JETIR.ORG ISSN: 2349-5162 | ESTD Year : 2014 | Monthly Issue
JOURNAL OF EMERGING TECHNOLOGIES AND INNOVATIVE RESEARCH (JETIR)
An International Scholarly Open Access, Peer-reviewed, Refereed Journal

Real Time Sign Language Recognition Using ASL

1. Vadipina Amarnadh, Assistant Professor, CSE Department, Anurag University
2. Harshitha Boredha, 20EG105211, AU 3. Nagesh Bhukya, 20EG105209, AU
4. Jonnal Varun Reddy, 20EG105220, AU

***Abstract*—** In a world reliant on communication for connection and idea exchange, the challenges faced by 95 million individuals globally affected by deafness and speech difficulties underscore the crucial need for innovative solutions. Our commitment lies in pioneering advancements American Sign Language (ASL) using cutting-edge Computer Vision technology. This groundbreaking software aims to break down communication barriers, enabling intuitive gesture-based interactions across various applications. Our core mission is to bridge the gap between the richness of ASL communication and the lack of technological support. We aspire to empower individuals with hearing and speech impairments to express themselves effectively while fostering deeper understanding and engagement within the wider community. Leveraging the intricacies of ASL and advancements in machine learning, our system aims to establish an easily accessible, reliable mechanism for accurate sign language detection. Our pursuit is more than technological innovation; it's a passionate endeavor to elevate inclusivity, empathy, and meaningful connections in our diverse global landscape.

Keyword*—*ASL, Machine Learning, Accurate detection, Computer Vision technology, Communication barrier, hearing and speech impariments

I. INTRODUCTION

Effective communication serves as the cornerstone of human interaction and knowledge exchange, forming the basis of our social and intellectual fabric. Yet, for the approximately 95 million individuals worldwide affected by deafness and speech difficulties, this fundamental aspect of human connection poses profound challenges, often resulting in isolation and limited access to crucial information. In light of this, the development of innovative solutions becomes imperative, not only to facilitate communication but also to foster inclusivity and equal participation within our communities.

The motivation behind this project arises from a deep-seated commitment to bridge the communication gap faced by individuals with speech and hearing impairments. Our primary focus lies in the creation of an avant-garde sign language recognition software specifically tailored for American Sign Language (ASL), leveraging state-of-the-art Computer Vision technology. This groundbreaking software endeavors to empower the disabled community by facilitating seamless communication through intuitive and efficient gesture-based interactions with a range of applications.

Understanding the critical importance of comprehensive support, our project is dedicated to addressing the existing challenges in technology and accessibility faced by those utilizing ASL as their primary means of communication. By amalgamating the complexities of ASL with the advancements in machine learning, our envisioned system aims to serve as a reliable and comprehensive mechanism for accurately detecting and interpreting sign language, thereby fostering greater connectivity and understanding within the broader community.

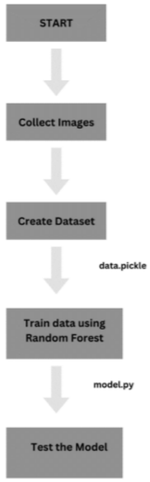
Through this comprehensive undertaking, we aspire not only to provide a practical solution for individuals with hearing and speech impairments but also to catalyze a broader shift towards a more inclusive and empathetic society. By amalgamating cutting-edge technology with a deep understanding of the needs and nuances of ASL communication, our project seeks to be a catalyst for positive change and equitable access to communication for all.

II. METHODOLOGY

The methodology employed in this research project is multifaceted, encompassing a comprehensive approach to data collection, preprocessing, and model development. The meticulous curation of the dataset stands as the cornerstone of this endeavor, ensuring a well-balanced representation of various American Sign Language (ASL) gestures. Each step in the data collection process is carefully orchestrated to capture the intricacies and nuances of ASL, allowing for a more nuanced and accurate recognition system.

Collecting I*mages*

This initial step involves collecting a dataset of images that represent various American Sign Language (ASL) gestures. These images typically include hand gestures corresponding to different letters or words in the ASL alphabet. The dataset should be diverse and include variations in hand shapes, hand orientations, and backgrounds to ensure robust recognition.While collecting the data set, each symbol is captured for 100 times, with various angles and distance of the hand. Overall, the dataset size is 2,700 images



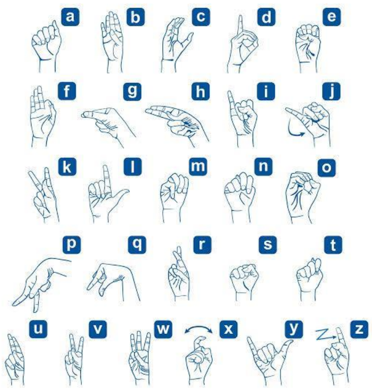


Fig 1: Architecture **Fig 2:** ASL Gestures

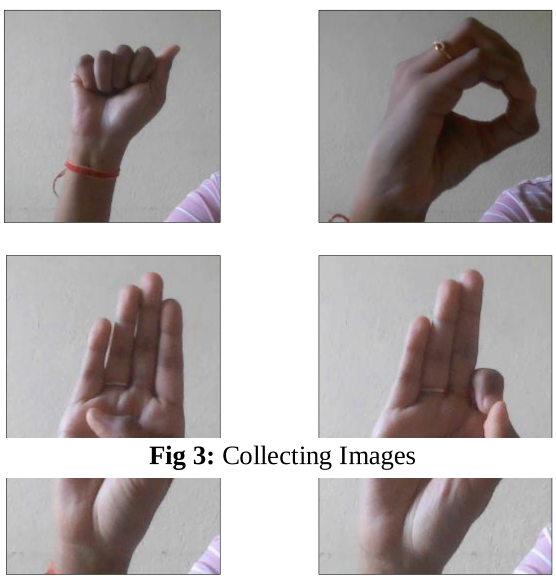


Fig 3: Collecting Images

Creating Dataset

After collecting the ASL gesture images, the next task is to organize and preprocess this data to create a structured dataset. This dataset may include image-label pairs where each image corresponds to a specific ASL gesture, and labels represent the corresponding letter or word.

The dataset is typically stored in a suitable format, often a pickle file (data.pickle), which allows for efficient loading and use in machine learning models.

Training Images and Testing

Model: Random Forest Algorithm:

The trained model, often serialized into a Python file (model.py), can be used for real-time ASL gesture recognition

Random Forest is an ensemble machine learning algorithm that combines the predictions of multiple decision trees to improve accuracy and reduce overfitting.

In sign language recognition, the Random Forest voting algorithm works in a similar way to how it operates in other contexts, such as classification tasks. The primary goal is to combine the predictions of multiple decision trees to recognize sign language gestures accurately.

Once all the decision trees are built, they are used to make predictions. For classification tasks, Random Forest combines the predictions of individual trees through a majority vote, with the class that receives the most votes becoming the final prediction. For regression tasks, it averages the predictions of individual trees to obtain the final output.

From the dataset size of 2700 images, 80% (2160 images) are used for training and 20% (540 images) for testing.

Creating model:

model = RandomForestClassifier()

Fitting the model:

model.fit(x_train, y_train)

Testing Dataset

After training the dataset, the model is tested and accuracy is calculated, the model classified 99.3% images correctly.

Accuracy:

score = accuracy_score(y_predict, y_test)

print('{}% of samples were classified correctly !'.format(score * 100))

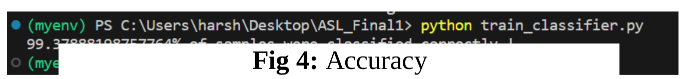


Fig 4: Accuracy

III. Technology

A. *OpenCV*

Version: opencv-python-4.8.0.76

OpenCV is a powerful open-source software library dedicated to computer vision and image processing tasks. It provides a comprehensive set of tools and functions for image and video analysis, enabling applications in areas like object detection, facial recognition, and more.

- Capturing video frames from a webcam (cv2.VideoCapture).
- Converting the video frame from BGR to RGB color space (cv2.cvtColor).
- Recognizing and tracking the position of a person's hand (Hand tracking).
- Detecting and marking hand landmarks (cv2.rectangle, cv2.putText).
- Converting frames for displaying in Tkinter (cv2.imencode).
- Handling user interaction, like quitting the application (cv2.waitKey).

B. TensorFlow

TensorFlow, an open-source deep learning library, is crucial in ASL sign language detection. Its flexible tools allow for building custom models tailored for sign language recognition. Using high-level APIs like Keras, it simplifies complex neural network creation with less code. TensorFlow's GPU support enables efficient handling of large ASL datasets. Leveraging these features, it optimizes neural network design, leading to a robust ASL recognition system adept at interpreting diverse gestures accurately.

C. MediaPipe

Version: mediapipe-0.10.3

MediaPipe is designed to simplify the development of machine learning-based applications that involve multimedia inputs. It is known for its robustness and accuracy in handling various computer vision tasks, making it valuable for applications that rely on visual data.

- Enhancing hand recognition and tracking using the MediaPipe framework.
- Analyzing hand gestures through landmarks (Hand gesture prediction).

D. Scikit-Learn

It is a widely used machine-learning library for Python. It provides an efficient approach to implementing machine learning algorithms and conducting tasks such as classification, regression, clustering, dimensionality reduction, and more.

Training a machine learning model (Random Forest) for recognizing ASL signs (Model creation and training).

Utilizing the machine learning algorithm for gesture classification (Gesture classification).

E. Tkinter

Tkinter is the standard Graphical User Interface (GUI) library for Python. It is known for its simplicity and ease of use, making it a suitable choice for developing applications with user-friendly interfaces.

- Creating a graphical user interface (GUI) for displaying recognized text and video feeds.
- Dividing the user interface into frames for the recognized text and video feed (Frame division).
- Updating the recognized text in a text widget (Text widget updates).
- Managing the main application loop (Tkinter main loop).

IV. CONCLUSION

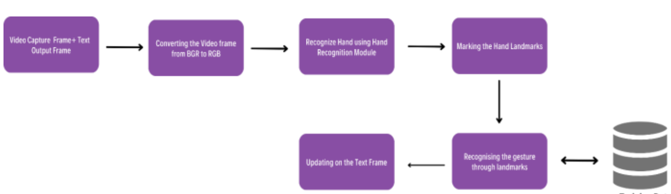


Fig 5: Flow Of Execution

The core of our solution lies in the real-time detection and translation system, where the trained ASL recognition model will be integrated. The program uses the tkinter library to create a user-friendly graphical interface. Within this GUI, two distinct frames are established to organize and display different types of data. The left frame displays recognized ASL signs, which are continuously updated, and the right frame displays real-time video data.

Fig 6: Tinker GUI (User Interface)

The simultaneous display of real-time data allows users to see live video and corresponding ASL text, providing an interactive and informative experience. The recognized text will be updated and displayed within this text output frame.

The captured video frames are typically in the BGR (Blue-Green-Red) colour format, which is the default format for many cameras and OpenCV. To work with these frames more effectively and display them using tkinter, the frames are converted from BGR to RGB format. This conversion ensures that the color representation of the frames is suitable for display in the GUI.

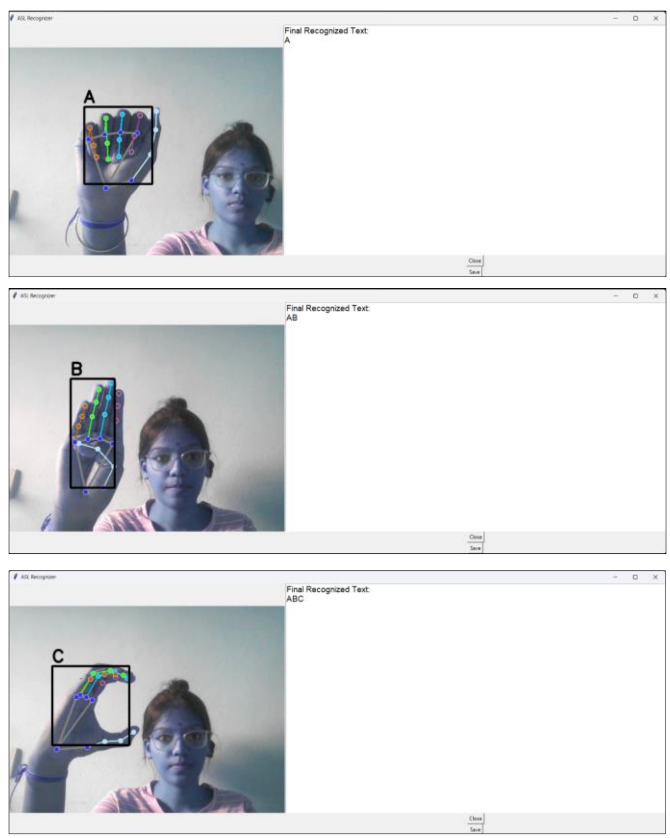


Fig 7: Result

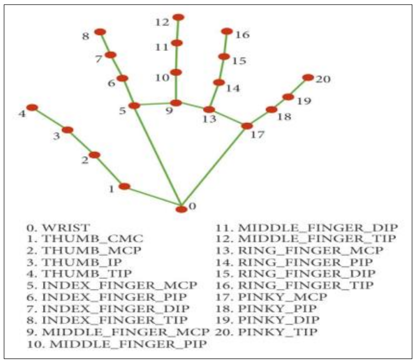


Fig 9: Media Pipe Hand Landmarks

The hand recognition module plays a central role in the ASL recognition system. It identifies and tracks human hands within

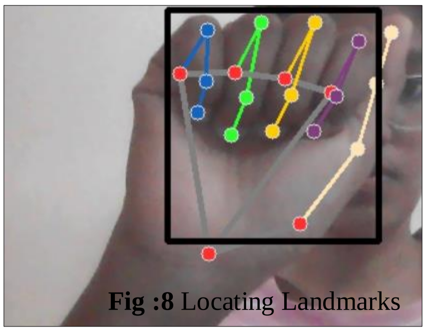


Fig :8 Locating Landmarks

the video frames. For this task, the project uses MediaPipe, a library designed for hand tracking.

MediaPipe is responsible for detecting and locating hands accurately within each video frame. Once detected, it provides a set of landmarks, which are specific points on the hand, to aid in hand pose estimation.

After successfully recognizing the hand within the frame, the project marks specific landmarks on the hand. These landmarks are key points on the hand's surface, such as fingertips and joints, that help in understanding and interpreting the hand's pose and gestures.

These landmarks are essential for accurately recognizing and classifying ASL gestures, as they provide a detailed description of the hand's configuration.

As ASL gestures are recognized and classified, the corresponding text representation of the recognized gesture is continuously updated. The recognized text is displayed within the text output frame on the left side of the GUI. The frame displays the final recognized ASL text, providing a textual representation of the gestures as they are performed. The user can create multiple files which is saved using timestamp. The file is saved within the local directory.

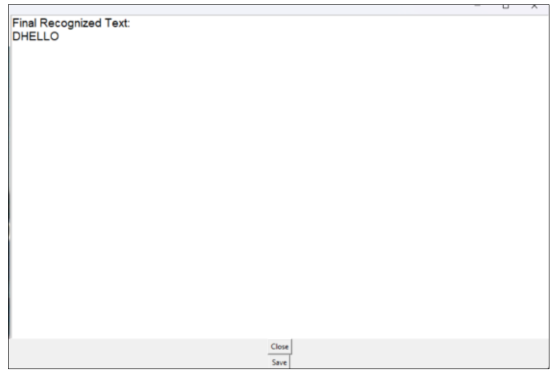


Fig 10: The real time updating on right side of UI

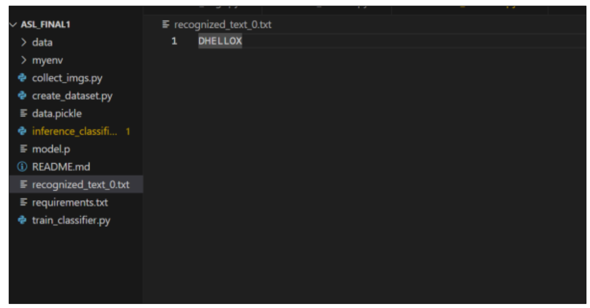


Fig 11: The Files been saved based on timestamp

REFERENCES

[1] S. Y. Kim, H. G. Han, J. W. Kim, S. Lee, and T. W. Kim, "A hand gesture recognition sensor using reflected impulses," IEEE Sens. J., vol. 17, no. 10, pp. 2975–2976, 2017, doi:10.1109/JSEN.2017.2679220.

[2] A. Kumar and A. Kumar, "Dog Breed Classifier for Facial Recognition using Convolutional Neural Networks," pp. 508–513, 2020.

[3] J. Wu, L. Sun, and R. Jafari, "A Wearable System for Recognizing American Sign Language in RealTime Using IMU and Surface EMG Sensors," IEEE J. Biomed. Heal. Informatics, vol.20,no.5,pp.1281–1290,2016, doi: 10.1109/JBHI.2016.2598302.

[4] M. Safeel, T. Sukumar, K. S. Shashank, M. D. Arman, R. Shashidhar, and S. B. Puneeth, "Sign Language Recognition Techniques- A Review," 2020 IEEE Int. Conf. Innov. Technol. INOCON 2020, pp. 1–9, 2020, doi: 10.1109/INOCON50539.2020.9298376.

[5] W. Aly, S. Aly, and S. Almotairi, "Userindependent american sign language alphabet recognition based on depth image and PCANet features," IEEE Access, vol. 7, pp. 123138–123150,2019, doi:10.1109/ACCESS.2019.2938829.

[6] Aman Bhatia, and Mayand Kumar ,"Sign language Recognition using Convolutional neural network", IEEE conference doi: 10.1109/ICICCS51141.2021.9432296

[7] Jinalee Jayesh Kumar Raval; Ruchi Gajjar "Real time sign language recognition using computer vision"IEEE conference , 2021,DOI: 10.1109/ICSP51351.2021.9451709