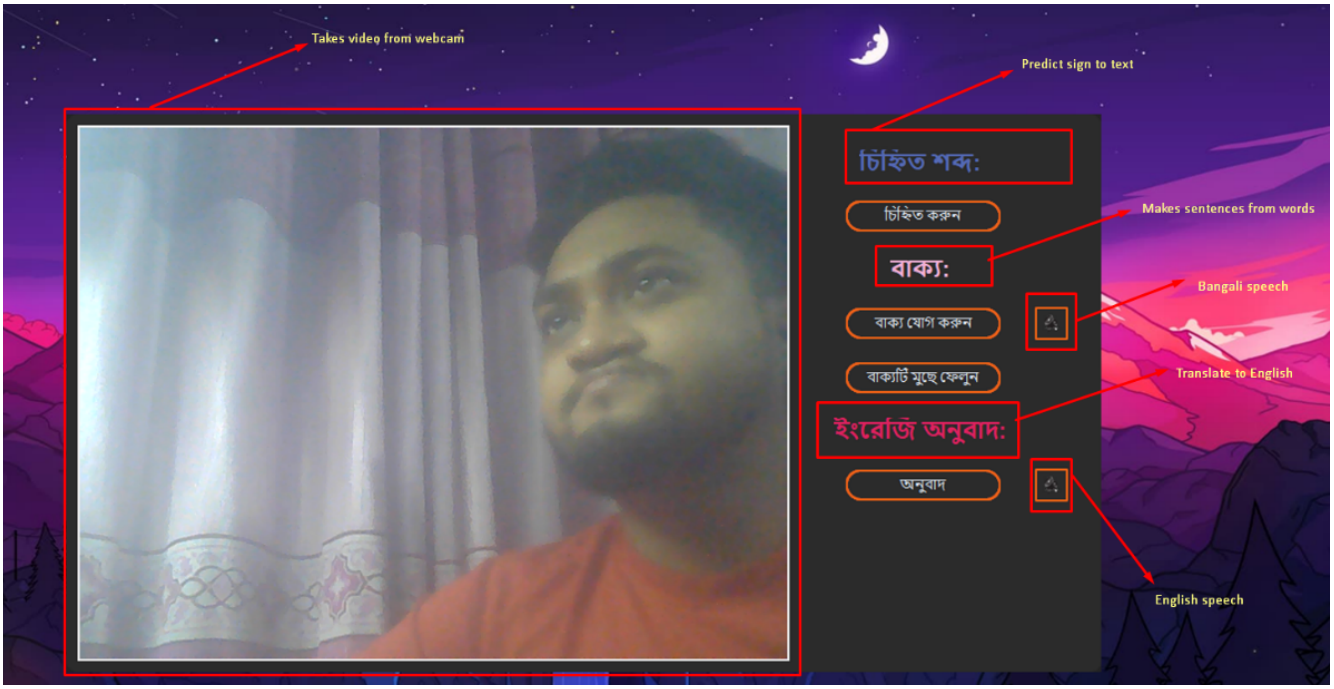

Figure 18 Main Interface

As we see in **Figure 18**, this is the main interface, here we have the video frame which takes the video from the webcam. This is the most important part of our project. Because without this we cannot predict the sign words. Then we have the function of চিহ্নিত শব্দ

(Identified Word) which displays the word corresponding to the recognized gesture. বাক্য (Sentence) combines multiple recognized words into a coherent sentence by clicking add the word button. We can clear the sentence by clear the sentence button. ইংরেজি অনুবাদ (English Translation) shows the English translation of the recognized sentence. Takes sentence from the বাক্য (Sentence) and translate it to English, and this happens when the অনুবাদ (translate) button is clicked. This button converts the sentence into speech in Bengali with English also.

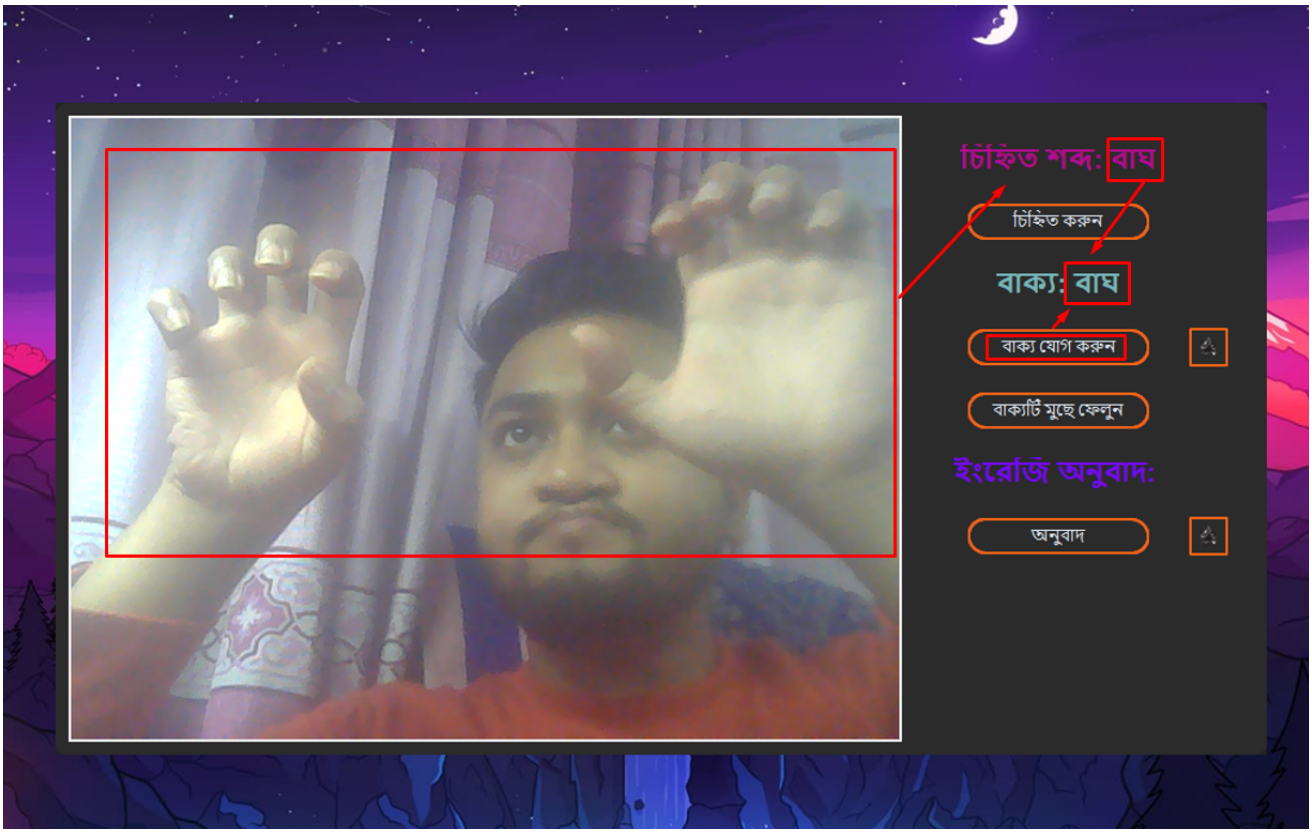

Figure 19 Prediction of Tiger (বাঘ)

We have discussed earlier about our Main interface in **Figure *19***. We just discuss about that what is try to show in this figure. Here the application can predict the sign the word and the বাক্য sentence are take that word also correctly by the button of বাক্য যোগ করুন add the sentence. Updates the বাক্য (Sentence) section with the constructed Bengali sentence.

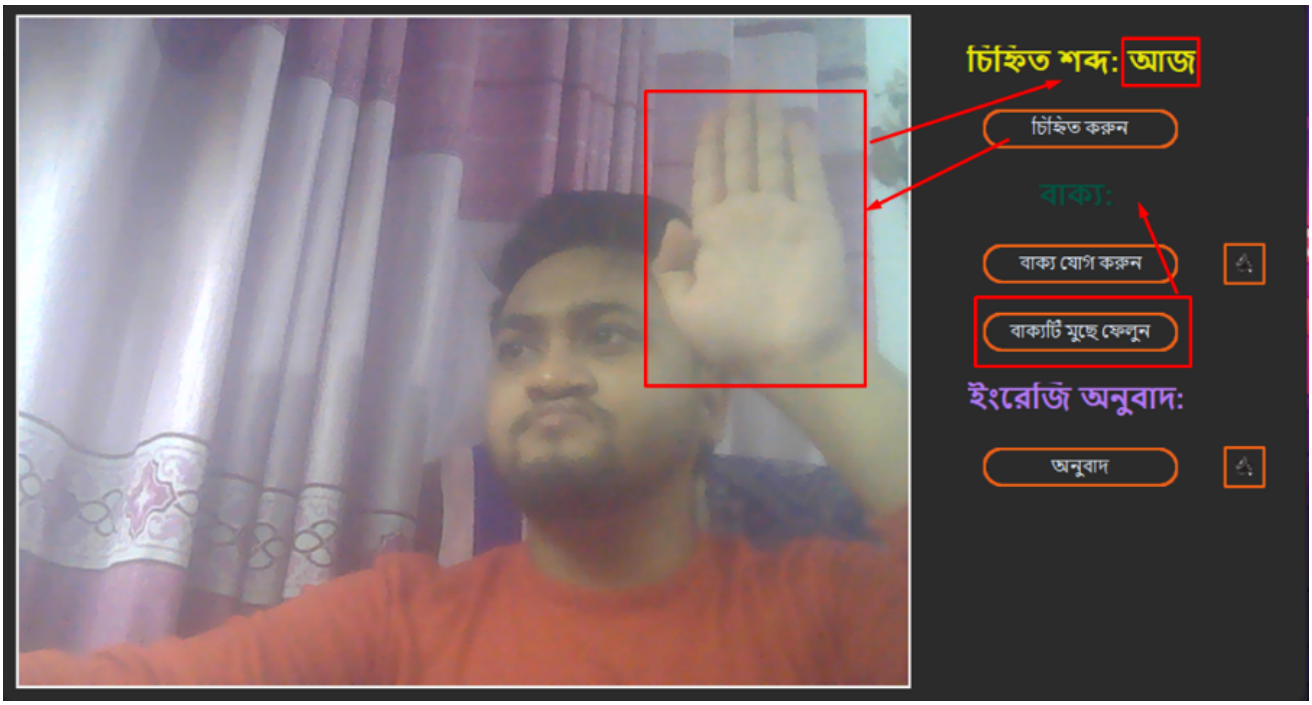

Figure 20 Working of clear button

This **Figure 20** shows that the function of the বাক্যটি মুছে ফেলুন buttons working.

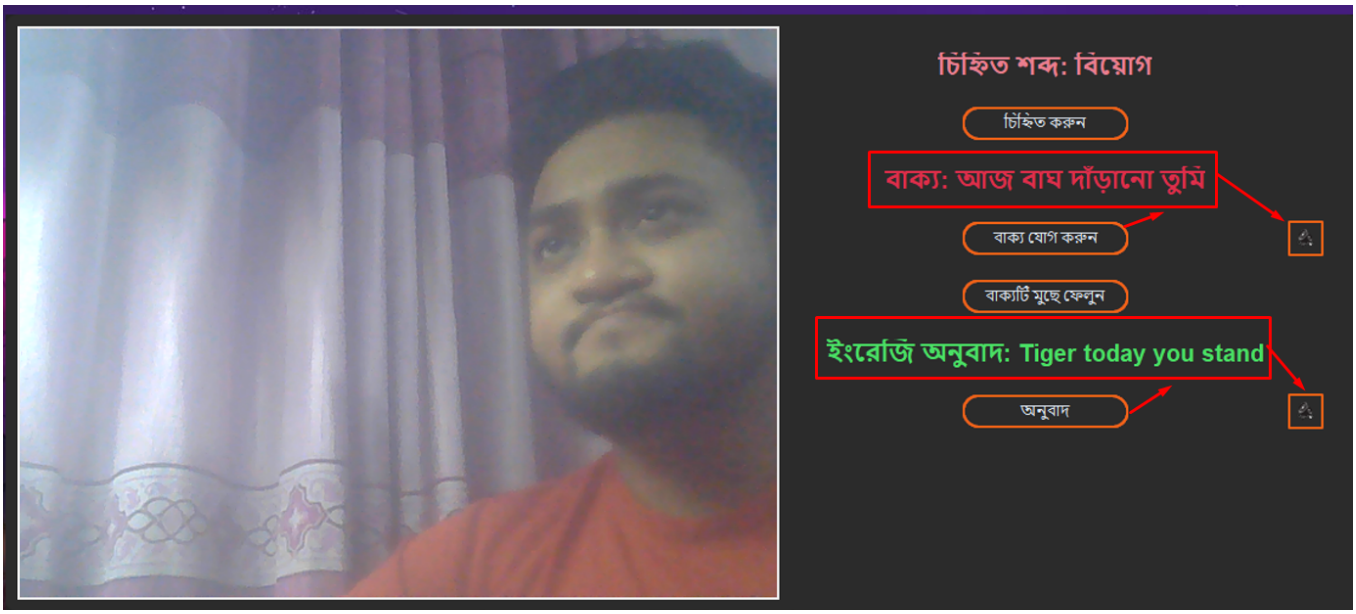

Figure 21 Sentence Formation of 2-3 words

Figure 21 contains the sentence formation part. For the lack of image dataset this part is still facing problem to make sentences. বাক্য: আজ বাঘ দাঁড়ানো তুমি This section shows the full sentence being constructed in Bengali by adding recognized words sequentially. বাক্য যোগ করুন (Add Sentence) used to append recognized words to

this sentence. অনুবাদ (Translate) this button translates the constructed Bengali sentence into English from the বাক্য (Sentence).

4.8 Discussion

The Real-time Bengali Sign Language recognition system [18] has proved its high sensitivity in detecting and classifying complex hand gestures with considerable accuracy and effectiveness, which results in smooth and proper communication for everyone. It is working-bilingual elegance, as it renders nuances of gesture into Bengali flow-text, and then English text via speech synthesis with pitch modulation, makes it incredibly useful for complex cross-lingual interactions across inter-lingual barriers with each partner. The clever way the system is able to create 2–3 word phrases, meaningful yet changeable, swaps this input to any number of combinations, offers the possibility of much more complex communication The interface has a well-crafted login/registration page, a layout which allow us to explore, and a one-click to select prompt, only enhances the accessibility and useability of the advanced application for everybody.

But despite advances, some intimidating barriers impede the full scope of the system. Environmental dependencies, including low light conditions, cluttered backgrounds, and diverse user postures, lead to decreased gesture recognition performance [15]. The requirement for a bystander to confirm two-handed movements with the “select” button strips away autonomy and makes it less user-friendly. Latency on-real time working prevents exchanges from being smooth, the limited training data limits the generalization of the model. On top of this, the graphics in the interface are also flat. However, text-to-speech output often does not produce natural and contextually appropriate speech, particularly in Bengali, resulting in belief and satisfaction of users diminished.

CHAPTER 5 Impact of the Project

5.1 Impact of this project on Societal, Health, Safety, Legal, and Cultural Issues

The genesis of a real-time Bengali Sign Language system has far-reaching social, health, safety, legal and cultural implications. It improves social understanding of Bengali words and maintains an easy-to-learn format through visualization sign categories that make language learning fun and accessible for educational purposes. Based on this their well-practiced abilities to hear nothing would ensure that this accessibility would be improved as these attendees would contribute to communication and participation, and would be encouraged from a safety and health perspective. The project is legal because it strictly follows ethical guidelines, meaning that the use of data is by the law to ensure that copyright is not infringed and also prevents personal information from being abused so that confidence is maintained within the stakeholders and users. In terms of culture, it is a commendable effort in preservation and evolution of the Bengali language as this blends it with the technology which keeps enriching Bengali culture and preserving it for the coming generations. By focusing on Bengali Sign Language, our project honors and safeguards the cultural identity of the deaf individuals in Bengal. It increases awareness of the linguistic and cultural relevance of BdSL, which encourages general public respect and understanding

5.2 Impact on Environment and Sustainability

Sustainability in the environment is emphasized in the real-time Bengali Sign Language system. By using pre-trained models, the project significantly reduces computing expenses, which saves energy and resources. Without compromising functionality, the method guarantees that the system is both cost-effective and environmentally benign. Additionally, throughout the training and deployment phases, carbon footprints are minimized by making the most efficient use of cloud platforms or GPUs. By employing these strategies, the project encourages a more sustainable and ecologically friendly method of developing machine learning in addition to fostering technological innovation.

5.3 Enhancing Accessibility for the Deaf and Hard of Hearing Community

By enabling real-time gesture recognition and translating signs into speech and text, the system helps people who use Bengali Sign Language (BdSL) communicate more effectively. This promotes inclusion and accessibility in both personal and professional contexts by making it easier to communicate with those who do not understand sign language.

5.4 Promoting Bilingual Communication

By translating gestures into Bengali and English, the system's bilingual text-to-speech feature serves a multilingual audience. This characteristic facilitates greater communication and understanding across language boundaries, making it especially useful in multilingual cultures and global environments.

5.5 Empowering Education and Learning

The initiative helps people learn and practice sign language gestures efficiently by acting as an instructional instrument for Bengali Sign Language instruction. Additionally, it can be incorporated into classrooms to promote fairness in learning environments and assist deaf children in receiving an inclusive education.

5.6 Real-World Applications and Scalability

The real-time functionality ensures practical usability in dynamic environments, such as hospitals, banks, and government offices, where rapid and accurate communication is critical. From mobile applications to extensive institutional implementations, its scalability enables adaptability to a wide range of use cases.

CHAPTER 6 Complex Engineering Problems and Activities

6.1 Complex Engineering Problems (CEP)

Table 4 Complex engineering problem attributes

Problem	Attributes	Addressing the Complex Engineering Problems (P) in the Project
P1	Depth of knowledge required (K3-K8)	The project requires knowledge of Machine Learning Algorithms (K5), Deep Learning Models like MobileNet-V2 (K6), and Scientific Research Papers (WK8) on Sign Language Recognition and Text-to-Speech Systems.
P2	Range of conflicting requirements	High accuracy and real-time performance require more computational resources such as GPU memory and processing time. Balancing latency and accuracy is a challenge.
P3	Depth of analysis required	Multiple frameworks and tools like TensorFlow, PyTorch, OpenCV, and Gradio can be used. An optimal combination of preprocessing, feature extraction, and recognition methods is required.
P4	Familiarity of issues	Familiarity with Python, PyTorch, OpenCV, Huggingface, and TensorFlow is crucial for implementing the model and building the system.
P5	Extent of applicable codes	Existing pretrained models such as MobileNet-V2 are adapted for hand gesture recognition and enhanced with custom layers for bilingual text-to-speech conversion.
P6	Extent of stakeholder involvement	Stakeholders include sign language experts, translators, Bengali language professionals, and educational institutions for system validation.
P7	Interdependence	Dependency on compatible versions of libraries (e.g., TensorFlow, PyTorch, Gradio) and integration between sign recognition and text-to-speech modules.
P8	Dataset Challenges	Limited availability of comprehensive and diverse BdSL datasets.
P9	TTS Model Development	Ensuring natural-sounding speech in both languages.
P10	User Interface Design	Creating an intuitive interface for diverse user groups.
P11	Cross-Lingual Transfer Learning	Difficulty in achieving high-quality TTS for Bengali.

6.2 Complex Engineering Activities (CEA)

Table 5 Complex engineering problem activities

Problem	Attributes	Addressing the complex engineering problems(P) in the project
A1	Error Handling and Robustness	Implement error handling and fallback mechanisms to ensure robustness.
A2	Latency Minimization	Optimize system components to reduce latency. Achieving real-time system performance.
A3	Cameras and Sensors	Ensure use of high-quality cameras, depth sensors.
A4	Range of resources	This project involves human resources (language experts, developers), financial resources for computational infrastructure, and modern tools like Google Colab.
A5	Level of interactions	Requires collaboration among developers, linguists, and sign language experts. Interaction with users for feedback during testing and validation is also necessary.
A6	Innovation	Introduces innovative bilingual technology for recognizing Bengali sign language gestures and converting them into text and speech in Bengali and English.
A7	Consequences to society	Positively impacts society by enabling communication for people with hearing or speech impairments, improving accessibility in educational and public spaces.
A8	Familiarity	Familiarity with machine learning tools (e.g., TensorFlow, PyTorch) and cloud-based computational environments (e.g., Google Colab) is essential for implementation.
A9	Challenges in Bilingual TTS	Address the phonetic and syntactic differences between Bengali and English in TTS models.
A10	Image Processing and Feature Extraction	Accurate detection and interpretation of hand gestures. Use hand landmark detection via tools like MediaPipe.

CHAPTER 7 Conclusion

7.1 Summary

This paper presents a novel approach to real-time Bengali Sign Language (BSL) recognition [22], that is a manipulation of BdSL gestures alongside the text-to-speech conversion and Bengali-English translation integrated in the text output that meets the basic needs of accessibility for the Deaf as facing complication in everyday communication. Real-time gesture detection is ensured using a complex framework of Bengali Sign Language (BSL) for dynamic input processing and categorization, and providing seamless interaction. The system improves its communication capabilities by identifying gesture patterns which correspond to multiple words (typically two to three) and composing sentences. This enables the construction of self-contained micro-phrases which enhance real-time communication. A full-fledged advanced pipeline implemented in the text conversion procedure of the real-time gesture recognition system receiving commands of recognized motions and providing a synchronized interactive experience for a seamless multilingual dialogue. Then, after successfully transforming Bengali gestures, motion and movement of fingers and hands into textual format (for the most part in English) [8], the system passes the generated text to an integrated text to speech (TTS) engine, which produces the audio. There is minimal research on Sign Language that covers most of what we need to know.

7.2 Limitations

This work, like all of research, has its limitations, which is the reason it can, and should, achieve better outcomes. One thing to take into consideration also, is that the system is sometimes incorrect when the user is standing up — it seems to perform much better when sitting down and ensuring to look directly at the camera. Variations in hand gestures due to differences in size, shape, orientation, and angles lead to reduced generalization across diverse users. The interface has usability issues, including latency, errors in gesture-to-text mapping, and sentence formation inconsistencies. When both hands are used for gestures, a third person is required to click the "select" button, making the system less independent. The bilingual text-to-speech (TTS) system faces challenges in ensuring natural and clear speech synthesis, particularly in timing and pronunciation. Moreover, because of the hand gestures be performed against an unclear or messy background, it makes it difficult for the system to identify gestures by ensuring that it runs smoothly in real time via the accuracy of the decision of the model. Apart from this, poor lighting conditions tend to be common as well, wherein the clarity of gestures diminishes and gestures may be misinterpreted. Inadequate collaboration with the deaf community may result in inaccurate representations of Bengali Sign Language (BdSL).

7.3 Future Improvements

Our team’s future work will focus on enhancing the system’s capabilities to better serve a diverse range of users. We will optimize the system for varying user postures, including standing positions, ensuring consistent performance regardless of the user's orientation. We will also integrate advanced background subtraction techniques and noise reduction methods to improve gesture accuracy in cluttered or unclear environments. Additionally, we will explore lighting normalization algorithms to enhance the system’s performance in low-light or uneven lighting conditions, ensuring reliable gesture recognition in various settings. Our work will also include expanding the system to support more complex sentence formation, enabling longer sequences of gestures for natural and fluent communication.

To further improve the system, we will plan to incorporate adaptive learning capabilities, allowing the system to customize its predictions based on individual user variations, such as hand size, orientation, or motion speed. These improvements will make the system more versatile, accessible, and effective for the Deaf community and a broader user base. Equally important is the refinement of the graphical and interactive dimensions of the application, necessitating the involvement of dedicated UI/UX expertise to develop an engaging, visually appealing, and user-friendly frontend interface. Moreover, gaining a profound understanding of Bengali word signs is vital for improving the accuracy and efficiency of the system in recognizing and predicting signed words in real-time, thereby reducing latency and enhancing performance. Addressing these challenges demands a focused commitment to research, innovation, and hard work, ensuring that the system remains adaptable to future advancements and continues to meet the evolving needs of its users.

References

[1] S. Akash, D. Chakraborty, M. Kaushik, B. Babu and M. Zishan, "Action Recognition Based Real-time Bangla Sign Language Detection and Sentence Formation," in *2023 3rd International Conference on Robotics, Electrical and Signal Processing Techniques (ICREST)*, Dhaka, 2023.

[2] B. C. Karmokar, K. M. R. Alam and M. K. Siddiquee, "Bangladeshi Sign Language Recognition employing Neural Network Ensemble," *International Journal of Computer Applications (0975 – 8887),* vol. 58, no. 16, pp. 43-46, 2012.

[3] A. Pathak, A. Kumar, P. P. Gupta and G. Chugh, "Real Time Sign Language Detection," *International Journal for Modern Trends in Science and Technology,* vol. 8, no. 01, pp. 32-37, 2022.

[4] A. Haque, R. A. Pulok, M. M. Rahman, S. Akter, N. Khan and S. Haque, "Recognition of Bangladeshi Sign Language (BdSL) Words using Deep Convolutional Neural Networks (DCNNs)," *Emerging Science Journal,* vol. 7, no. 6, pp. 2183-2201, 2023.

[5] N. Begum, R. Rahman, N. Jahan, S. S. Khan, T. Helaly, A. Haque and N. Khatun, "Borno-Net: A Real-Time Bengali Sign-Character Detection and Sentence Generation System Using Quantized Yolov4-Tiny and LSTMs," *Applied Sciences,* vol. 13, no. 9, pp. 1-16, 2023.

[6] D. M. Khan, M. Z. Alom, N. N. Choudhury and G. Kayas, "Automatic Recognition of Bangla Sign Language," BRAC University, Dhaka, 2012.

[7] M. Uddin, "Creating Multiclass Bangladeshi Sign Word Language for Deaf and Hard of Hearing People and Recognizing using Deep CNN Techniques," University of Information Technologyand Sciences (UITS), Dhaka, 2024.

[8] S. Siddique, S. Islam, E. E. Neon, T. Sabbir, I. T. Naheen and R. Khan, "Deep Learning-based Bangla Sign Language Detection with an Edge Device," *Intelligent Systems with Applications,* vol. 18, no. 200224, pp. 1-12, 2023.

[9] M. Islam, M. M. Rahman, M. H. Rahman, M. Arifuzzaman, R. Sassi and M. Aktaruzzaman, "Recognition Bangla Sign Language using Convolutional Neural Network," in *2019 International Conference on Innovation and Intelligence for Informatics, Computing, and Technologies (3ICT)*, Kustia, 2019.

[10] H. Rubaiyeat, H. Mahmud, A. Habib and M. K. Hasan, BdSLW60: A Word-Level Bangla Sign Language Dataset, Dhaka: 10.13140/RG.2.2.12193.79202, 2024/02/14.

[11] D. Talukder and F. Jahara, "Real-Time Bangla Sign Language Detection with Sentence and Speech Generation," in *2020 23rd International Conference of Computer and Information Technology (ICCIT)*, Chittagong, 2021.

[12] M. W. Foysol, S. Sajal and M. Alam, "Vision-based Real Time Bangla Sign Language Recognition System Using MediaPipe Holistic and LSTM," Daffodil International University, Dhaka, 2023.

[13] T. Abedin, K. Prottoy, A. Moshruba and S. Hakim, Bangla sign language recognition using concatenated BdSL network, Dhaka: 10.48550/arXiv.2107.11818, 2021/07/25.

[14] O. Hoque, M. I. Jubair, M. Islam, A.-F. Akash and A. Paulson, Real Time Bangladeshi Sign Language Detection using Faster R-CNN, Dhaka: 10.48550/arXiv.1811.12813, 2018/11/30.

[15] R. Pranto and S. Siddique, Real-time Bangla Sign Language Translator, Dhaka: 10.48550/arXiv.2412.16497, 2024/12/21.

[16] T. ABEDIN, K. S. S. PROTTOY and A. Moshruba, "Bangla Sign Language Recognition Using Concatenated BdSL Network," Islamic University of Technology (IUT), Dhaka, 2021.

[17] N. Malik and N. Walia, "ML-BASED HAND SIGN DETECTION SYSTEM FOR DEAF-MUTE PEOPLE," *International Journal of Advance and Innovative Research ISSN: 2394-7780,* vol. 09, no. 02, pp. 43-48, 2022.

[18] H. Mural, S. M. F. Faisal and S. Banik, "Real Time Bangla Sign Language Detection Using Action Recognition," Chittagong University of Engineering and Technology, Chittagong, 2024.

[19] S. Shurid, K. Amin, M. MirBahar, D. Karmaker, M. Mahtab, F. Khan, M. G. R. Alam and M. A. Alam, "Bangla Sign Language Recognition and Sentence Building Using Deep Learning," in *2020 IEEE Asia-Pacific Conference on Computer Science and Data Engineering (CSDE) | 978-1-6654-1974-1/20/$31.00 ©2020 IEEE*, Dhaka, 2021.

[20] A. Hasib, J. F. Eva, S. S. Khan, M. N. Khatun, A. Haque, N. Shahrin, R. Rahman, H. Murad, M. R. Islam and M. R. Hussein, " BDSL 49: A comprehensive dataset of Bangla sign language," *Data in Brief,* vol. 49, pp. 2-10, 2023.

[21] A. Hasib, S. S. Khan, J. F. Eva, M. N. Khatun, A. Haque, N. Shahrin, R. Rahman, H. Murad, M. R. Islam and M. R. Hussein, "A Comprehensive Dataset of Bangla Sign Language," *Elsevier,* vol. 49, pp. 2-8, 2023.

[22] G. S. Surjo, B. K. Ghosh, M. J. Alam, M. M. Razib and S. Bilgaiyan, "A comparative analysis between single & dual-handed Bangladeshi Sign Language detection using CNN based approach," in *2023 International Conference on Computer Communication and Informatics (ICCCI -2023), Jan. 23-25, 2023, Coimbatore, INDIA*, Delhi, 2023.