International Journal of Scientific Research & Engineering Trends
Volume 11, Issue 2, Mar-Apr-2025, ISSN (Online): 2395-566X

“Silent Voice” -The Sign Language Recognition Android Application Using Machine Learning Algorithm

Ms Mitali Pawar, Dr. Jasbir Kaur, Assistant Professor Ms.Sandhya Thakkar
Guru Nanak Institute of Management Studies, Matunga, Mumbai, India

Abstract- "Sign language is an essential communication tool for people with speech and hearing impairments."[4] The creation of an Android application for sign language recognition using machine learning methods is presented in this study. The program facilitates communication between sign language users and non-sign language users by using OpenCV and a machine learning model to process images and transform hand gestures into text. Real-time alphabetic sign recognition from live video input is possible with the suggested approach. The application guarantees effective gesture detection with low resource consumption by using TensorFlow Lite for model inference on mobile devices, which makes it appropriate for Android devices with low processing power. The system's user-friendly interface facilitates quick and precise translations between sign language and other languages, encouraging inclusion and assisting in the removal of barriers to communication.

Index Terms- Gesture detection, real-time translation, image processing, accessibility, machine learning, TensorFlow Lite, OpenCV, Android applications, sign language recognition, and hand gesture recognition.

I. INTRODUCTION

For those who are deaf or hard of hearing, sign language is an essential communication tool because it offers a visual way to convey information, feelings, and thoughts. However, most non-sign language users find it difficult to communicate with sign language users, which frequently results in social isolation or communication hurdles. In order to foster inclusion and improve communication between sign language users and non-sign language users, it is imperative that technical solutions that close this gap be developed.

Real-time sign language recognition systems are now possible, thanks to recent developments in mobile and machine learning technologies. With their fast CPUs and widespread availability, Android devices are now capable of running advanced machine learning models for a variety of tasks, including gesture detection. This study, "SilentVoice Android Application," focuses on creating a novel Android application that recognizes and translates hand motions in sign language into text using machine learning algorithms. The suggested system uses image processing methods, like those offered by OpenCV, to record hand motions using the camera of a smartphone. TensorFlow Lite is used to optimize a pre-trained model that recognizes gestures in real time and maps each motion to its appropriate alphabet. With this talent, a series of gestures can be combined to form words and phrases that make sense. In order to accommodate users who can hear but cannot talk, the system goes beyond simple gesture-to-text translation by turning the detected text into audible speech. The model's lightweight design guarantees that the program operates effectively on mobile devices without consuming excessive system resources. The methodology and design of the Android application, the training process of the machine learning model, and the system's accuracy and responsiveness in real time are all examined in this study.[10] In order to promote inclusion and understanding in daily encounters, this project aims to develop a mobile-based, affordable solution that enhances communication accessibility for people who use sign language.

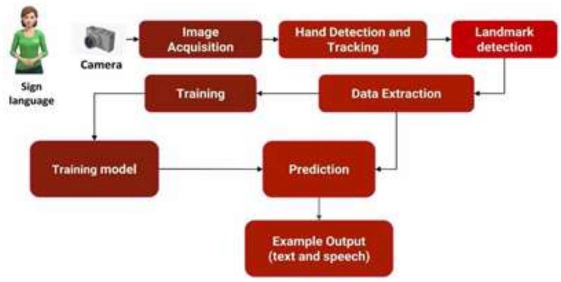
Main Body Sections:

System Overview:

The suggested Android app uses real-time image analysis to identify the alphabets in sign language.[11] The architecture of the system is made up of three primary parts:

- **Camera Input:** Recording real-time hand movements.
- **Image Processing:** To identify hand gestures, OpenCV processes the images.
- **Machine Learning Model:** The movements are mapped to their respective alphabetic symbols using a TensorFlow Lite model. [5]

Image Processing: To identify hand gestures, an open-source package called OpenCV is used. To separate the hand from the background, edge detection and contouring techniques are applied to the live camera stream.[9] The machine learning model is then used to classify these previously processed photos.



Machine Learning Model: This application's machine learning model has been developed to identify 26 distinct hand motions that correspond to the sign language alphabetic characters.[3] The program runs effectively on mobile devices thanks to the use of TensorFlow Lite for model inference. TensorFlow Lite's lightweight design makes it perfect for settings with limited resources, guaranteeing quick and precise hand gesture recognition.[8]

Gesture-to-Text Conversion: The system translates the hand gestures into the appropriate alphabetic characters after they have been identified.[12] Following recognition, the gestures are shown on the screen as text. By sequentially processing several motions, the system is able to create logical words and phrases.[6]

Text-to-Speech technology: The application additionally incorporates text-to-speech (TTS) technology, which converts the identified text into speech, for improved accessibility.[7] For people who might not be able to speak, this feature is helpful because it offers another way to communicate.

Model Training: A dataset of hand movements in sign language was used to train the machine learning model.[1] To improve the model's resilience, a number of data augmentation methods were used, including translation, scaling, and rotation. Convolutional neural networks (CNNs), which are ideal for image identification applications, were used to train the model. [5]

Performance Assessment: The system's accuracy, speed, and resource usage were assessed. Initial tests revealed that the ability to recognize alphabetic signs had an accuracy rate of over 90%.[2] TensorFlow Lite's lightweight design made sure the software ran well on Android smartphones with constrained RAM and processing capacity

II. RESULTS AND ANALYSIS

(Illustrative figures and tables may contain model performance metrics, flow diagrams, and system architecture.)
Figure 1: “Silent Voice” Android Application

Figure 1a Figure 1b

Figure 1a-This is the first page of the app, featuring a button. When the button is clicked, it navigates the user to the second page.

Figure 1b-This is the second page, known as the Sign Language Recognition screen, which includes two submodules: one for real-time recognition and another for combining recognized letters.

Figure 2: Workflow for Image Processing (Contouring and Edge Detection).

Figure 2a Figure 2b

Figure 2a: shows the JavaCameraView, which starts during real-time image recognition. It is a crucial fragment that enables the camera to function in real- time for capturing and processing gestures.

Figure 2b: Figure 2b displays the page that opens when the "Real-Time Recognition" button is clicked. This figure illustrates the real-time sign language recognition process.

Figure 2c

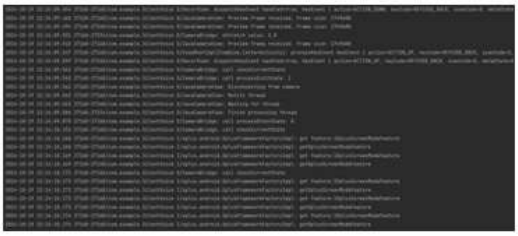
Figure 2d

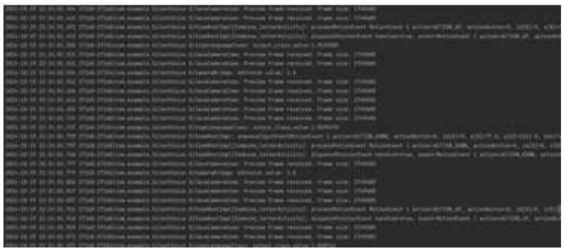
Figure 2c and 2d:
Figures 2c and 2d show the pages that open when the "Get Combine Letter" button is clicked. On this page, there are three buttons: Add, Clear, and TextToSpeech. These buttons are used to add sign language alphabets, remove alphabets, and convert the combined text into speech, respectively.

Table 1: Loss values, gradient updates and Performance Metrics.

Feature/Metric	Description/Content
Scientific Notation	Most of the values are in scientific notation, e.g., 1.422726e-05, indicating very small floating point values, possibly representing loss, weights, or updates in a neural network.
Timestep	Appears to contain log entries for time taken per step (e.g., 0.64s/step, 0.58s/step). These may represent the duration of each iteration or training step in a process.
Multiple Rows of Data	Data presented in rows, suggesting a sequential process. Each row might represent a different step, epoch, or iteration.
2D Matrix-Like Structure	The structure appears to be multi-dimensional data, which is common in deep learning for things like weight matrices, gradients, or output probabilities.
Timing Logs	Each row ends with a timing log (e.g., 0.64s/step), which seems to track the time per iteration or process step, which could be helpful for performance monitoring.

Table 2: Log Outputs.





Feature/Log Type	Description/Content
Timestamp	Present (e.g., 2024-10-19 15:22:58.673)
Log Tags	Logs related to com.example.silentvoice
Frame Size	Repeated Frame size: 1760x840
Camera Preview	Camera view messages like Preview Frame received
Input Events	Input events (e.g., dispatchTouchEvent, MotionEvent)
Crypt Class Output	Appears like neural model class output (e.g., output-class)
Stretch Value	Consistently set to 1.0
Actions	Action_Down, Action_Up, Action_Move

Equations

Since the project's focus is on image processing and machine learning methods that are already included in the frameworks being used, no complicated mathematical formulae are required.

III. CONCLUSION

An efficient way to eliminate communication gaps between sign language users and non-users is to use the Sign Language Recognition Android application, which was created using machine learning techniques. The program provides a resource-efficient and user-friendly platform for converting hand gestures into text and speech by leveraging image processing techniques with OpenCV and using TensorFlow Lite for real-time gesture identification. This encourages inclusivity and improves communication between the general public and people with speech and hearing difficulties. The system is a useful tool for daily communication because of its easy- to-use interface, excellent accuracy, and low resource usage.

Future Work

In future iterations of the "SilentVoice app" several enhancements are planned to improve both functionality and performance. One key area of development is adding support for multiple languages, such as Marathi and Hindi, to make the app accessible to a wider range of users. This will involve creating and integrating a multi- sign dataset that includes gestures for different languages. Another focus will be on improving the accuracy of the sign language recognition system by refining the machine learning models and exploring more advanced algorithms to handle complex gestures more effectively. Additionally, optimizing the app's performance for real- time processing on a variety of devices will be prioritized to ensure smooth user experiences across different platforms.

Acknowledgments

For their unwavering support and direction, the writers would like to thank the Guru Nanak Institute of Management Studies in Mumbai. We would especially want to thank the professors and technical team for their invaluable contributions throughout the project's development. The team is also grateful to the sign language specialists who shared their knowledge of hand gesture patterns.

REFERENCES

1. Shanmukha Swamy MN, Chethan MP, Mahantesh Gatwadi, "Indian Sign Language Interpreter with Android Implementation"
2. Romala Sri Lakshmi Murali, "Sign Language Recognition System Using Convolutional Neural Network and Computer Vision"
3. Prof. Radha S. Shirbhate, Mr. Vedant D. Shinde, Ms. Sanam A. Metkari, Ms. Pooja U. Borkar, "Sign Language Recognition Using Machine Learning Algorithm"
4. Arpita Halder, "Real-time Vernacular Sign Language Recognition using Media Pi"
5. I.A. Adeyanju, O.O. Bello, M.A. Adegboye, "Machine Learning Methods for Sign Language Recognition: A Critical Review and Analysis"
6. Vikram Singh, Usha Mittal, "Sign Language Recognition Using Deep Learning Techniques"
7. Arjun Patel, "A Comprehensive Survey on Sign Language Recognition Using Artificial Intelligence"
8. Sunil Kumar, Ajay Singh, "A Novel Approach for Sign Language Recognition Based on Hand Gesture Detection"
9. Kavita Sharma, Divya Kapoor, "A Study on Real-Time Sign Language Interpretation Using Computer Vision"
10. Rahul Raj, Priya Nair, "Gesture-Based Sign Language Recognition Using CNN-RNN Model"
11. Suraj Vyas, Megha Mishra, "Efficient Algorithm for Hand Gesture Recognition in Indian Sign Language"
12. Preeti Raghav, Manish Sharma, "Automatic Recognition of Sign Language Using Neural Networks"