International Journal of Multidisciplinary Engineering in Current Research - IJMEC
Volume 10, Issue 6, June-2025, http://ijmec.com/, ISSN: 2456-4265

Hand Sign Detection (ASL) Using AI and Image Processing

Ms. B Eleena, Vanteddu Sony, Alleti Sravanthi, Nomula Sruthi

[1]Assistant Professor,ECE Department Bhoj Reddy Engineering College for Women

[2,3,4]B. Tech Students, Department Of Ece, Bhoj Reddy Engineering College For Women, India.

shruthireddynomula26@gmail.com

ABSTRACT

The increasing need for accessible communication for the hearing-impaired community has led to advancements in technology, particularly in the field of Hand Sign Detection for American Sign Language (ASL). This project explores the development of an AI-driven hand sign detection system using image processing techniques in Python. By leveraging convolutional neural networks (CNNs) and machine learning algorithms, the system is capable of recognizing and interpreting ASL gestures from live video streams or static images.The model is trained on a dataset of ASL hand signs, using Python libraries such as OpenCV for image preprocessing and TensorFlow or Keras for building and training the neural network. The system processes input images, identifies hand gestures, and maps them to their corresponding ASL letters or words, providing real-time feedback. This project aims to bridge communication gaps by offering a tool that can be used for learning ASL or assisting in daily interactions between the hearing and hearing-impaired communities. The proposed system has applications in educational tools, assistive technologies, and real-time translation services. With further training and optimization, it has the potential to improve the quality of life for individuals who rely on sign language as their primary mode of communication.

1-INTRODUCTION

Humans can communicate with one another in a variety of ways. This include behaviour such as physical gestures, facial expressions, spoken words, etc. However, those who have hearing loss are restricted to using hand gestures to communicate. People with hearing loss and/or speech impairments communicate using a standard sign language that is incomprehensible to non-user communicate using a standard sign language that is incomprehensible.

Sign language is the communication system for those who are hard of hearing and deaf. It ranks as the sixth most utilized language worldwide. It is a type of communication that uses hand movements to communicate ideas. Each region has its specific sign language like normal language. In 2005 there were an estimated 62 million deaf people worldwide and about 200 different sign languages in use around the world, many of which have distinctive features.

ASL is the primary language of many deaf citizens in North America. Hard-of-hearing and hearing people also use it. Hand gestures and facial expressions are used to convey this language. The deaf community has access to ASL as a means of communication with the outside world and inside the community. But not everyone is familiar with the signs and motions used in sign language. Understanding sign language and being familiar with its motions takes a lot of practice. Since there are no reliable, portable tools for identifying sign language, learning sign language takes a lot of time.

However, since the development of neural networks and deep learning, it is now possible to create a system that can identify things, or even objects of different categories.

ISSN: 2456-4265
IJMEC 2025

2-SOFTWARE REQUIREMENTS

The software requirements section outlines the tools, platforms, and frameworks necessary to develop, implement, and test the proposed system for hand sign detection in American Sign Language (ASL) using AI and image processing. These requirements are carefully chosen to ensure efficiency, scalability, and compatibility with the objectives of the project. This includes programming languages, development environments, and libraries for image processing and deep learning. Additionally, the software must support real-time processing, handle diverse datasets, and be adaptable for deployment on various platforms, such as mobile devices or edge devices. The right combination of software tools and frameworks will enable the development of a robust, efficient, and user-friendly system, ensuring high accuracy and performance in real-world scenarios. This section will provide a detailed list of all required software components, their roles in the project, and the rationale for their selection.

i. **Image Processing**

This module covers various image processing operations such as image filtering, geometrical image transformations, color space conversion, histograms, etc. In the Java library of OpenCV, this module is included as a package with the name org.opencv.imgproc.

ii. **Video**

This module covers the video analysis concepts such as motion estimation, background subtraction, and object tracking. In the Java library of OpenCV, this module is included as a package with the name org.opencv.video.

iii. **Video I/O**

This module explains the video capturing and video codecs using OpenCV library. In the Java library of OpenCV, this module is included as a package with the name org.opencv.videoio.

iv. **calib3d**

This module includes algorithms regarding basic multiple-view geometry algorithms, single and stereo camera calibration, object pose estimation, stereo correspondence and elements of 3D reconstruction. In the Java library of OpenCV, this module is included as a package with the name org.opencv.calib3d.

v. **features2d**

This module includes the concepts of feature detection and description. In the Java library of OpenCV, this module is included as a package with the name org.opencv.features2d.

vi. **Objdetect**

This module includes the detection of objects and instances of the predefined classes such as faces, eyes, mugs, people, cars, etc. In the Java library of OpenCV, this module is included as a package with the name org.opencv.objdetect.

vii. **Highgui**

This is an easy-to-use interface with simple UI capabilities. In the Java library of OpenCV, the features of this module is included in two different packages namely, org.opencv.imgcodecs and org.opencv.videoio.

NumPy

NumPy is a Python package. It stands for 'Numerical Python'. It is a library consisting of multidimensional array objects and a collection of routines for processing of array.

Numeric, the ancestor of NumPy, was developed by Jim Hugunin. Another package Numarray was also developed, having some additional functionalities. In 2005, Travis Oliphant created NumPy package by incorporating the features of Numarray into Numeric package. There are many contributors to this open source

3-HAND SIGN DETECTION USING AI AND

IMAGE PROCESSING

Communication is a fundamental aspect of human interaction, yet millions of hearing and speech-impaired individuals face challenges in expressing themselves due to language barriers. American Sign Language (ASL) serves as a vital medium for communication within this community, but its limited understanding among non-ASL speakers often hinders effective interaction. Hand sign detection using Artificial Intelligence (AI) and image processing offers a promising solution to bridge this gap by automatically recognizing and interpreting ASL gestures.

The rapid advancements in AI, particularly in computer vision and deep learning, have enabled the development of systems capable of recognizing complex hand gestures with remarkable accuracy. By leveraging these technologies, it becomes possible to create a real-time, robust, and user-friendly system for ASL recognition.

Such a system can empower the hearing and speech-impaired community, facilitating better communication in various contexts, including education, workplaces, and public services.

Existing System

Previous researchers have emphasised their work on the prediction of sign language gestures to support people with hearing impairments using advanced technologies with artificial intelligence algorithms. Although much research has been conducted for SLR, there are still limitations and improvements that need to be addressed to improve the hard-of-hearing community. This section presents a brief literature review of recent studies on SLR using sensor and vision-based deep learning techniques.

Gesture recognition is a hot topic in Human-Computer Interaction research. It has a wide range of applications, including virtual environment control, sign language translation, robot control, and music creation. We will create a real-time Sign Language Recognizer using the MediaPipe framework and Tensorflow in OpenCV and Python in this machine learning project on American Sign Language Recognition.

Proposed System

OpenCV is a real-time computer vision and image processing framework built on C/C++. However, we will use it in Python via the OpenCV-python package.

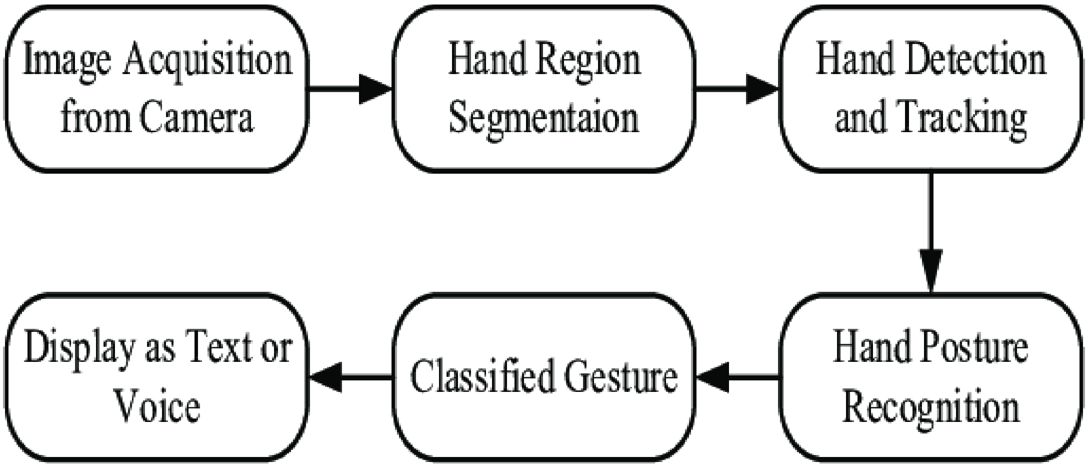


Fig.3.1:Block diagram of hand sign to text

1. **Image Acquisition from Camera**

   This is the first step of the system. A camera continuously captures live video or images of a person performing hand gestures (sign language). These visual inputs are then forwarded to the system for further processing. The goal here is to collect raw data of hand movements in real- time.

2. **Hand Region Segmentation**

   After capturing the image, the system isolates the hand region from the background. This is done using image segmentation techniques such as color thresholding, background subtraction, or deep learning-based segmentation. The aim is to remove any irrelevant background and focus only on the hand for better detection accuracy.

3. **Hand Detection and Tracking**

   Once the hand is segmented, the system detects the presence, shape, and movement of the hand. This step includes:

1. Identifying the coordinates and orientation of the hand.
2. Continuously tracking its movement across video frames.

   This ensures consistent monitoring of dynamic gestures (like waving or drawing shapes in the air).

4. Hand Posture Recognition

   In this stage, the system analyzes the posture or gesture of the hand (e.g., open palm, closed fist, specific finger positions). Using machine learning or CNN models, the system recognizes these postures and maps them to corresponding signs or characters in sign language.

5. **Classified Gesture**

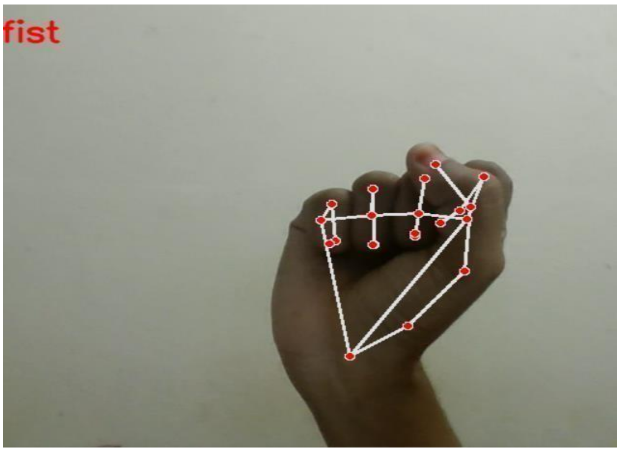
   The recognized posture is now classified as a specific letter, word, or symbol. Classification models match the detected gesture to a predefined set of gestures stored in a trained database. This step transforms visual data into meaningful information.

Methodology

The proposed system follows a vision-driven methodology to perform gesture-based sign recognition from frames extracted from video inputs. The sign recognition process consists of three phases, namely data collection, data pre-processing and feature extraction and gesture recognition. After the data collected is pre-processed and augmented, the feature extraction process is initiated. In this, facial features, landmarks of both hands, and bodily postures are extracted as keypoints from a sequence of input frames captured via a web camera. Then the extracted essential data points are considered as crucial features to be fed to the implemented classifier that recognizes the gestures performed by the user. These recognized gestures and moments are further converted to the textual form and displayed on the screen in real time. Figure 8 shows the architectural diagram of the proposed system.

4-RESULT

Hand Sign to speech/Text



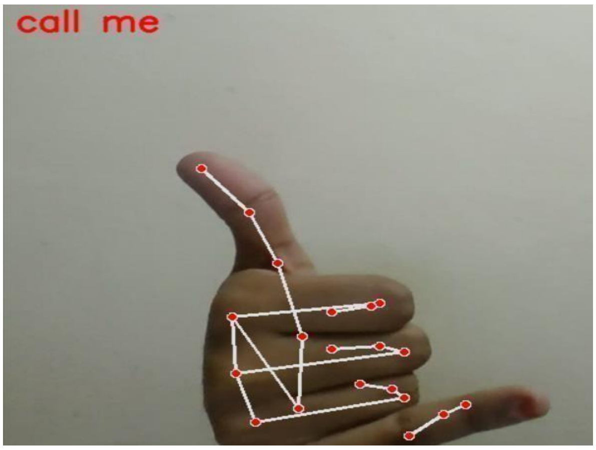
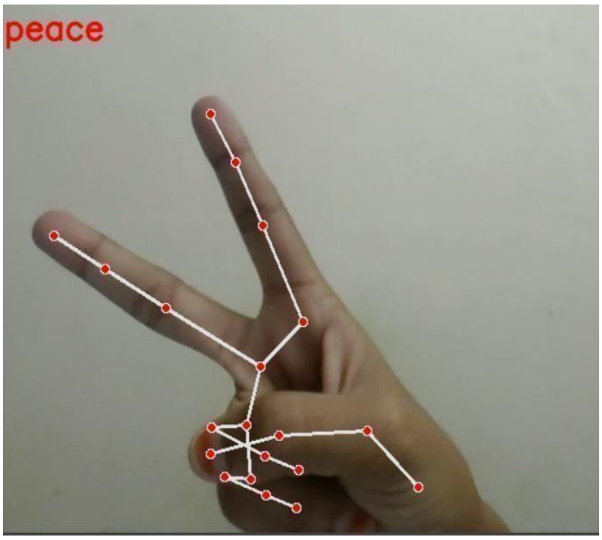


Fig.1: Fist hand gesture

Fig.2: Call me hand gesture

Figure.5.1:The system has successfully identified and connected key landmark points (in red dots) across the knuckles and joints using white lines. The structure shown highlights the bent fingers, indicating that the hand is closed. The label detected is "fist", which corresponds accurately with the physical hand gesture.

Figure.5.2:The system detects key hand landmarks using red points and white connecting lines.The shape formed clearly distinguishes the raised thumb and little finger, with other fingers folded. The gesture is classified correctly as "call me", showing the model's ability to differentiate complex hand poses.



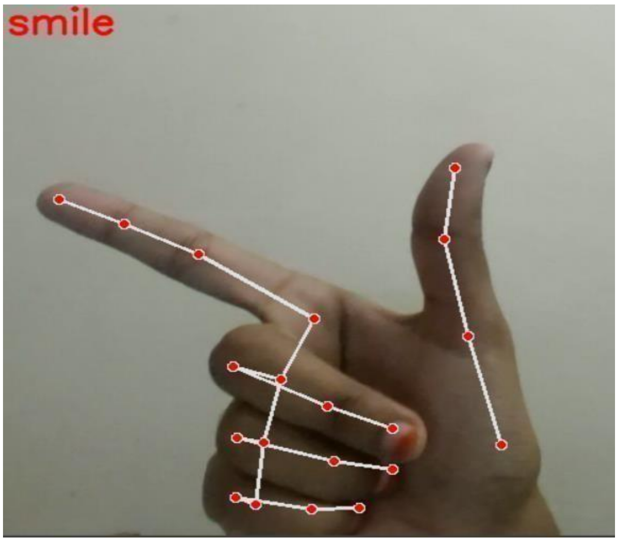


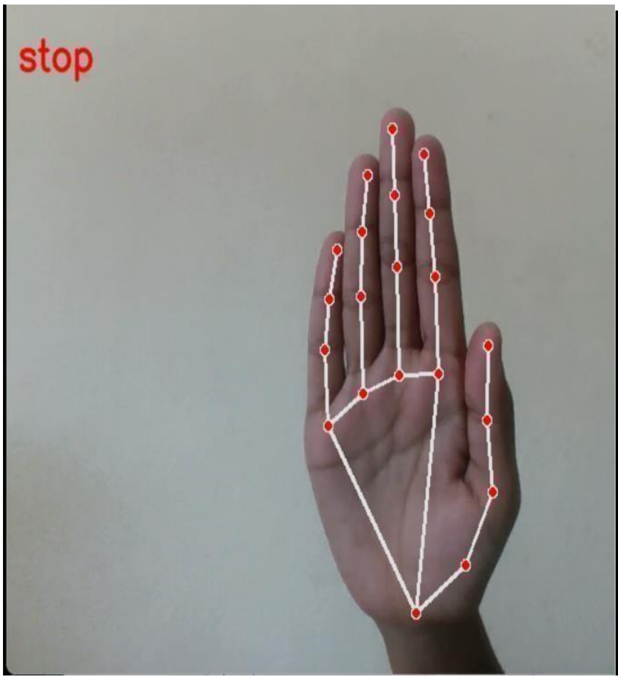
Fig.3:Peace hand gesture

Fig.4:Smile hand gesture

International Journal of Multidisciplinary Engineering in Current Research - IJMEC
Volume 10, Issue 6, June-2025, http://ijmec.com/, ISSN: 2456-4265

Figure.3: The image shows a classic "peace" sign, where the index and middle fingers are raised and spread apart, while the other fingers are folded into the palm. The system has effectively detected and highlighted key hand landmarks using red dots and connected lines.This type of gesture is used commonly in both casual communication and specific sign languages.

Figure.4: Based on the detected landmarks, the thumb and index finger are forming a semicircular shape, while the other fingers are relaxed. Although the exact gesture meaning might vary, the system has classified this particular hand pose as "smile."The hand tracking shows clear and precise positioning of each finger joint, proving the model's capacity to handle diverse gestures.



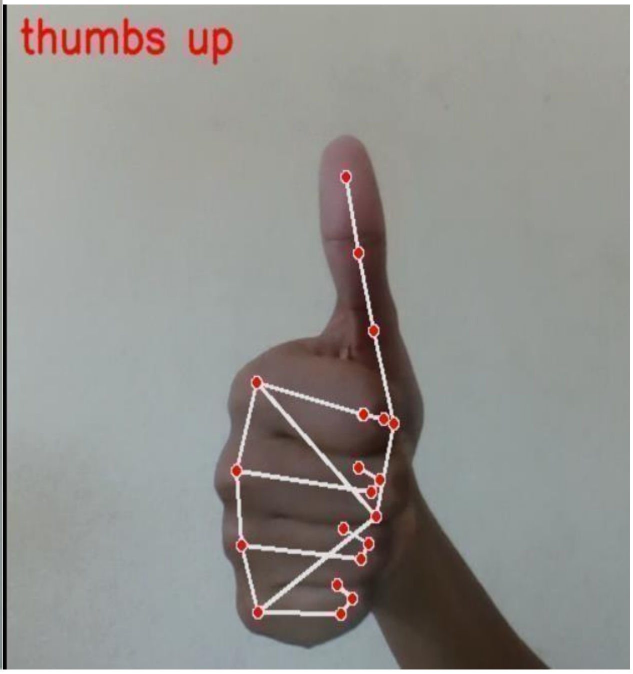


Fig.5:Stop hand gesture

Fig.6:thumbs up hand gesture Figure.5.5:"Stop" hand gesture, where the palm is fully visible with all fingers extended and spread apart. The system marks each joint with red dots connected by lines, clearly outlining the hand's structure and emphasizing the open palm posture, which is typically interpreted as a command to halt or pause. Figure.5.1:"Thumbs Up" gesture, where the thumb is extended upward while the remaining fingers are curled into the palm. Similar landmark annotations highlight the thumb's upright position, indicating a gesture commonly used to express approval or agreement. These examples demonstrate the system's capability to accurately detect finger joints and classify gestures based on precise hand positions, supporting its potential use in gesture-controlled applications or human-computer interaction systems.

Speech to Hand Sign

Fig.7:ComeAgain

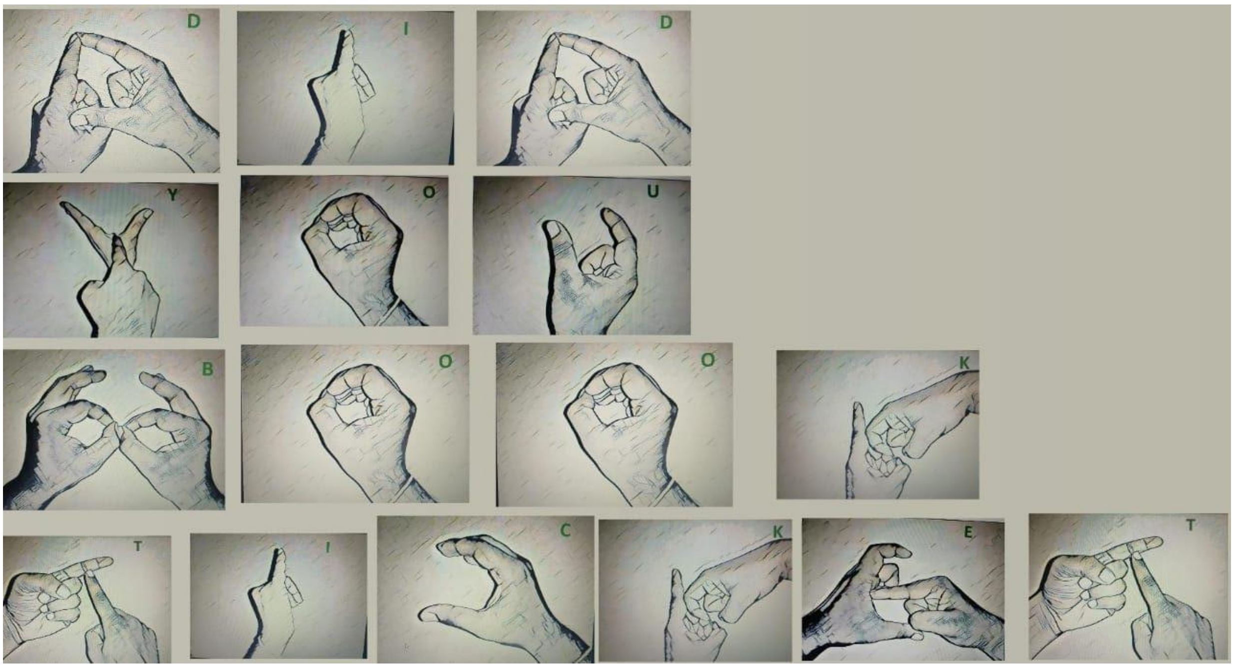


Fig.8:Did You Book Tickets

5-CONCLUSION

Using the popular MediaPipe and LSTM, this comprehensive study proposed and developed a system for American Sign Language recognition. Initially, a folder was created for gestures, and for each gesture, 30 subfolders were created; these subfolders can be thought of as video folders, and each subfolder contains 30 frames, each of which is in the form of a numpy array containing landmark values detected and extracted using Mediapipe Holistic Solution. The data was used to train the LSTM network, which yielded an accuracy of 80% on testing data. Finally, the system was tested using real-time data that was directly fed into the model, and the results for each gesture were displayed on the screen and the text is translated into speech. There was some lag when recognising gestures in real time. We learned about how sometimes basic approaches work better than complicated approaches. We also realised the time constraints and difficulties of creating a dataset from scratch.

REFERENCES

[1] K. Cheng, “Top 10 & 25 American sign language signs for beginners – the most know top 10 & 25 ASL signs to learn first: Start ASL,” Start ASL Learn American Sign Language with our Complete 3-Level Course!, 29-Sep-2021. [Online]. Available: Top 10 & 25 American Sign Language Signs for Beginners – The Most Know Top 10 & 25 ASL Signs to Learn First.

[2] Mittal, P. Kumar, P. P. Roy, R. Balasubramanian, and B.B. Chaudhuri, "A Modified LSTM Model for Continuous Sign Language Recognition Using Leap Motion," in IEEE Sensors Journal, vol. 19, no. 16, pp. 7056-7063, 15 Aug.15, 2020.

[3] V. Sharma, M. Jaiswal, A. Sharma, S. Saini and R. Tomar, "Dynamic Two Hand Gesture Recognition using CNN-LSTM based networks," 2021 IEEE International Symposium on Smart Electronic Systems (iSES), 2021,pp. 224-229, doi: 10.11.09.

[4] M. Wurangian, “American sign language alphabet recognition,” Medium,15-Mar-2021. [Online]. Available: American Sign Language Alphabet Recognition | by Marshall Wurangian | MLearning.ai | Medium.

[5] A. Dennisan, “American sign language alphabet recognition using Deep Learning,” ArXiv, 10-Feb-2022. [Online]. Available: American Sign Language Alphabet Recognition using Deep Learning.

[6] P. Likhar, N. K. Bhagat and R. G N, "Deep Learning Methods for Indian Sign Language Recognition," 2020 IEEE 10th International Conference on Consumer Electronics (ICCE- Berlin), 2020, pp. 1-6.

[7] Shivashankara, Ss, and S. Srinath. “American sign language recognition system: an optimal approach.” International Journal of Image, Graphics and Signal Processing 11, no. 8 (2022): 18.

[8] D. G. Lowe, Distinctive Image Features from Scale-Invariant Keypoints, International Journal of Computer Vision, vol. 13, no. 2, pp. 111122, 2020.