SIGN LANGUAGE CONVERSION INTO TEXT

Mini-Project Report submitted to the SASTRA Deemed to be University in partial fulfillment of the requirements for the award of the degree of

B. Tech. ELECTRONICS & COMMUNICATION ENGINEERING

Submitted by

KARTHICK RAJA B - 126004121
RAMANA GUHAN GR - 126004218
AVINASH M - 126004041

April 2025

SCHOOL OF ELECTRICAL & ELECTRONICS ENGINEERING

THANJAVUR, TAMIL NADU, INDIA – 613 401

SCHOOL OF ELECTRICAL & ELECTRONICS ENGINEERING
THANJAVUR, TAMIL NADU, INDIA – 613 401

Bonafide Certificate

This is to certify that the report titled "**Sign Language Conversion Into Text**" submitted in partial fulfillment of the requirements for the award of the degree of B. Tech. Electrical & Electronics Engineering to the SASTRA Deemed to be University, is a bona-fide record of the work done by

Karthick Raja B (126004121)

Ramana Guhan G R (126004218)

Avinash M (126004041)

during the Sixth semester of the academic year 2024-25, in the School of Electrical & Electronics Engineering, under my supervision.

Signature of the Project Supervisor:

Name with Affiliation : Dr. RAJU N, Sr.Asst.Professor, ECE, SEEE

Date : 30/04/2025

Project *Vivavoce* held on :

Examiner 1 **Examiner 2**

SCHOOL OF ELECTRICAL & ELECTRONICS ENGINEERING
THANJAVUR, TAMIL NADU, INDIA – 613 401

Declaration

We declare that the report titled "**Sign Language Conversion into Text**" submitted by me/us is an original work done by me/us under the guidance of **Dr. Raju N, Assistant Professor, School of Electrical and Electronics Engineering, SASTRA Deemed to be University** during the sixth semester of the academic year 2024-25, in the **School of Electrical and Electronics Engineering**. The work is original and wherever I/We have used materials from other sources, I/We have given due credit and cited them in the text of the report. This report has not formed the basis for the award of any degree, diploma, associate-ship, fellowship, or other similar title to any candidate of any University.

Signature of the Candidate(s) :

Name of the Candidate(s) : KARTHICK RAJA B - 126004121
RAMANA GUHAN G R - 126004218
AVINASH M - 126004041

Date : 30/04/2025

Acknowledgements

First of all, we express our gratitude to **Dr. S. VAIDHYASUBRAMANIAM**, Vice Chancellor, SASTRA Deemed to be University for providing us with all facilities and an encouraging platform in the course of study.

We sincerely thank **Dr. R. CHANDRAMOULI**, Registrar, SASTRA Deemed to be University, for providing the opportunity to carry out this project.

We convey our warm gratitude to **Dr. K. THENMOZHI**, Dean, SEEE, .
Dr. K. SRIDHAR, Dean Research, ECE, **Dr. A. KRISHNAMOORTHY**, Associate Dean – Academics, SEEE SASTRA Deemed to be University for introducing mini projects in our curriculum.

We owe a debt of earnest gratitude towards our guide **Dr. RAJU N**, Assistant Professor, ECE, SEEE, SASTRA Deemed to be University for her continued support and guidance throughout the course of our work.

We would especially like to thank the panellists **Dr. RAMYA VIJAY** and **Dr. PADMAPRIYA PRAVINKUMAR**, Department of ECE, SASTRA deemed to be University for helping us in shaping this project worthwhile.

We thank all our professors and assistant professors who have directly or indirectly assisted for our project. We also want to thank the research scholars, staff members in the office and laboratory for their full cooperation in doing our project.

Abstract

We developed a real-time computer vision application for American Sign Language (ASL) recognition using OpenCV, Mediapipe, TensorFlow, and Python 3.6.6. The system captures live hand gesture inputs through a webcam and processes them with efficient landmark detection. Mediapipe's hand tracking module extracts key hand features, while a TensorFlow-based model classifies the gestures. The recognized ASL signs are instantly translated and displayed as text on the screen. This seamless gesture-to-text conversion enables clear communication for individuals relying on sign language. The application ensures high accuracy, low latency, and ease of use. It can be adapted for educational, assistive, and communication tools. Overall, the project demonstrates the potential of integrating AI and computer vision to bridge communication gaps.

Specific Contribution :

- **KARTHICK RAJA B :** I have worked with image processing for the images/data collected. In this process we convert the webcam images into "ImageWhite" format. And cropping the hand area from the webcam feed.
- **RAMANA GUHAN G R :** I have worked with the creation of datasets for all the 26 alphabetical letters by capturing more than 500 samples of each letter using the webcam module.
- **AVINASH M :** I have worked with the python codes and also trained the model to test the inputs using machine learning algorithms to carry out the mapping of gestures.

Specific Learning :

- **KARTHICK RAJA B :** I Learned about the application of preprocessing techniques like background removal, image augmentation and normalization.
- **RAMANA GUHAN G R** : I Learned how to capture real-time video frames using OpenCV . Learned to create a custom dataset by collecting thousands of labeled hand gesture images using a webcam module.
- **AVINASH M** : Learned how to load custom models into Python projects for inference. Improved problem-solving skills during debugging.

Table of Contents

TABLE OF FIGURES

ABBREVIATIONS

- ASL - American Sign Language
- OpenCV - Open source Computer Vision
- AI - Artificial Intelligence
- CNN - Convolutional Neural Network
- FRCNN - Faster Convolutional Neural Network
- YOLO - You Only Look Once
- MATLAB - Matrix Laboratory
- ML - Machine Learning
- TFLite - TensorFlow Lite
- ROI - Region Of Interest
- NLP - Natural Language Processing
- TTS - Text To Speech
- ISL - Indian Sign Language
- SLR - Sign Language Recognition
- IDE - Integrated Development Environment

CHAPTER 1

INTRODUCTION

The Sign Language to Text Conversion project focuses on bridging communication gaps for people with hearing or speech impairments. This project captures hand gestures via a webcam and converts them into readable text in real-time. Using OpenCV, live video streams are processed to detect and segment the hand region.

MediaPipe is employed for accurate hand landmark detection, identifying key points such as fingertips and joints. Images of various gestures are collected and preprocessed by resizing, normalizing, and augmenting them to build a robust dataset.

To simplify and speed up model training, Google's Teachable Machine platform is used, allowing easy training of a Convolutional Neural Network (CNN) model. This model learns to classify different hand signs based on the features extracted from the images.

The trained model is then integrated into the project using TensorFlow and Keras libraries. During real-time operation, webcam frames are passed through the model to predict the corresponding text label for each gesture.

Challenges such as background noise, lighting variations, and hand orientation were tackled through preprocessing and data augmentation techniques. Performance improvements were achieved by optimizing frame rates and fine-tuning model thresholds.

This project demonstrates the practical application of machine learning, computer vision, and deep learning in creating assistive technologies. It highlights the power of integrating AI tools with accessible platforms for social impact.

In future expansions, the system could be extended to recognize dynamic gestures (full sentences) or integrate speech synthesis for more complete communication support.

1.1. LITERATURE SURVEY

In[1], This paper presents a system to convert Indian Sign Language (ISL) hand gestures into text using image processing techniques. It uses a webcam to capture hand gestures and processes them with preprocessing, segmentation, and morphological filtering. Important features like Eigenvalues and Eigenvectors are extracted for gesture recognition. The Linear Discriminant Analysis (LDA) algorithm is used for classifying the signs, helping reduce dimensionality and improve recognition accuracy. The system recognizes 26 static ISL gestures and converts them into text and voice. It mainly uses MATLAB for implementation. Overall, it aims to help hearing-impaired people communicate easily with non-signers.

In[2], This paper presents a real-time system for converting sign language into text and speech, and vice-versa. It uses Convolutional Neural Networks (CNN) to recognize gestures from video input and image processing to clean and enhance the data. Natural Language Processing (NLP) is used to translate recognized signs into readable text, which is then converted to speech using Text-to-Speech (TTS) technology. A unique part of the system is animation gesture recognition, helping better capture hand movements. The system ensures fast communication with minimal delay. This project aims to improve communication for the hearing-impaired and promote inclusivity.

In[3], This paper presents a system to convert Indian Sign Language (ISL) gestures into text and speech using deep learning techniques. It explores different models like CNN, Faster R-CNN (FRCNN), YOLO, and MediaPipe for recognizing hand gestures from live webcam input. MediaPipe was found best for real-time performance with high accuracy. The project captures hand movements, preprocesses images, extracts features, classifies gestures, and finally converts the recognized gesture into both text and voice output. Other models (FRCNN, YOLO) performed well but had issues with speed or live input. This system aims to help deaf-mute people communicate easily and plans future improvements using better cameras and server-based systems.

1.2. MOTIVATION

1. Importance of Communication

Communication is essential for building relationships and sharing ideas. However, individuals with hearing and speech impairments often face major challenges in communicating with the general public.This project addresses that communication gap.

2. Problem Statement

Even though sign language is widely used among deaf and mute communities, most people do not understand it. This lack of understanding leads to communication barriers and social isolation.

3. Purpose of the Project

This project aims to create a real-time system that can convert sign language gestures into readable text and audible speech. By doing so, we hope to make communication easier and more natural for hearing-impaired individuals.

4. Technology Integration

We utilize technologies like OpenCV, MediaPipe, Convolutional Neural Networks (CNN),TensorFlow, Keras, Numpy and deep learning models to detect, process, and translate hand gestures accurately and efficiently.

5. Impact and Vision

This system promotes social inclusivity and accessibility. Through this work, we aim to promote inclusivity, bridge communication gaps, and showcase how artificial intelligence and computer vision can be applied to solve real-world problems that improve lives. We used American sign language to convert the sign language into text.

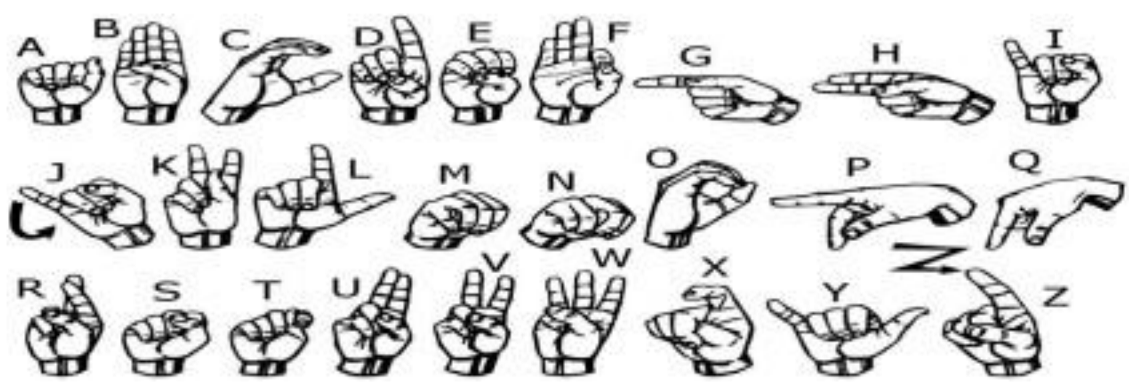


Fig 1.2. American Sign Language (ASL)

CHAPTER 2

METHODOLOGY

Communication is fundamental to human interaction, but millions of deaf and hard-of-hearing individuals face daily barriers. Sign language is their primary mode of communication, but not everyone understands it. Converting sign language into text bridges this gap, enabling smoother communication between the hearing and non-hearing communities. It promotes inclusivity in education, workplaces, healthcare, and public services. Real-time conversion helps deaf individuals express themselves in environments where interpreters are not available. It also fosters independence and confidence in social and professional settings. For businesses, it enhances customer service accessibility. In education, it supports better classroom integration. In emergency situations, it can save lives by ensuring critical information reaches everyone. Technological advances using OpenCV, Mediapipe, and AI models have made real-time conversion practical and efficient. This innovation aligns with global goals of equality and accessibility. Ultimately, sign language to text conversion is a vital step towards a more inclusive, connected world.

2.1. PROPOSED SYSTEM

The user performs a specific hand gesture representing a letter, word, or phrase.A camera captures live video frames containing the hand gesture. Each frame from the video feed is treated as an input image for further processing. The captured image is cleaned by removing unnecessary background elements, resizing, and normalizing the image to highlight only the hand gesture.

The resulting image focuses mainly on the hand, ensuring better clarity for gesture recognition. Key features such as hand shape, position, and movement are extracted. These features typically include points or patterns that represent the structure of the hand gesture. The extracted information is structured into a format that can be easily analyzed for gesture recognition.

The structured features are compared with trained models or predefined datasets to identify the most probable gesture performed. The recognized gesture is then converted into corresponding text output, which can be displayed for easy communication.

Entire project development and debugging are done on Pycharm.

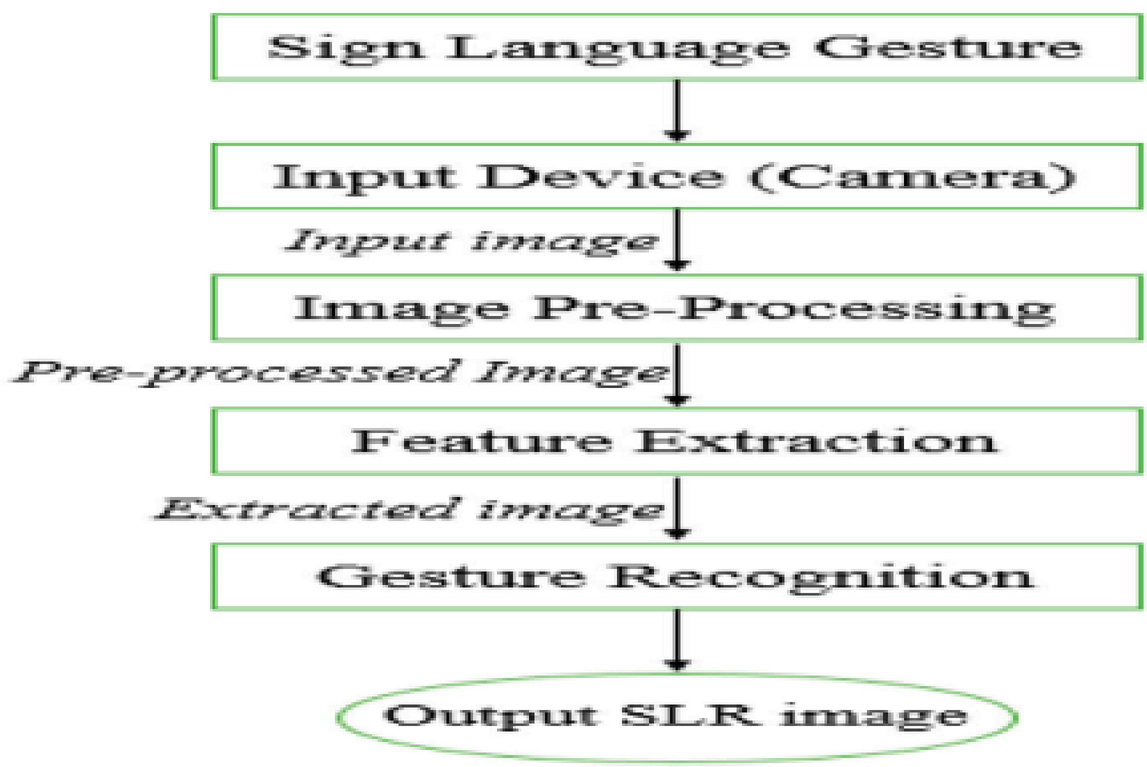


Fig 2.1. Flow Chart of conversion of sign language into text

`

2.2 SOFTWARE/LIBRARY FILES DESCRIPTION

2.2.1. PYCHARM

Fig 2.2. Pycharm

PyCharm serves as the IDE for writing, organizing, and debugging the project code efficiently. It helps manage multiple files like gesture detection scripts, model training, and real-time recognition modules. Smart code suggestions, error highlighting, and an integrated terminal make development faster and smoother. PyCharm also simplifies handling external libraries through virtual environments. Its powerful debugging tools help in troubleshooting and speeding up the final deployment.

2.2.2. MEDIAPIPE

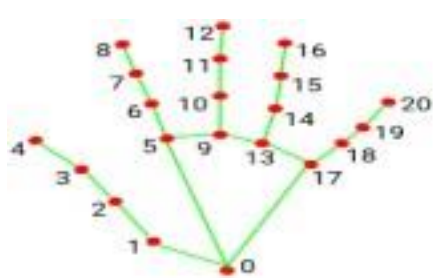


Fig 2.3. Mediapipe

Mediapipe is used to detect and track the hand from the live camera feed. It identifies 21 key landmarks on the hand, such as fingertips and joints. This helps in accurately capturing the hand's shape, movement, and position. The extracted landmark data is structured for easy input into the recognition model. Finally, the model uses this data to classify and convert the hand sign into text.

2.2.3 OpenCV

Fig 2.4 OpenCV

OpenCV is used to access the camera and capture real-time video frames. It handles image processing tasks like resizing, color conversion, and frame extraction. OpenCV helps display the live video feed with hand landmarks drawn on it. It is also used to overlay the recognized text onto the video output. Overall, OpenCV manages all image-related operations needed for smooth real-time detection.

2.2.4. Numpy

Fig 2.5. NumPy

NumPy is used to handle and process the numerical data of hand landmarks. It helps in storing, manipulating, and organizing the (x, y, z) coordinate values efficiently. Normalization of landmark values is done using NumPy to improve model accuracy. NumPy arrays make it easy to feed the structured data into the machine learning model. Overall, it ensures faster mathematical operations and better data management in the project.

2.2.5. TENSORFLOW / KERAS

Fig 2.6. TensorFlow / Keras

TensorFlow/Keras is used to design, train, and build the machine learning model for gesture recognition. The model learns patterns from hand landmark data and maps them to corresponding text outputs. Keras simplifies building the model architecture with layers like Dense and Dropout. TensorFlow handles the training process, optimization, and evaluation of the model. Finally, the trained model is used for real-time prediction of sign language gestures.

2.2.6 TEACHABLE MACHINE

Fig 2.7. Teachable Machine

Teachable Machine is used to quickly train a gesture recognition model without complex coding. It allows uploading images or using webcam inputs to create a dataset of hand signs. The platform automatically trains a machine learning model based on the provided examples. After training, the model can be exported for integration into the main project. It helps speed up the model development process, especially for quick testing and deployment.

CHAPTER 3

RESULTS

The step by step process involved and their results are shown below

3.1 DATA ACQUISITION

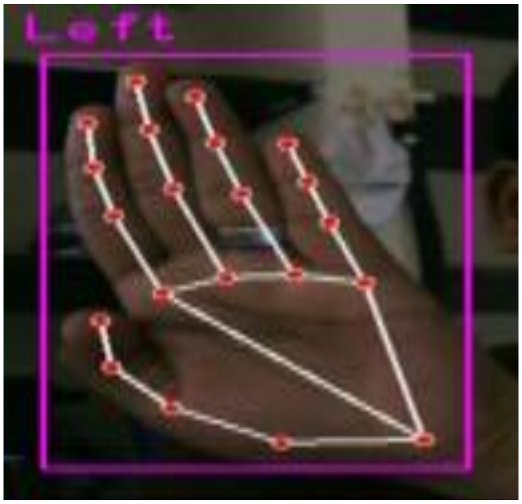


Fig 3.1. Left Hand Image

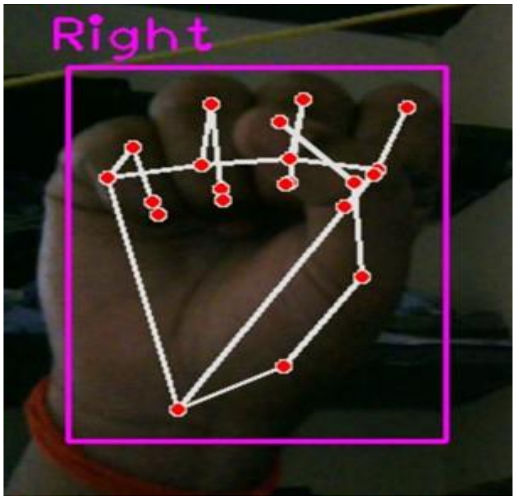


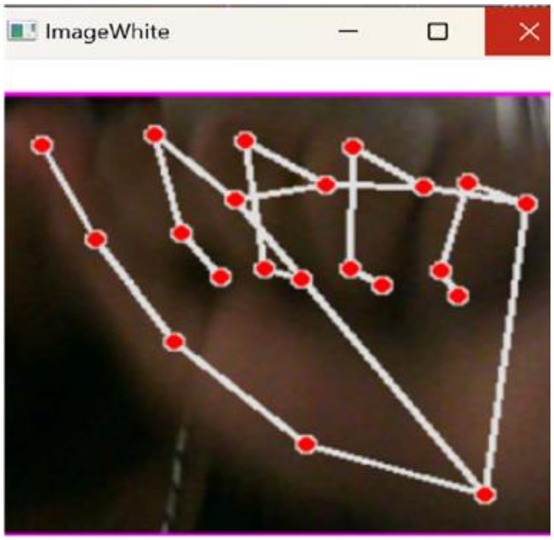
Fig 3.2. Right Hand Image

The first step involves capturing real-time video input from a webcam. The user performs different sign language gestures in front of the camera. Each gesture is shown clearly, one at a time, to ensure accurate capture. Frames are extracted from the video feed and stored as individual images.

Consistent lighting and background are maintained for better clarity. Images are saved in a folder structure for further use in training. This step ensures a variety of gestures are collected for recognition. It also includes capturing multiple images per gesture to handle variations.

Proper labeling or categorization is not yet applied here—raw data is gathered. This forms the base for building a reliable sign language dataset.

3.2 PREPROCESSING



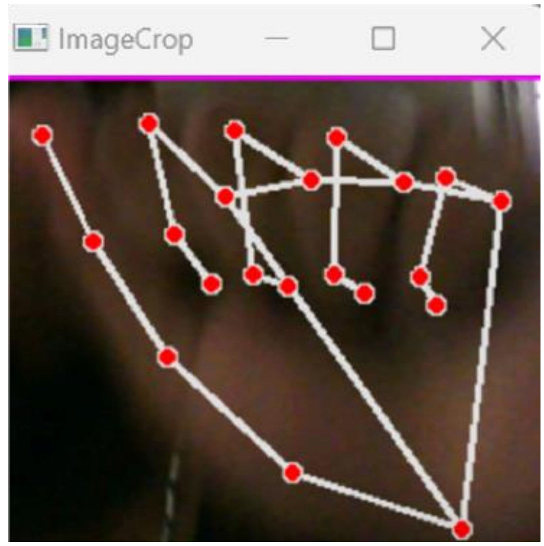


Fig 3.3. ImageWhite

Fig 3.4. ImageCrop

Preprocessing cleans and standardizes the raw images. First, the hand region is cropped from each image (ImageCrop). Cropping removes unnecessary background and focuses on the gesture.

Next, the cropped image is placed onto a fixed white background (ImageWhite). This ensures all images have the same size and centered hand positions. The process helps in reducing inconsistencies caused by background noise.

Uniformity in input images makes model training more accurate. It also handles different image aspect ratios by padding with white space. The result is a clean, square image with only the hand clearly visible. These processed images are saved for use in dataset creation.

3.3 FEATURE EXTRACTION

Fig 3.5. Dataset Creation

After preprocessing, the images are sorted based on gesture type. Each gesture is stored in a separate folder with its label name. The structure creates a clear, labeled dataset for training. Feature extraction involves identifying unique patterns in the hand's shape. This could include angles, distances, or the arrangement of fingers. In this project, images themselves are used as features via the Teachable Machine. Multiple samples per gesture improve accuracy and reduce overfitting. Images are uploaded in bulk to prepare for the model training stage. This dataset serves as the learning base for the machine learning model. Properly curated data is critical for accurate gesture recognition.

3.4 GESTURE RECOGNITION

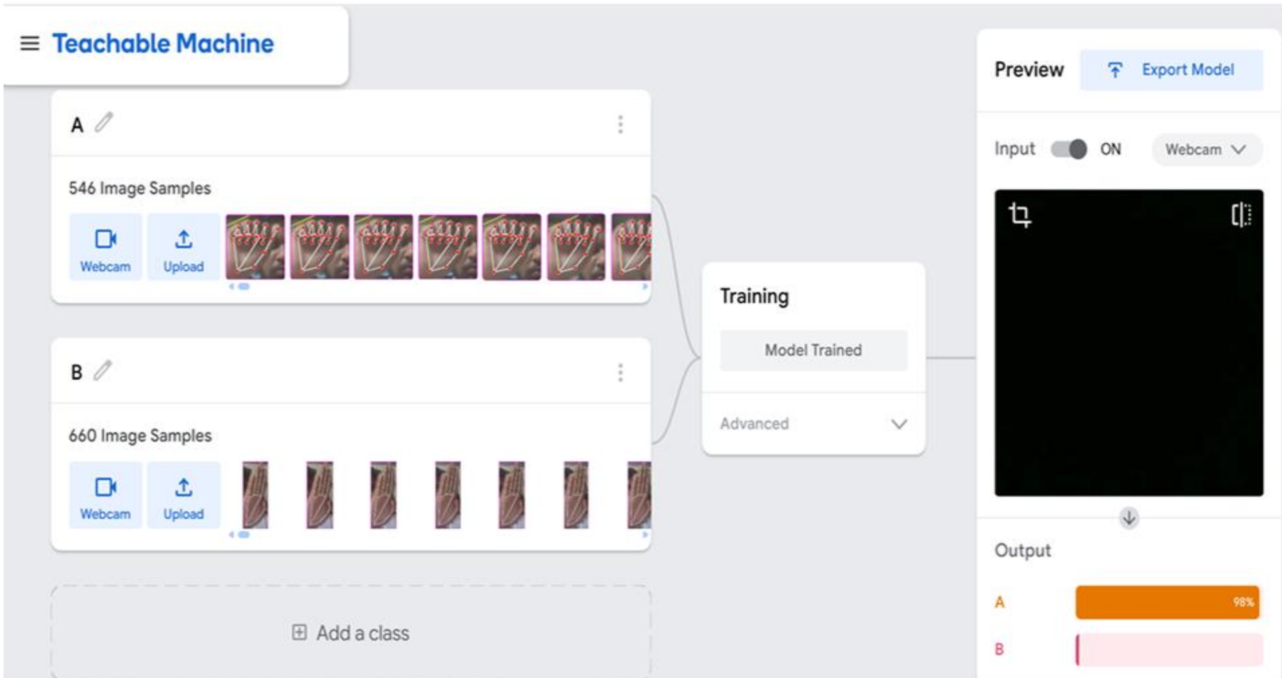


Fig 3.6. Gesture Recognition

The labeled dataset is uploaded to Teachable Machine's interface.A new image classification project is created for training the model. The model learns to differentiate gestures based on image patterns. No coding is required—training happens with a few clicks.

Once trained, the model can predict gestures from new images. The model is then exported in a format suitable for integration (like TensorFlow.js). It is integrated into the main application to classify gestures in real time. When a gesture is shown, the model processes it and returns the corresponding label. This label is then displayed as text output on the screen.

The model can recognize and convert signs to text instantly and accurately.

3.5 MAPPING

Each recognized hand gesture or sign is converted into a letter and displayed on the screen.

Desired Outputs of some of the matched gesture is given below

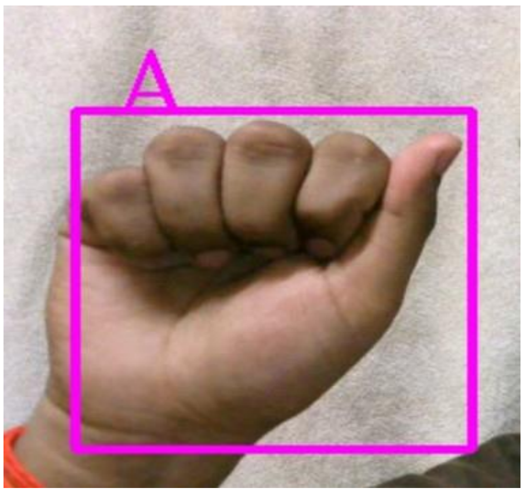


Fig 3.7. A

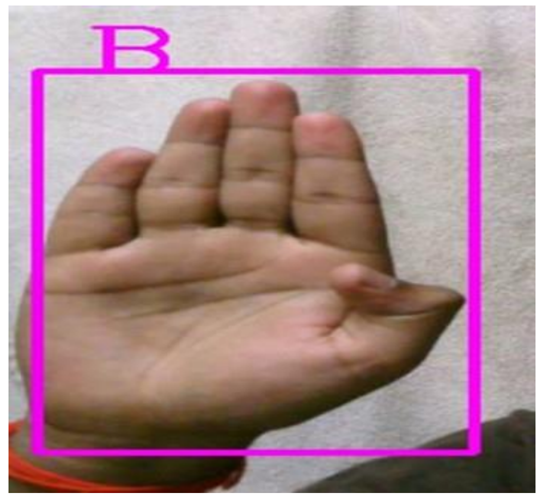


Fig 3.8. B

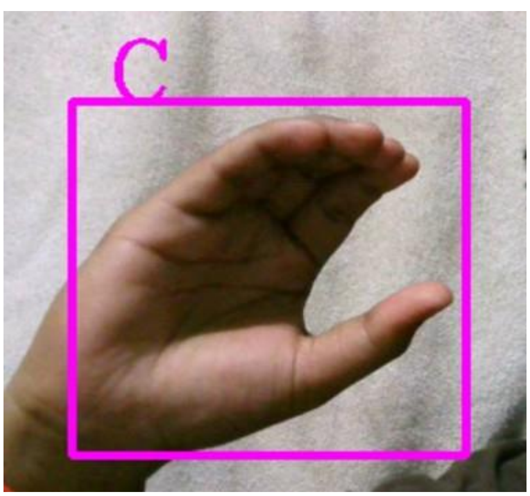


Fig 3.9. C

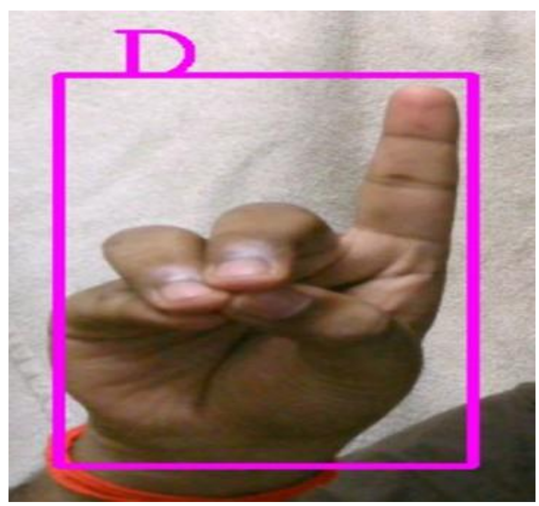


Fig 3.10. D

CHAPTER 4

CONCLUSION

This project successfully demonstrates the conversion of sign language gestures into readable text using computer vision and machine learning. By using a webcam to capture real-time gestures, the system enables hands-free, natural communication. The preprocessing techniques, including cropping and standardizing images, help in maintaining input consistency. The trained model performs gesture recognition with minimal latency and high accuracy.

This real-time conversion into text helps enhance accessibility in public spaces, education, healthcare, and daily interactions. The project encourages inclusivity and supports equal participation of the deaf and mute community in social environments. It also showcases how AI and computer vision can be used meaningfully in solving real-world social problems. The modular design of the project allows easy updates and improvements, such as adding new gestures or words.

The system is lightweight and can be deployed on basic hardware, making it scalable for broader usage. With continued improvements, this project can evolve into a complete assistive communication tool. The use of open-source tools keeps the project cost-effective and customizable.

Real-time performance and accuracy show that the system is suitable for practical applications. Overall, this project proves that technology can be a powerful ally in promoting inclusivity and accessibility.

CHAPTER 5

REFERENCES

[1] R. Kumar, A. Bajpai, and A. Sinha, "Mediapipe and CNNs for Real-Time ASL Gesture Recognition," *arXiv preprint arXiv:2305.05296*, May 2023.

[2] G. T. Sahoo, G. S. Mishra, and K. K. Ravulakollu, "Sign Language Recognition: State of the Art," *International Journal of Computer Applications*, vol. 75, no. 13, pp. 1–9, Aug. 2013.

[3] K. Bantupalli and Y. Xie, "American Sign Language Recognition Using Deep Learning and Computer Vision," in *Proceedings of the 2018 IEEE International Conference on Big Data (Big Data)*, Seattle, WA, USA, Dec. 2018, pp. 4896–4899.

[4] S. He, "Research of a Sign Language Translation System Based on Deep Learning," in *Proceedings of the 2020 International Conference on Artificial Intelligence and Computer Engineering (ICAICE)*, Beijing, China, Oct. 2020, pp. 1–5.

[5] K. Antupalli and Y. Xie, "Sign Language Classification Using Deep Learning Convolution Neural Network," *Journal of The Institution of Engineers (India): Series B*, vol. 104, pp. 1–9, 2023.

CHAPTER 6

APPENDIX

6.1 SIMILARITY CHECK