

ASL Sign Language Recognition System

Prof. Keshav Tambre¹, Swarup Diwan², Parikshit Doye³, Siddharth Dixit⁴, Dnyanesh Bagulkar⁵,
Prashansa Dodke⁶

¹*Project Guide, Department of Engineering Sciences and Humanities (DESH), Vishwakarma Institute of Technology, Pune, Maharashtra, India*

^{2,3,4,5,6}*Department of Engineering Sciences and Humanities (DESH), Vishwakarma Institute of Technology, Pune, Maharashtra, India*

Abstract— This project focuses on designing a solution for the American Sign Language (ASL) recognition system based on software developed on PyCharm. The recognition of ASL gestures uses a combination of machine learning and computer vision techniques. OpenCV is responsible for capturing the videos in real time, whereas TensorFlow with Mediapipe is used for hand tracking and gesture capturing. The model performs recognition of ASL signs with ASL sign dataset and generates text or speech output. The system functions as an interpreter for sign language in real-time and improves the communication of people with hearing impairments. It removes the need for specialized, expensive hardware which makes it more affordable and easier to use. The project goes even further by providing an intuitive graphical user interface so that the user does not have to struggle with interaction. The ASL recognition system may serve education, assistive technology, and real time communications efforts for bridging the gap between the hearing and deaf world.

Index Terms— ASL, communication, deaf community, gesture recognition, sign language

I. INTRODUCTION

American Sign Language (ASL) is one of the most widely used sign languages among the deaf and hard-of-hearing communities, primarily in the United States and Canada. Unlike spoken languages that rely on sound, ASL uses hand gestures, facial expressions, and body movements to convey meaning. However, communication between hearing-impaired individuals and those unfamiliar with sign language often remains challenging. This communication gap motivates the development of automated systems capable of recognizing and translating ASL gestures into text or speech. With the advancement of computer vision and deep learning, it has become possible to recognize

hand gestures in real time with high accuracy. The integration of frameworks such as OpenCV, TensorFlow, and Mediapipe allows efficient hand tracking, feature extraction, and gesture classification without requiring specialized hardware. These technologies enable the creation of a cost-effective and user-friendly ASL recognition system that can operate on standard computing devices using only a webcam. The proposed ASL Sign Language Recognition System aims to bridge the communication gap between the hearing and the speech or hearing-impaired communities by converting ASL gestures into corresponding text or speech output. The system captures real-time hand gestures through a webcam, preprocesses the input using image enhancement and segmentation techniques, and classifies the gestures using a deep learning model trained on an ASL dataset. The output is displayed on-screen as text and can optionally be converted into speech for easier understanding by non-signers. This work contributes to the development of assistive technologies that promote accessibility and inclusion. By providing an affordable, software-based solution, the system eliminates the need for costly sensor gloves or external devices and supports real-time interpretation in educational, social, and communication settings.

II. LITERATURE REVIEW

Srivastava et al. (2021) proposed a real-time Sign Language Recognition (SLR) system using the TensorFlow Object Detection API to translate Indian Sign Language (ISL) gestures into English. Unlike previous ISL recognition models, which relied on machine learning with MATLAB and achieved high

accuracy (93-96%) but lacked real-time capabilities, this study aimed to develop a cost-effective, real-time detection system. The dataset was created using webcam images and trained using transfer learning with SSD MobileNet v2. The system achieved an average confidence rate of 85.45%, demonstrating its potential for real-world application. Future improvements include expanding the dataset and exploring alternative deep learning models to enhance accuracy and recognition capabilities.

Padmaja et al. (2022) proposed a real-time sign language detection system using deep learning techniques to facilitate communication between deaf/mute individuals and non-signers. The study highlights the limitations of traditional glove-based recognition methods, which, despite being highly accurate, are impractical for daily use. The authors developed a vision-based approach utilizing Faster R-CNN and ResNet-50 for detecting and classifying hand gestures. A dataset of 200 images, including letters V, L, U, and the phrase "I Love You," was used for training. The system achieved an 86% accuracy, demonstrating the feasibility of deep learning in sign language recognition. This research contributes to the development of cost-effective and accessible solutions for bridging communication gaps.

III. METHODOLOGY/EXPERIMENTAL

learning and computer vision techniques. The system captures hand gestures through a webcam, processes the images to extract hand features, classifies the gestures using a pre-trained deep learning model, and displays the recognized sign language alphabet.

The methodology consists of four major stages:

1. Data Collection & Preprocessing
2. Model Training & Classification
3. Real-Time Hand Detection & Feature Extraction
4. Gesture Prediction & Display Output

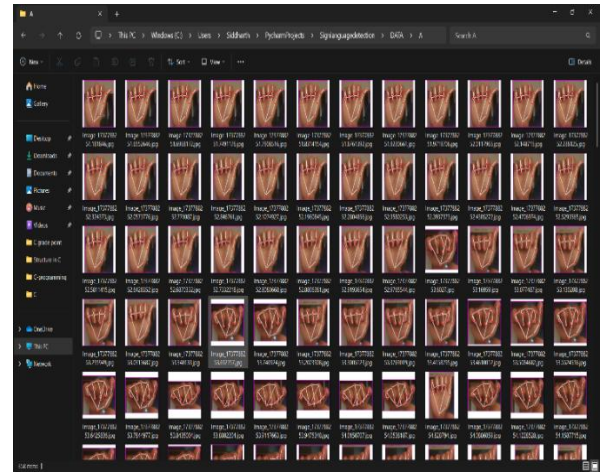
Each stage is elaborated below.

1) Data Collection

To train the model, a dataset of hand gestures corresponding to American Sign Language (ASL) alphabets (A–E) was created. The dataset was built using Google Teachable Machine, which enables rapid collection and annotation of gesture images. Each

class consists of n number of images, ensuring robust training data.

Fig. 1. Dataset



2) Preprocessing Steps

The following preprocessing techniques were applied:

- Image Resizing: All images were resized to 300×300 pixels for consistency.
- Normalization: Pixel values were normalized to improve training efficiency.
- Data Augmentation: Rotation, flipping, and contrast adjustments were applied to enhance model generalization.

The preprocessed images were then exported and used for training.

3) Model Architecture

The classification model was developed using TensorFlow and trained using Google Teachable Machine. The model consists of:

- Input Layer: Accepts 300×300 RGB images.
- Convolutional Layers: Extracts hand gesture features.
- Fully Connected Layers: Processes feature representations.
- Softmax Output Layer: Classifies hand gestures into five categories (A–E)

Fig 2. System architecture of the proposed ASL recognition model.

```

if aspectRatio > 1:
    k = imgSize / h
    wCal = math.ceil(k * w)
    imgResize = cv2.resize(imgCrop, (wCal, imgSize))
    imgResizeShape = imgResize.shape
    wGap = math.ceil((imgSize - wCal) / 2)
    imgWhite[:, wGap:wCal + wGap] = imgResize
else:
    k = imgSize / w
    hCal = math.ceil(k * h)
    imgResize = cv2.resize(imgCrop, (imgSize, hCal))
    imgResizeShape = imgResize.shape
    hGap = math.ceil((imgSize - hCal) / 2)
    imgWhite[hGap:hCal + hGap, :] = imgResize

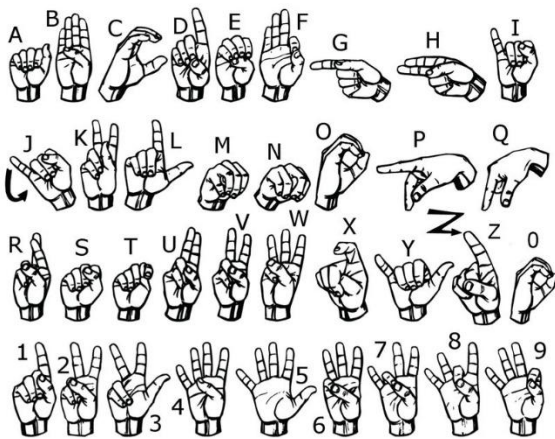
```

4) Training Process

- Loss Function: Categorical Cross-Entropy
- Optimizer: Adam Optimizer
- Batch Size: 32
- Epochs: 20
- Validation Split: 20% of the dataset for model validation

After training, the model was exported as a Keras model (keras_model.h5), and the corresponding label file (labels.txt) was generated.

Fig 3. Dataset samples of ASL alphabets



5) Image Cropping & Preprocessing

Once the hand is detected, the system performs the following steps:

1. Extracts the bounding box coordinates of the detected hand.
2. Adds an offset (20 pixels) around the hand for better feature extraction.

3. Crops the image from the original frame.
 4. Resizes the cropped image while maintaining the aspect ratio.
 5. Centers the image on a 300×300 white background to ensure uniform input to the model.
- These preprocessing steps enhance classification accuracy by reducing background noise and standardizing input dimensions.

6) Gesture Prediction & Display Output

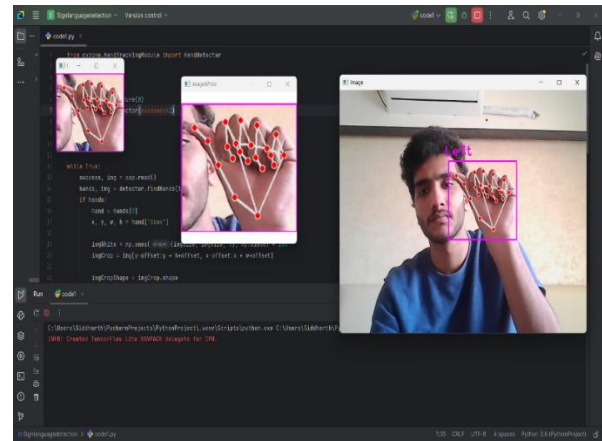
Prediction Using Trained Model

The preprocessed hand gesture image is passed to the trained classifier using the CvZone ClassificationModule, which loads the keras_model.h5 model and predicts the hand gesture.

The prediction process follows these steps:

1. The classifier receives the processed image as input.
2. Feature extraction is performed using the pre-trained convolutional layers.
3. The model generates probability scores for each class.
4. The class with the highest probability is selected as the final output.
5. The corresponding label is displayed on the screen.

Fig 4. Real-time gesture detection and classification interface.



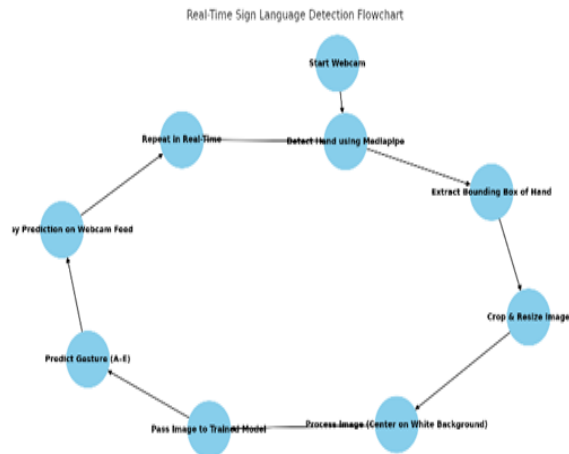
7) Real-Time Visualization

The system provides a real-time display of:

- The original webcam feed
- The cropped and processed hand image
- The predicted letter (A–E) overlaid on the webcam feed

This visualization ensures users can interact with the system effectively and receive immediate feedback on recognized gestures

Fig 5. Flow Diagram of ASL Recognition Workflow



IV. RESULT AND DISCUSSIONS

The high level of accuracy demonstrated in this project showcases the power of machine learning and computer vision in real-time sign language identification. The application of OpenCV and Mediapipe for hand detection and location tracking played a vital role in the performance of the system. Despite this, some challenges were experienced during development:

1. Variability in Hand Gestures – The system is good at handling trained gestures but has difficulties with hand size variations, skin tone, and lighting.
2. Real-time Performance – Although the recognition is good on a controlled dataset, real-time accuracy may be impacted by background noise and speed of hand movement.
3. Static vs. Dynamic Gestures – The system now recognizes static signs well but not dynamic gestures or continuous signing, which are necessary for smooth communication.

To address these limitations, future work will include:

- Increasing the dataset to incorporate more varied hand gestures.
- Improving real-time processing through the optimization of the model for speed.

- Support for dynamic gestures via Recurrent Neural Networks (RNNs) or Transformer models.
- The creation of an Android app in order to increase the accessibility of the system for users.

The project's significance is that it fills in the gap of communication between people with speech and hearing disabilities and the public. Through the integration of deep learning methods, the system can potentially provide greater accessibility and inclusion in daily life.

The aim of the project was to build a system that is both more accessible, as well as, real-time while translating American Sign Language (ASL) to text and audio. The objectives were accomplished as the system developed was able to identify hand movements through a combination of computer vision and machine learning. The model was able to identify ASL images with a 98.82% accuracy after some tweaking of parameters.

To text and speech output goes the system converted of sign language is based on the recognition of gestures by means of features extraction using HOG, and classification by SVM. Furthermore, there was also a CNN model built which resulted in a training accuracy of 99.86% and testing accuracy of 94.68%. These figures prove that the system is efficient in ASL static signs recognition.

In addition, the model was tried against a Kaggle ASL dataset of 27 classes which had space gesture for appropriate word segmentation. The text speech synthesis module also performed as expected, producing speech upon recognition of text which can be very important for blind people. The user interface(UI) of the system was also evaluated and it was found that deaf and normal hearing users were able to operate the system with ease.

V. FUTURE SCOPE

1. Linguistic Analysis of ASL – Analyzes ASL's phonology, morphology, syntax, grammar, and dialect differences, contrasting its structure with oral languages.
2. Cognitive and Psychological Studies – Explores ASL acquisition, cognitive processing, and brain accommodation to visual-spatial language.

3. Educational Approaches – Concentrates on ASL teaching methods, standardized programs, and bilingual education models.
4. Sociocultural and Anthropological Studies – Investigates the role of ASL in Deaf culture, its history, and social integration.
5. Human-Computer Interaction (HCI) – Designs accessible ASL technology, ML-free gesture recognition, and wearable communication devices.
6. Ethnographic and Qualitative Research – Examines the daily experience of ASL users and community aspects.
7. Legal and Policy Research – Investigates ASL's legal status, accessibility laws, and workplace accommodations.
8. Art and Media Representation – Explores ASL in performing arts, media representation, and Deaf storytelling.
9. Interdisciplinary Collaborations– Explores the role of ASL in music, dance, movement arts, and visual arts.
10. Ethical and Philosophical Considerations – Explores ethical research practices, language identity, and representation in a hearing-majority society.

VI. CONCLUSION

In this project, we proposed an idea for feasible communication between hearing and vision impaired people and a person with ability to hear and see more compared to the former, with the help of deep learning and machine learning approach. A real-time sign language detection system fosters inclusivity by instantly translating gestures, empowering communication for the hearing and speech impaired.

ACKNOWLEDGMENT

The authors express their gratitude to Prof. Keshav Tambre for guidance and to Vishwakarma Institute of Technology for supporting this research.

REFERENCES

- [1] M. J. C. Samonte, C. J. M. Guingab, R. A. Relayo, M. J. C. Sheng, and J. R. D. Tamayo, "Using Deep Learning in Sign Language Translation to Text," in *Proceedings of the International Conference on Industrial Engineering and Operations Management (IEOM)*, 2022.
- [2] N. Padmaja, B. N. S. Raja, and B. P. Kumar, "Real-Time Sign Language Detection System Using Deep Learning Techniques," *Journal of Pharmaceutical Negative Results*, vol. 13, Special Issue 1, 2022.
- [3] S. Srivastava, A. Gangwar, R. Mishra, and S. Singh, "Sign Language Recognition System Using TensorFlow Object Detection API," in *Proceedings of the International Conference on Advanced Network Technologies and Intelligent Computing (ANTIC 2021), Communications in Computer and Information Science (CCIS)*, Springer, 2021.
- [4] J. Ajay, R. Sumathi, A. Kalliyadath, D. Hemanth, and N. S. K., "Analysis of Machine Learning Techniques for Sign Language to Text Conversion for Speech Impaired," in *Proceedings of the 2023 International Conference on Computer Communication and Informatics (ICCCI)*, Coimbatore, India, IEEE, 2023.
- [5] H. Orovwode, I. D. Oduntan, and J. Abubakar, "Development of a Sign Language Recognition System Using Machine Learning," in *Proceedings of the International Conference on Artificial Intelligence, Big Data, Computing and Data Communication Systems (ICABCD)*, IEEE, 2023.