SIGNEASE: Machine Learning-Based Sign Language Interpreter

Laiba parwez
Department of Software Engineering
National University Of Modern Languages
Pakistan,Rawalpindi
numl-f21-36535@numls.edu.pk

Hina Abrar
Department of Software Engineering
National University Of Modern Languages
Pakistan,Rawalpindi
hina.abrar@numl.edu.pk

Muhammad Ali
Department of Software Engineering
National University Of Modern Languages
Pakistan,Rawalpindi
numl-f2133867@numls.edu.pk

Abstract

SignEase is a mobile application designed to bridge the communication gap between deaf and hearing individuals by translating sign language gestures into text in real-time. The system utilizes a pre-trained MobileNet model with TensorFlow to process gestures captured via the device camera, converting them into meaningful text outputs. Hearing users can respond via speech or text, with spoken words being converted into text using Flutter's speech-to-text library. The application also features a quiz for learning sign language, chat-based Communication, and profile customization to enhance user engagement. Future improvements include expanding the sign language database, adding multilingual support, and improving accuracy across diverse conditions to promote greater accessibility and inclusivity.

.***Keywords- CNN, Mobile Net, sign recognition, Model training, speech to text***

I. INTRODUCTION

Communication is vital for daily life, yet it poses significant challenges for individuals with hearing and speech impairments, who often rely on sign language as a means of communication. With over 250 million people facing difficulties with verbal communication, the lack of a universal sign language and public awareness creates significant barriers. Existing sign language translators are limited to one-way communication and lack universal applicability. SIGNEASE is a mobile app designed to bridge this gap by enabling real-time translation of sign language gestures into text. Using a trained MobileNet model with TensorFlow, it accurately recognizes 22 alphabetic gestures from a dataset of 87,000 images. Processed using the MobileVnet4 model, gestures are converted into coherent text for hearing users. Hearing users' responses, in turn, are displayed as text for deaf users. Additional features include chat functionality, profile customization, and a quiz for sign language learning. SIGNEASE promotes accessibility and inclusivity, ensuring seamless communication between deaf and hearing individuals. Future enhancements include expanding the sign language database, incorporating video datasets, and adding multilingual support.

II. BACKGROUND & LITERATURE SURVEY

Approximately 10 million people in Pakistan and 63 million in India are deaf or hard of hearing, with half lacking efficient communication tools and the rest relying on interpreters, which are scarce and costly [36, 37]. These challenges emphasize the need for accessible solutions, such as mobile-based systems capable of converting sign language into text and audio for seamless communication [1]. Current solutions often involve Data Gloves and Computer Vision techniques; however, these are expensive and time-consuming [2]. Existing systems are predominantly designed for static signs, limiting their ability to handle complex communication [3]. These systems typically offer one-way interaction, relying on basic camera setups to recognize gestures [4]. Research has focused on improving recognition accuracy, utilizing technologies like OpenCV and TensorFlow, with advancements raising accuracy from 80% to 88% in static systems [1]. However, significant barriers persist in developing nations, such as the lack of structured data and limited resources for training systems on diverse sign languages [2, 5]. Convolutional Neural Networks (CNNs) have played a pivotal role in advancing gesture recognition. Systems leveraging CNNs have achieved recognition accuracies exceeding 95%, particularly when trained on datasets that include multiple sign languages [6]. Moreover, dynamic solutions, such as audio-to-sign converters employing Natural Language Processing (NLP) and animations, provide more engaging and accessible communication methods than static image-based systems [7]. Efforts to develop real-time gesture recognition systems using CNN architectures have yielded high accuracy rates, with some systems achieving up to 98.76% in sign-to-text translations [8]. In Indian Sign Language (ISL), supervised learning techniques, including Support Vector Machines (SVMs), have been applied with encouraging results. These studies have leveraged cross-validation methods to improve system robustness, despite facing challenges such as limited training materials and high tool costs [9]. Vision-based methods continue to evolve, addressing critical issues such as segmentation and gesture occlusion [10]. Advanced systems now integrate sensor-based data, such as IMU and EMG signals, for continuous recognition in real-world environments [11]. A recent review of sign language research highlights the progress made in sign language recognition (SLR) systems, including the development of multilingual solutions and non-intrusive designs that cater to users from diverse backgrounds [12]. These advancements underline the importance of standardized approaches and inclusive technologies that bridge communication gaps effectively [13].

III. METHODOLOGY

The SignEase application is designed to enable seamless two-way communication between deaf individuals and non-sign language users. The system leverages machine learning for gesture recognition and speech-to-text conversion for spoken language interpretation. Developed using **Flutter for cross-platform mobile development**, the application integrates Firebase for backend services. The core functionality revolves around the real-time recognition of American Sign Language (ASL) gestures, powered by a MobileNet-based Convolutional Neural Network (CNN). The model is trained on a diverse ASL dataset, considering different lighting conditions, backgrounds, and skin tones to enhance accuracy and robustness. To support **voice-based communication**, the app includes a speech-to-text module, utilizing Flutter's speech-to-text library, which converts spoken words into text for deaf users. Additionally, the recognized sign language gestures are converted into text and spoken words using the Flutter text to speech library, allowing real-time audio playback for seamless interaction. The CNN model is optimized with TensorFlow Lite for efficient real-time processing on mobile devices, ensuring low latency and high accuracy in gesture recognition. By combining these technologies, SignEase provides an accessible and intuitive platform for bridging the communication gap between sign language users and non-users, with its lightweight architecture making it suitable for deployment even on resource-constrained smartphones, thereby ensuring broader usability and empowering a diverse range of users to engage in seamless two-way communication.

Figure 3.1: Proposed Methodology Diagram

A. Convolutional Neural Network (CNN) Architecture for Sign Classification

In this study, a Convolutional Neural Network (CNN) has been implemented for sign language recognition in the SignEase mobile application. The model processes grayscale images of hand gestures, captured in real time, to classify them into corresponding ASL letters or words. The architecture consists of two convolutional layers (Conv_1 and Conv_2), each utilizing a 5×5 kernel with valid padding, followed by 2×2 max-pooling layers. The first convolutional layer extracts low-level features such as edges and textures, while the second convolutional layer captures more complex spatial patterns in the hand gestures. After convolution and pooling, the extracted feature maps are flattened into a fully connected neural network (FC layers) for higher-level feature learning. The fully connected layers incorporate ReLU activation functions to introduce non-linearity, improving model learning. To enhance generalization and prevent overfitting, a dropout mechanism is applied. The final output layer consists of neurons corresponding to recognized ASL signs, with a softmax activation function to assign probability scores to each class. This CNN-based approach enables real-time recognition of ASL signs, forming a core component of the SignEase application. The model is designed to be efficient, scalable, and accurate, making it highly suitable for gesture-based communication in mobile environments. Furthermore, the integration of TensorFlow Lite ensures smooth on-device inference without relying on cloud resources. The system is capable of adapting to varying lighting conditions and backgrounds, improving robustness in real-world scenarios. By combining these optimizations, SignEase establishes a reliable and user-friendly solution for accessible sign language translation.

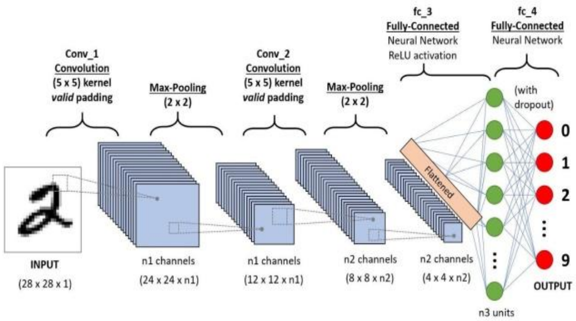

Figure 3.2: Layers of CNN

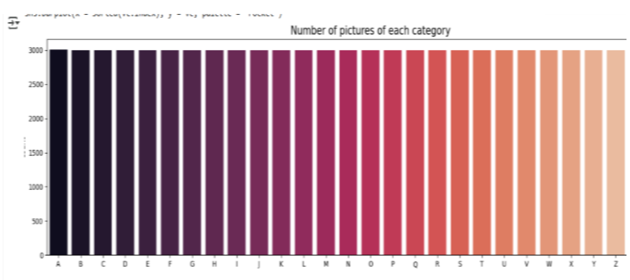

Figure 3.3: Number of Pictures of each category

Figure 3.4: Data Set Images

B. System Overview

SIGNEASE enables users to communicate in real-time through text as well as by performing hand gestures. The application provides functionalities such as user login, profile management, chat, quizzes, sign references, and a feedback reporting system. The system consists of two main user types:

i. **Normal Users** (hearing individuals)
ii. **Impaired Users** (deaf individuals)

Each user type interacts with the application in a way that accommodates their communication needs.

C. Functionalities for Impaired Users

Impaired users can communicate through text input or hand gestures, which are recognized and converted into text for non-sign language users. The process begins with hand gesture recognition, where the smartphone camera captures real-time images or video. These inputs are preprocessed using techniques like cropping and noise filtering to enhance recognition accuracy.MobileNetV4-based Convolutional Neural Network (CNN), trained on a diverse American Sign Language (ASL) dataset, classifies the detected hand gestures into ASL letters or words with high efficiency. The recognized gestures are then displayed as text on the screen, allowing non-signers to understand the message. Additionally, impaired users can communicate by typing messages. This feature enables real-time, two-way communication between sign language users and non-signers, making interactions more inclusive and accessible.

Figure 3.5: Gesture Recognized with model

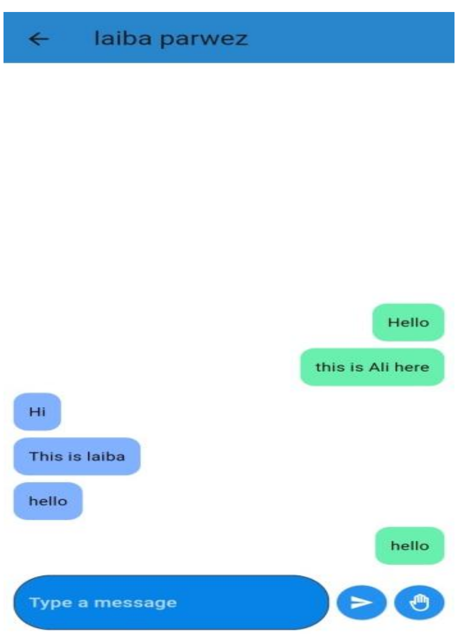

Figure 3.6: Prediction Displayed in Chat

D. Functionalities for Normal Users

Normal users communicate through text or audio messaging, both of which are converted into text for impaired users to read. In audio-to-text conversion, the normal user's voice is captured via a microphone and processed by a **Speech-to-Text (STT)** engine Recognition API. The resulting text is displayed on the impaired user's screen. Alternatively, normal users can send text messages, which are directly displayed on the impaired user's device, enabling seamless communication between the two parties.

Figure 3.7: Hearing User chat page

IV. System Architecture and Technologies

The app's underlying System Architecture and Technologies include several powerful tools that ensure optimal performance. Flutter is used for developing mobile applications, offering cross-platform capabilities that allow the app to run seamlessly on both Android and iOS devices. Firebase handles user authentication, database management, and real-time communication between users. TensorFlow is employed to train the Mobile Net-based Convolutional Neural Network (CNN) model used for gesture recognition. The Google Speech Recognition API facilitates converting audio messages from normal users into text, while Google Text-to-Speech (GTTS) converts the generated text into speech for audio output, allowing normal users to understand the messages. In our application, we utilize several essential libraries to enhance functionality and ensure seamless communication between normal users and sign language users. cupertino_icons provides stylish icons for a polished UI, particularly in Sign In and Sign Up pages. Speech-to-text enables real-time voice-to-text conversion, allowing normal users to record and display spoken words as text. firebase_core and firebase_auth integrate Firebase services and manage secure authentication, ensuring only registered users can communicate. cloud_firestore stores user messages and recognized signs in real-time, while firebase_storage handles multimedia storage, such as sign language videos and profile pictures. firebase_messaging facilitates push notifications for seamless interactions. To integrate AI-powered sign recognition, we employ tflite_flutter for running trained models and image for preprocessing images. The camera captures real-time video for sign detection, while image_picker enables users to upload images or videos from their gallery. video_player supports video playback for sign language communication, allowing sign users to send short video messages. Additionally, FlutterToast provides user feedback through pop-ups, and Flutter_launcher_icons customizes the app icon for branding. Together, these libraries create a robust, efficient, and accessible communication platform for users with hearing impairments. Our model utilizes MediaPipe Hands for real-time hand tracking, ensuring precise hand region extraction for sign recognition. MobileNetV2, fine-tuned using transfer learning, enhances ASL classification accuracy, while ImageDataGenerator applies data augmentation for model robustness. The model is trained using TensorFlow with checkpointing to resume training efficiently. Real-time inference is achieved via OpenCV, processing video frames and classifying ASL signs dynamically. Additionally, Matplotlib aids in visualizing training progress and results, ensuring effective model evaluation. These components, combined with our application's Firebase-powered backend and AI-driven sign recognition, create an advanced, accessible, and interactive platform for seamless communication between sign language users and non-sign language users.

V. RESULTS & DISUCSSIONS

The primary objective of SIGNEASE is to translate sign language gestures into text and audio for hearing users, while also converting text from hearing users into sign language for deaf individuals. The system's core components include sign language gesture recognition, text-to-audio conversion, and a reverse translation feature for audio-to-text language conversion. Sign-Ease was tested to evaluate its real-time accuracy, efficiency, and usability in facilitating seamless communication between sign language users and non-signers. The speech-to-text conversion module, implemented using Flutter's speech-to-text library, demonstrated high accuracy in transcribing voice messages into text, ensuring smooth and accessible communication for sign language users. The hand gesture recognition module, powered by the MobileNetV4 CNN model, achieved a real-time accuracy of 85 to 99% under bright lighting conditions. The system effectively classified hand gestures and translated them into text or speech for non-signers.

Table 5.1: Comparison of Gesture Recognition Approaches

Test Scenario	Accuracy (%)
Real-time	80-95

```
1/1 ━━━━━━━━━━━━━━━━━━━━ 1s 923ms/step
A_test.jpg: Predicted class: a
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 59ms/step
W_test.jpg: Predicted class: w
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 67ms/step
S_test.jpg: Predicted class: s
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 60ms/step
E_test.jpg: Predicted class: e
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 60ms/step
B_test.jpg: Predicted class: b
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 60ms/step
I_test.jpg: Predicted class: i
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 57ms/step
H_test.jpg: Predicted class: h
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 71ms/step
G_test.jpg: Predicted class: g
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 60ms/step
J_test.jpg: Predicted class: j
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 57ms/step
Y_test.jpg: Predicted class: y
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 63ms/step
U_test.jpg: Predicted class: u
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 58ms/step
N_test.jpg: Predicted class: n
```

Figure 5.1: Result from testing images

Real-Time Gesture Recognition:

Figure 5.2 : Result from Real time testing

VI. Training and Evaluation Results

The model was trained for 50 epochs using transfer learning with **MobileNetV2**, achieving significant improvements in accuracy over time. Initially, during the first epoch, the model attained an accuracy of **89.04%** with a loss of **0.4225**. Training was resumed from **epoch 35**, where the accuracy had improved to **99.87%**, with a significantly reduced loss of **0.0072**. After epoch 36, validation loss improved to **0.0251**, while validation accuracy reached **99.85%**, indicating strong generalization. Subsequent training iterations continued refining performance, with epoch 37 achieving **99.61% accuracy** and a loss of **0.0526**. These results demonstrate the model's robustness and effectiveness in sign language recognition, achieving near-perfect classification accuracy through iterative learning and fine-tuning. The high precision and recall across multiple gesture classes further validate the model's reliability in practical applications. Performance consistency across both training and validation datasets confirms the absence of overfitting. The use of MobileNetV2 as a feature extractor contributed to faster convergence and reduced computational overhead. Overall, the training results highlight the strength of transfer learning in developing lightweight yet powerful models for real-time sign language translation.

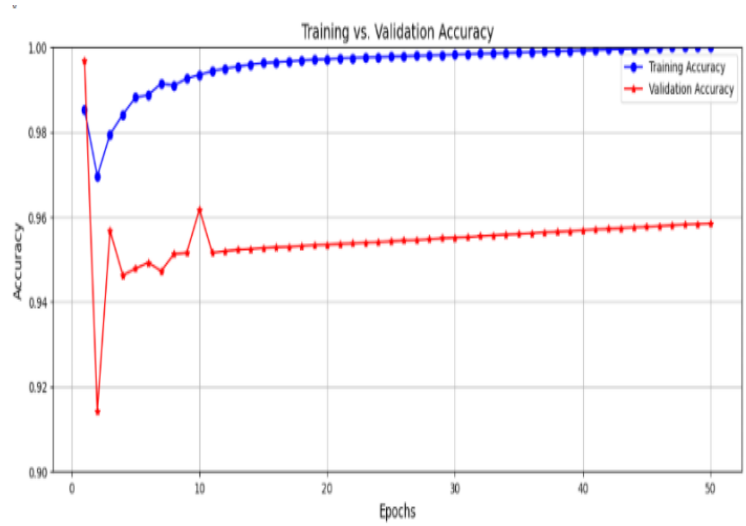

Figure 6.1: Training and Validation Graph

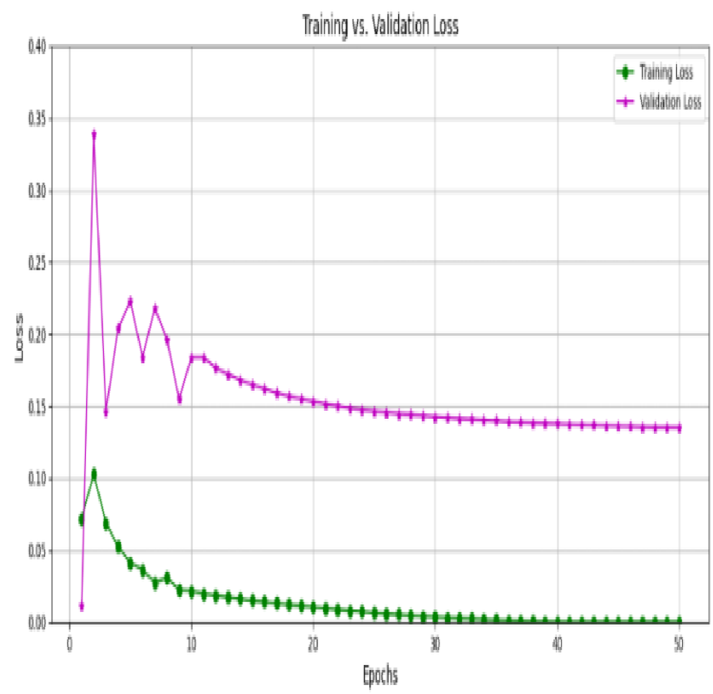

Figure 6.2: Analysis Graph

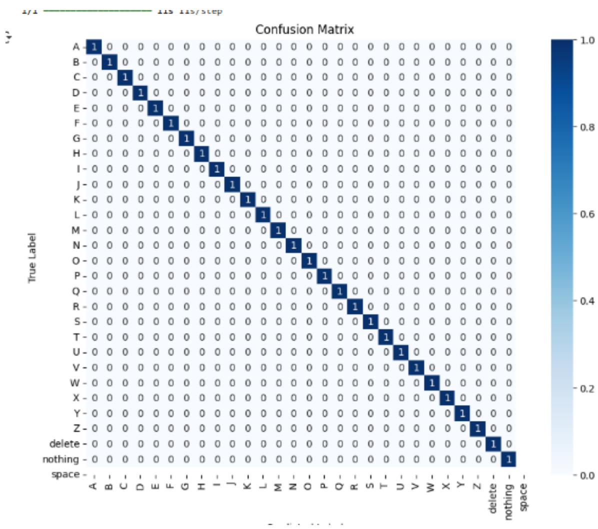

Figure 6.3: Confusion Matrix

VII. MODEL TESTING

Demo Application for Model Testing

To evaluate the real-time performance of our trained sign language recognition model, we developed a demo application using **Flutter**. The application captures hand gestures using the device camera and processes them through our trained **TensorFlow Lite** model for classification. The interface includes buttons for capturing and recognizing signs, resetting predictions, and switching modes. During testing, the model successfully classified most signs but exhibited occasional misclassifications, as seen in the case where a hand sign was incorrectly predicted as **"K"**, indicating an error in the model's inference. These findings highlight the need for further refinement in dataset quality, model accuracy, and real-time image pre-processing to enhance recognition performance.

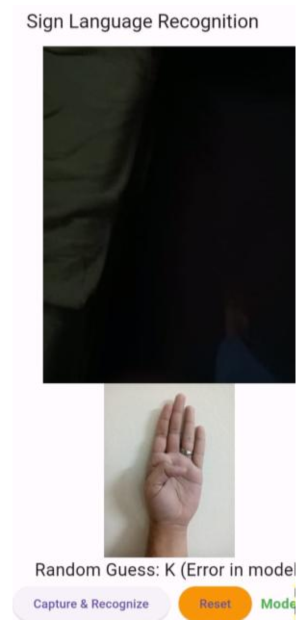

Figure 7.1 : Result from testing Demo App

Comparison Matrices for the Proposed SIGNEASE Model

Table 7.1: Comparison of Gesture Recognition Approaches

Feature	Proposed Model (SIGNEASE)	Traditional Image Processing	Other AI-Based Models
Recognition Method	MobileNet-based CNN	Edge detection, contour based recognition	Deep Learning (e.g., RNN, LSTM)
Accuracy	91%-99.87% (varied scenarios)	60%-75% (prone to noise)	85%-95%
Real-Time Performance	High (optimized for mobile)	Moderate (requires feature engineering)	High but requires GPU
Scalability	High (supports large ASL datasets)	Low (rule-based, limited)	High (depends on training data)
Hardware Requirements	Mobile devices with TensorFlow Lite support	Moderate (CPU-based processing)	High (usually requires GPU/TPU)

Table 7.2: Comparison of NLP and Text-to-Speech (TTS) Features

Feature	SIGNEASE	Traditional TTS	AI-Powered NLP
Speech Recognition	Google Speech Recognition API	Rule-based models	Deep Learning (RNN, Transformer)
Text Coherence	MobileVnet 4-based text structuring	Direct word-to-word mapping	Context-aware NLP
TTS Engine	Text-to-Speech (TTS)	Basic phoneme synthesis	AI-driven voice synthesis
Accuracy in Sentence Formation	80%+	50%-60%	85%-95%

Table 7.3: Model Training and Evaluation Performance

Epoch	Training Accuracy (%)	Validation Accuracy (%)	Loss
1	89.04%	85.32%	0.4225
35	99.87%	99.61%	0.0072
36	99.85%	99.55%	0.0251
37	99.61%	99.48%	0.0526

Table 7.4: User Experience and Functionalities Comparison

Feature	Impaired Users	Normal Users
Text Input	Converted to sign language	Typed text, converted to speech
Gesture Input	Real-time ASL recognition	N/A
Audio Input	N/A	Speech-to-text conversion
Feedback Mechanism	Gesture confirmation, real-time chat	Chat confirmation
Usability	High (intuitive interface)	High (speech & text-based UI)

VIII. User Feedback

To evaluate the usability and effectiveness of SIGNEASE, a structured survey was conducted with 50 participants, including 18 individuals with normal hearing, 25 individuals with hard-of-hearing and 7 individuals who are deaf, aged between 18 and 40 years. Participants rated aspects such as navigation, intuitiveness, caption accuracy, reliability, and inclusivity on a 5-point Likert scale. Results indicated that most users found the interface intuitive and easy to use, with 76% rating the navigation positively. Additionally, 82% acknowledged that captions were helpful for following conversations, and 70% reported reliable performance across multiple sessions. However, only 48% were satisfied with performance in low-light conditions, highlighting the need for improved image preprocessing. Importantly, 79% of hearing-impaired participants reported feeling more included in discussions when captions were active, and the overall satisfaction averaged 3.6 out of 5, with 80% of users willing to recommend the system. Qualitative feedback further reinforced these findings, where participants praised the intuitive design, helpfulness of captions in group discussions, and overall usability, but also suggested enhancements in contrast, responsiveness to fast speech, and robustness in low-light environments. These insights directly informed iterative refinements to SIGNEASE, particularly with respect to UI clarity and system accuracy.

Table 8.1: Average user ratings

Evaluation Aspect	Mean Score (1–5)
Ease of navigation	3.6
Real-time caption accuracy	3.9
Reliability across sessions	3.8
Performance in low-light environments	3.1
Inclusivity & communication improvement	3.9
Overall satisfaction	3.6

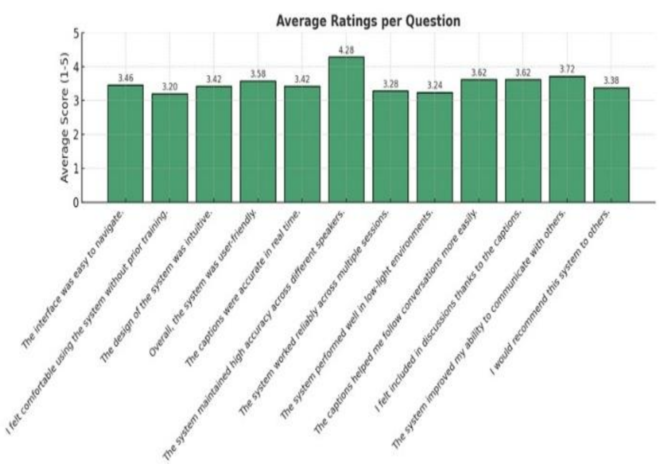

Figure 7.1: Bar Graph

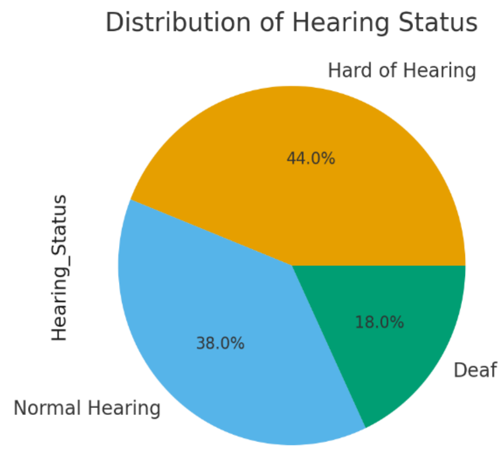

Figure 7.2: Pie Chart

IX. Limitations and Future Work

While SIGNEASE already performs well, future improvements will focus on making it more accurate, inclusive, and user-friendly. The system will be enhanced by expanding the dataset with more hand gestures and video-based variations to improve real-time recognition, even in challenging environments like low lighting or varying camera quality. Multilingual support will be introduced to make communication more accessible across regions, while better handling of similar signs will ensure smoother translations. Additional features such as emotion detection, integration with wearable devices, and cloud-based storage for personalized communication history will also be explored to provide a richer and more seamless user experience.

X. CONCLUSION

SIGNEASE is a mobile-based solution designed to bridge the communication gap between hearing and impaired individuals, enabling seamless interaction through real-time gesture recognition and conversion into text and audio. It offers an accessible, user-friendly platform that can be used by everyone, regardless of their hearing abilities, on their smartphones. The application is built with the aim of being widely available, making it easy for both deaf and hearing individuals to communicate effectively in their day-to-day interactions. In the future Moving forward, SIGNEASE plans to expand its dataset, incorporating videos alongside the existing 87,000-image dataset, and add multilingual support to cater to a broader audience. These advancements will improve the accuracy of sign language detection and enhance the system's ability to support diverse languages, making the app more inclusive and efficient in bridging communication gaps.

XI. REFERENCES

1. Zahid, H., Syed, S. A., Rashid, M., Hussain, S., Umer, A., Waheed, A., Nasim, S., Zareei, M., & Mansoor, N. (2023). A Computer Vision-Based System for Recognition and Classification of Urdu Sign Language Dataset for Differently Abled People Using Artificial Intelligence. Volume 2023, Article ID 1060135, 17 pages. Retrieved from

2. Naseem, M., Sarfraz, S., Abbas, A., & Haider, A. (2019). Developing a Prototype to Translate Pakistan Sign Language into Text and Speech While Using Convolutional Neural Networking. Journal of Educational Psychology, 10(15), 147. ISSN 2222-1735 (Paper), ISSN 2222-288X (Online). DOI: 10.7176/JEP.

3. Saleem, Muhammad Imran, Atif Siddiqui, Shaheena Noor, Miguel-Angel Luque-Nieto, and Enrique Nava-Baro. 2023. "A Machine Learning Based Full Duplex System Supporting Multiple Sign Languages for the Deaf and Mute" Applied Sciences 13, no. 5: 3114. https://doi.org/10.3390/app13053114.

4. Swathi, V., Reddy, C. G., Pravallika, N., Reddy, P. V., & Redd, Ch. P. (2021). Audio To Sign Language Converter Using Python. Volume 10, Issue 1, pp. 2621-2630.

5. Prof.Radha, S, Shirbhate, Mr. Vedant, D. Shinde, A Ms.Sanam, Metkari, Ms. Pooja, U. Borkar, A Ms.Mayuri and Khandge. “Sign language Recognition Using Machine Learning Algorithm.” (2020).

6. Khan, Imran & Rana, Sohail & Ansari, Nadia & Karachi, & Tariq, Talha & Ur, Maqsood & Awan, Rehman & Waqar, Adnan. (2022). DESIGN AND IMPLEMENTATION OF CNN FOR SIGN LANGUAGE RECOGNITION. Jilin Daxue Xuebao (Gongxueban)/Journal of Jilin University (Engineering and Technology Edition).

Vol: 41. 135-145. 10.17605/OSF.IO/NKHJT.

7. Z. Wang, H. Shao, T. Zhao, J. Ma, Q. Wang, H. Chen, K. Liu, and J. Ren, "Hear Sign Language: A Real-Time End-to-End Sign Language Recognition System," IEEE Transactions on Mobile Computing, vol. 21, no. 7, pp. 2398-2410, July 2022.

8. Adeyanju, O. O. Bello, and M. A. Adegboye, "Machine learning methods for sign language recognition: A critical review and analysis," Computers in Biology and Medicine, vol. 138, p. 104834, Dec. 2021, doi: 10.1016/j.compbiomed.2021.104834.

9. M. Raees, S. Ullah, S. U. Rahman, and I. Rabbi, "Image-based recognition of Pakistan Sign Language," Department of Computer Science and IT, University of Malakand, Pakistan.

10. Y. Obi, K. S. Claudio, V. M. Budiman, S. Achmad, and A. Kurniawan, "Sign language recognition system for communicating to people with disabilities," Procedia Computer Science, vol. 216, pp. 13–20, 2023

11. A.Sultan, W. Makram, M. Kayed, and A. A. Ali, "Sign language identification and recognition: A comparative study," Computer Science Review, vol. 22, no. 4, pp. 192-207, 2022.

12. M. A. A. Razak, S. Shahbuddin, F. Y. A. Rahman, Y. W. M. Yusof, R. Mohamad, and S. I. Suliman, "Hand Gesture Recognition based on Convolution Neural Network (CNN) and Support Vector Machine (SVM)," 2023 IEEE 14th Control and System Graduate Research Colloquium (ICSGRC), Shah Alam, Malaysia, 2023, pp. 1-6, doi: 10.1109/ICSGRC57744.2023.10215427.

13. K. Bantupalli and Y. Xie, "American Sign Language Recognition using Deep Learning and Computer Vision," in 2018 IEEE International Conference on Big Data (Big Data), Seattle, WA, USA, 2018, pp. 4896-4899, doi: 10.1109/BigData.2018.8622141.

14. S. He, "Research of a Sign Language Translation System Based on Deep Learning," in 2019 IEEE International Conference on Artificial Intelligence and Advanced Manufacturing (AIAM), 2019, pp. 392-396, doi: 10.1109/AIAM48774.2019.00083.

15. International Conference on Trendz in Information Sciences and Computing (TISC), 2012, pp. 30-35.

16. H. C. M. Herath, W. A. L. V. Kumari, W. A. P. B. Senevirathne, and M. Dissanayake, "Image Based Sign Language Recognition System for Sinhala Sign Language," 2013.

17. M. Geetha and U. C. Manjusha, "A Vision-Based Recognition of Indian Sign Language Alphabets and Numerals Using B-Spline Approximation," in International Journal on Computer Science and Engineering (IJCSE), vol. 4, no. 3, pp. 406-415, 2012.

18. L. Pigou, S. Dieleman, P. J. Kindermans, and B. Schrauwen, "Sign Language Recognition Using Convolutional Neural Networks," in Computer Vision - ECCV 2014 Workshops, Lecture Notes in Computer Science, vol. 8925, L. Agapito, M. Bronstein, and C. Rother, Eds., Springer, Cham, 2015, pp. 572-578, doi: 10.1007/978-3-319-16178-5_40.

19. S. Escalera et al., "ChaLearn Looking at People Challenge 2014: Dataset and Results," in Workshop at the European Conference on Computer Vision, Springer, Cham, 2014, pp. 459-473.

20. J. Huang, W. Zhou, and H. Li, "Sign Language Recognition using 3D Convolutional Neural Networks," in

IEEE International Conference on Multimedia and Expo (ICME), Turin, Italy, 2015, pp. 1-6, doi: 10.1109/ICME.2015.7177375.

21. Grassknoted. (n.d.). *ASL alphabet dataset*. Kaggle. Retrieved November 24, 2024

22. Ngrok. (n.d.). *Your app's front door: All-in-one API gateway, Kubernetes Ingress, DDoS protection, firewall, and global load balancing as a service*. Retrieved November 24, 2024, from

23. Databeru. (n.d.). *Recognition of ASL (Acc 99.66%)*. Kaggle. Retrieved November 24, 2024, from https://www.kaggle.com/code/databeru/recognition-of-asl-acc-99-66

24. Google. (n.d.). *Firebase console*. Retrieved November 24, 2024,

25. Z. R. Saeed, Z. B. Zainol, B. B. Zaidan, and A. H. Alamoodi, "A Systematic Review on Systems-Based Sensory Gloves for Sign Language Pattern Recognition: An Update From 2017 to 2022," *IEEE Access*, vol. 10, pp. 9938436, 2022.

26. M. Imran Saleem, A. A. Siddiqui, S. Noor, and M. Á. Luque-Nieto, "A Novel Machine Learning-Based Two-Way Communication System for Deaf and Mute," Applied Sciences, vol. 13, no. 453, 2022. [CrossRef]

27. M. I. Saleem, P. Otero, S. Noor, and R. Aftab, "Full Duplex Smart System for Deaf Dumb and Normal People," in Proceedings of the 2020 Global Conference on Wireless and Optical Technologies (GCWOT), Malaga, Spain, 6–8 October 2020. [CrossRef]

28. A.Sood and A. Mishra, "AAWAAZ: A communication system for deaf and dumb," in Proceedings of the 2016 5th International Conference on Reliability, Infocom Technologies and Optimization (ICRITO): Trends and Future Directions, Noida, India, 7–9 September 2016, pp. 620–624. [CrossRef]

29. L. Boppana, R. Ahamed, H. Rane, and R. K. Kodali, "Assistive sign language converter for deaf and dumb," in Proceedings of the 2019 IEEE International Congress on Cybermatics: 2019 International Conference on Internet of Things (iThings), GreenCom, CPSCom, and SmartData, Atlanta, GA, USA, 14–17 July 2019, pp. 302–307. [CrossRef]

30. S. Ameur, A. Ben Khalifa, and M. S. Bouhlel, "A novel hybrid bidirectional unidirectional LSTM network for dynamic hand gesture recognition with Leap Motion," Entertainment Computing, vol. 35, 2020, Art. no. 100373. [CrossRef]

31. S. Ameur, A. Ben Khalifa, and M. S. Bouhlel, "Chronological pattern indexing: An efficient feature extraction method for hand gesture recognition with Leap Motion," Journal of Visual Communication and Image Representation, vol. 70, 2020, Art. no. 102842. [CrossRef]

32. R. A. Suharjito, F. Wiryana, M. C. Ariesta, and G. P. Kusuma, "Sign Language Recognition Application Systems for Deaf-Mute People: A Review Based on Input-Process-Output," Procedia Computer Science, vol. 116, pp. 441–448, 2017. [CrossRef]

33. S. Deb and P. Bhattacharya, "Augmented Sign Language Modeling (ASLM) with interaction design on smartphone—An assistive learning and communication tool for inclusive classroom," Procedia Computer Science, vol. 125, pp. 492–500, 2018. [CrossRef]

34. R. S. Bhadauria, S. Nair, and D. K. Pal, "A survey of deaf-mutes," Medical Journal Armed Forces India, vol. 63, pp. 29–32, 2007. [CrossRef]

35. S. Patwary, Z. Zaohar, A. A. Sornaly, and R. Khan, "Speaking System for Deaf and Mute People with Flex Sensors," in Proceedings of the 2022 6th International Conference on Trends in Electronics and Informatics (ICOEI), Tirunelveli, India, 28–30 April 2022, pp. 168–173. [CrossRef]