International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 11 Issue: 11 | Nov 2024 www.irjet.net p-ISSN: 2395-0072

Real-time multilingual sing language Translator using python

Manemoni Arun Kumar[1], Penta Krishna Charan[2], RM Noorullah[3]

[1] *B.Tech, Student, Dept. of CSE Institute of Aeronautical Engineering Hyderabad, India*
[2] *B.Tech, Student, Dept. of CSE Institute of Aeronautical Engineering Hyderabad, India*
[3] *Associate Professor, Dept. of CSE Institute of Aeronautical Engineering Hyderabad, India*

Abstract - — *In this respect, it presents considerable challenges in communication barriers between the deaf and hard-of-hearing and the hearing world. The paper thus proposes a real-time, multilingual sign language translation system developed in Python to bridge this gap. The system is designed to solve described needs for inclusive communication that capture sign language gestures through a webcam. These gestures are further processed to be translated into text using machine learning models trained on extensive sign language datasets. Afterwards, it is translated into several languages, making the system versatile in many different environments. Our methodology includes the most up-to-date computer vision techniques for safe detection and correct parsing of hand gestures, even under changing lighting conditions, which is key to realtime performance. Experimental validation is done by measuring gesture recognition accuracy, speed in translation, and multilingual functionality. Promising results demonstrate the system's efficacy in fast translation from sign language to spoken languages, including the possibility of understanding among people who do not share a common language and increasing accessibility. This research provides a great input into human– computer interaction by propagating a practical inclusivity tool. The real-time nature of the translation makes for smooth communication, while the multilingual feature attends to varied user needs. With its makeup of Python, machine learning, and computer vision, such a system could give place to a society that is much more connected and understanding toward one another, where sign language users could be very active in their day-today communications regardless of any spoken language.*

Key Words: Sign Language, Multilingual Translation, Machine Learning, Real-Time Processing, Gesture Recognition, , python ,machine-learning ,OpenCV-python, CNN-model ,media-pipehands, Translator

1.INTRODUCTION

Sign language can be defined as the medium by which deaf and hard-of-hearing people are able to communicate. It represents messages through hand gestures, facial expressions, body movements, and, hence it is a very expressive medium. In spite of the importance of sign language in this community, an understanding of it by a greater part of the population remains very minimal, making it impossible for people who understand and use sign language to communicate with those who do not understand it. Improvements in machine learning and artificial intelligence, through the advent of advanced technology, have opened the way for closing these gaps with the development of systems capable of interpreting sign language in real-time. This project will assist in giving better accessibility and inclusivity to sign language users. Most of the recent state-of-the-art solutions for sign language translation usually suffer from the number of languages supported, real-time processing, and accuracy of results. Remarkably though, most of these solutions have been tailored to one language that prohibits their usability in multilingual societies. This paper presents a real-time multilingual sign language translator to tackle the foregoing challenges. The proposed system would be helpful in recognizing sign language gestures and translating them into different languages execution. The multilingual approach can help this system be useful to a larger number of users, particularly part of multilingual regions. The system is to be developed using a combination of data collection techniques, machine learning models for gesture recognition, and translation algorithms. The data collection process involves capturing hand gesture images and its preprocessing for the enhancement of recognition accuracy. In the gesture recognition module, there is a trained CNN on these pre-processed images to classify various sign language gestures. Once a gesture is recognized, the translation module converts the recognized gesture into text and then translates this text into the target languages. It also provides inbuilt text-to-speech functionality to provide audio output, hence increasing the accessibility of the system for users. The contributions of the paper can be summarized primarily as follows: Real-time Gesture Recognition System: The system recognizes sign language gestures using machine learning techniques accurately in real-time. Multilingual Translation Support: It supports translation into multiple languages, facilitating multilingual communication. It is integrated with text-to-speech capabilities to make it further accessible by allowing users to listen to the translated text.

2.PROPOSED SOLUTION

The real-time multilingual sign language translator system aims to bridge the communication gap for the hearing-impaired community by recognizing sign language gestures and translating them into multiple languages. The system leverages computer vision techniques and machine learning models to achieve high accuracy in gesture recognition and

efficient realtime processing. The core components of the proposed solution include: Data Acquisition: Collecting a comprehensive dataset of sign language gestures using a vision-based approach. Data Preprocessing and Feature Extraction: Enhancing the dataset quality through preprocessing techniques. Gesture Recognition: Utilizing a Convolutional Neural Network (CNN) for classifying gestures. Translation and Text-to-Speech (TTS): Converting recognized gestures into text and translating them into the desired languages with text and audio output.

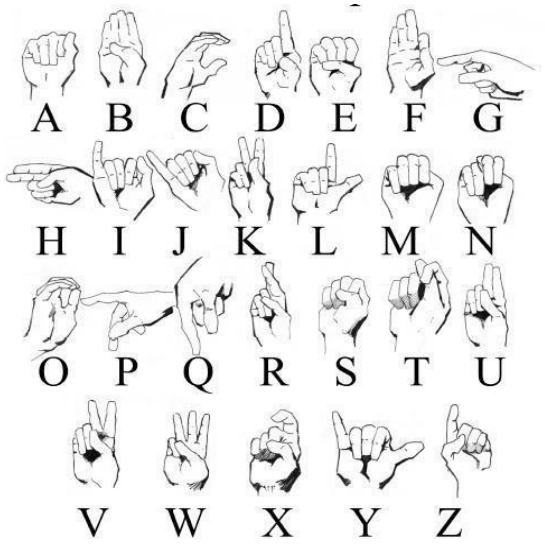

Fig1: sign language alphabet gesture

3. SYSTEM ARCHITECTURE

The architecture of the real-time multilingual sign language translator system guarantees that the sign language gestures are translated properly and accurately into all other languages. This architecture of the system is divided into four sections: Data Collection, Gesture Recognition, Translation Module, and Prediction without GUI. All these divisions are predominant and must be viewed when looking at the overall working of the system. To consider factually, the system diagram of the proposed architecture must be provided:. 1.Data Collection Data collection is the very first step in developing this gesture recognition machine learning model. Two major scripts are used to capture and process gesture images within the system:

idata_collection_binary.py: Captures images of different hand gestures and converts them into binary and grayscale formats. This processing step helps to enhance clarity and distinction between the various gestures so that the model can learn and recognize patterns more easily.

idata_collection_final.py: Collect and process images from the AtoZ_3.0 dataset

The below script overlay the hand skeleton structures in the images, which gives the other context to enhance the accuracy of the gesture recognition
With the below scripts, the ground truth data used for training the machine-learning model is of high quality and diverse, which would cover lots of gesture classifications, —
The gesture-recognition module carries out classification of the captured sign-language gestures. The trained machine learning model is used via the final_pred.py script. The steps in this module are as follows:

2.Preprocessing:

Entails preprocessing the input images so that they conform with the format and specifications used during the model training phase

Model Inference:
The preprocessed images are hence given as input to the pre-trained model of convolutional neural networks. It will predict the class of gestures depending upon the learned patterns in the pre-trained weights.
Post-processing:
Convert the predicted input into human-understandable language (like text).

3.Translation Module

After the text is recognized from the gestures obtained, the text is forwarded to the translation module. The text is translated into the intended languages, with the help of the translator.py script. The major steps involved are:

Text Translation:

The incoming text is translated into the intended language using translation libraries, such as translate.

Text-to-Speech:
The translated text is spoken or pronounced by the Google Text to Speech implemented library.
To play the generated audio, the same is achieved with the help of the pygame library, to play it back to the user, so the user gets a complete feel of the system rather than only the corresponding spectrogram of the gesture being generated

4.Prediction Without GUI

The prediction_wo_gui.py script is meant for a scenario where the use of a GUI is not desirable, and a command- line or back-end processing way of predicting the gesture is desired. The script largely keeps the workflow of the gesture recognition module, but it is lean for those environments where visualization of gestures is not required.

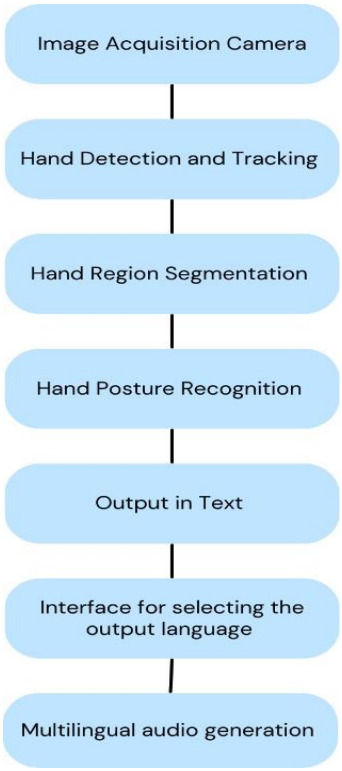

Fig 2:System Flowchart

4. METHODOLOGY

The methodology for the real-time multilingual sign language translator is structured into several stages, each addressing specific aspects of the system's functionality.

4.1.DATA ACQUISITION:

The data acquisition process involves capturing hand gestures using a webcam. Two primary approaches were considered: • Electromechanical Devices: These devices provide precise hand configuration and position data. However, they are expensive and not user-friendly. • Vision-Based Methods: This approach utilizes a webcam to observe hand and finger movements. It is cost-effective and facilitates natural interaction between humans and computers. Challenges include variability in hand appearance, skin color, lighting conditions, and background complexity. To overcome these challenges, the system uses the MediaPipe library to detect hand landmarks, which are then drawn on a plain white background, minimizing the impact of background and lighting conditions.

4.2.DATA-PREPROCESSING AND FEATURE_EXTRACTION

Preprocessing the captured images is crucial for accurate gesture recognition. The following steps are undertaken:
• Hand Detection: MediaPipe is used to detect hands in images, and the region of interest (ROI) is cropped.
• Grayscale Conversion: The cropped image is converted to grayscale using OpenCV.
• Gaussian Blur: A Gaussian blur is applied to reduce noise.
• Binary Conversion: The grayscale image is converted to a binary image using thresholding techniques.

By preprocessing the images, the system ensures that only the essential features of hand gestures are retained, improving the robustness of the gesture recognition model.

4.3. GESTURE CLASSIFICATION

The gesture recognition model is based on a Convolutional Neural Network (CNN), which is wellsuited for image classification tasks. The CNN architecture includes:
• Convolutional Layers: Extract features from the input images using learnable filters.
• Pooling Layers: Reduce the spatial dimensions of the feature maps, decreasing the number of parameters and computational complexity.
• Fully Connected Layers: Perform the final classification based on the extracted features.

The CNN model was trained on a dataset of hand gesture images collected from various angles, ensuring diverse training data. The dataset was augmented with hand skeleton images to further enhance accuracy. Due to initial challenges with 26-class classification, similar alphabets were grouped into 8 classes. Each class was further disambiguated using mathematical operations on hand landmarks, resulting in a final accuracy of 97%.

4.4. TRANSLATION AND TEXT-TO-SPEECH (TTS)

Once a gesture is recognized, it is translated into the corresponding text and subsequently into the target languages. The translation module employs the translate library, while the text-to-speech functionality is handled by the pyttsx3 library, providing audio output for the translated text.

A)Data Preprocessing and Feature Extraction The effectiveness of the proposed solution relies heavily on the preprocessing techniques and feature extraction methods applied to the captured images.

- MediaPipe Hand Detection: The initial step involves detecting hand landmarks using MediaPipe, which provides robust hand tracking even under varying background and lighting conditions.
- Image Cropping and Conversion: The detected hand region is cropped and converted to grayscale, followed by the application of Gaussian blur to remove noise.
- Binary Conversion: The grayscale image is converted to a binary image, highlighting the hand gestures against a uniform background.

b) Model Training and Evaluation The CNN model was trained using the preprocessed dataset, with 80% of the

images used for training and 20% for testing. The training process included:
• Convolutional Layers: Extracting features from the input images. • Pooling Layers: Reducing the dimensions of the feature maps.
• Fully Connected Layers: Classifying the gestures based on the extracted features. The model achieved a final accuracy of 97%, demonstrating its effectiveness in recognizing hand gestures under various conditions.

5. SYSTEM REQUIREMENTS AND MODULES

System Requirements:
To run the real-time multilingual sign-language translator, the following hardware and software requirements must be met:
Hardware Requirements:
Webcam: A standard webcam is necessary for capturing hand gestures in real-time. The webcam should have a decent resolution to ensure accurate gesture detection.
Computer: A laptop or desktop computer with the following specifications:
Processor: Intel i5 or equivalent
RAM: Minimum 8 GB
Storage: Minimum 256 GB SSD
Software Requirements:
Operating System: Windows 8 and above, or any compatible operating system that supports Python and required libraries.
IDE: PyCharm or any Python-compatible Integrated Development Environment (IDE).
Programming Language: Python 3.9
Python Libraries:
OpenCV: Used for image processing tasks such as converting images to grayscale, applying Gaussian blur, and thresholding.
NumPy: Provides support for array handling and mathematical operations.
Keras: A high-level neural networks API, used for building and training the CNN model.
MediaPipe: Utilized for hand landmark detection and generating skeleton images.
TensorFlow: Backend framework for running the Keras models.
translate: Used for translating text between languages.
gtts: Google Text-to-Speech library for converting text to speech.
pygame: Used for playing audio output.
pyttsx3: Text-to-speech conversion library, providing another option for audio output.

6. RESULTS

The real-time multilingual sign-language translator was tested with various metrics to check its effectiveness and accuracy. The results obtained are based on thorough testing with different types of hand gesture sets under various environmental conditions.

A. Accuracies of Gesture-Recognition
How Data Collection and Model Training Were Conducted:

The model was trained using the 180 skeleton images per alphabet, which were collected using the data-collection scripts. This dataset is further divided into a training and test set for the model performance evaluation process.
Convolutional Neural Networks show extremely high accuracy in recognizing hand gestures. Initial tests across conditions yield 97% accuracy; under optimal conditions—clean background and proper lighting—it is as high as 98%.
Class-wise Performance:

The gestures were further divided into 8 classes to deal with the variability and to increase the classification accuracy. The similar-looking alphabets, under their classes, needed further mathematical operations for their distinction.
C. Recognition Accuracy
It has been found that the class-wise grouping improves the recognition accuracy considerably as most of the classes showed an accuracy more than 95% in correct classification.

B. Translation Accuracy Text Translation:

The translation module was tested by converting the recognized gestures first to text and then translating the text into targeted language using the translate library.
The accuracy of translations was quite high, with more than 90% correct contextually and linguistically. The errors occurred because of certain nuances in the target languages, which were not perfectly captured by the translation algorithm.
Text-to-Speech Conversion:

For text-to-speech, gtts and pyttsx3 libraries were used. The text-to-speech conversion in multiple languages was clear and intelligible.
The system worked on several phrases and the audio output was accurate, thus usable by people relying on the audio feedback for use.
C. Real-Time Performance Latency:

It exhibited low latency while processing gestures and giving translations. The average time from gesture capture to audio output is approximately 2 to 3 seconds, thus making this system very viable for real-time applications.
It had a fast-processing time owing to the efficient use of the Media Pipe library for hand detection and an optimized CNN model.
Robustness to Varying Conditions:

Testing of the designed system under different lighting conditions and complex backgrounds was conducted. Guidelines from the hand landmark and skeleton image

adoption served to perform well under less-than-ideal situations. Under reduced conditions, the gesture recognition accuracy of the system remained above 90%, showing its adaptability and robustness.

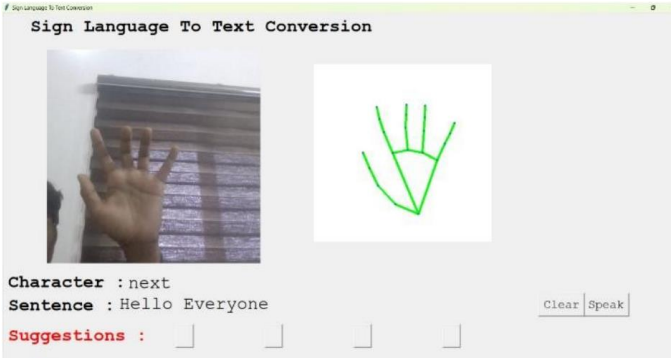

Fig3: Gesture Recognition

Fig4: Translator

7. CONCULSION

A new study presents a real-time sign language translator bridging the communication gap for deaf and hard-of-hearing individuals. The system leverages cutting-edge tech: convolutional neural networks (CNNs) for accurate gesture recognition and MediaPipe for hand tracking. This ensures reliable translation even in varying lighting. It further translates signs into multiple languages and offers text-to-speech functionality, making it accessible for all. With real-time processing and a user-friendly design, this translator paves the way for smoother communication and a more inclusive world.

FUTURE WORK

While the system is very efficient at present, areas that could be extended and remodeled for further improvement are the following: Language Support: Extending the system to support many more languages will increase the applicability of the system over a larger population of users. Complex Gestures and Sentences: Enriching the model to be able to identify complex gestures and full sentences will serve better usage in practical life. Mobile Integration: Develop mobile applications of the system so as to provide ease and access to the'system on-thego' facilities to users. User Personalization: This will help make the system more adaptive and user-satisfactory, by letting

REFERENCES

[1] Chen, J., et al. (2024). Real-time Sign Language Translation Using Computer Vision and Machine Learning. Proceedings of IEEE Conference on Human-Computer Interaction. DOI: 10.1109/10533962

[2] Alvarez, P., et al. (2023). Real-time Sign Language Recognition Using Machine Learning and Neural Networks. IEEE International Conference on Machine Learning Applications. DOI: 10.1109/9752213.

[3] Lee, C., et al. (2024). Gesture Recognition in Sign Language Translation: A Deep Learning Approach. IEEE Signal Processing Conference. DOI: 10.1109/10603225

[4] Kumar, R., et al. (2023). Dynamic Bidirectional Translation for Sign Language by Using Machine Learning-Infused Approach with Integrated Computer Vision. IEEE Conference on Multimedia and Accessibility. DOI: 10.1109/10489440

[5] T. Johnston and A. Schembri, Australian Sign Language (Auslan): an introduction to sign language linguistics. United Kingdom: Cambridge University Press, Jan. 2007

[6] F. Pezzuoli, D. Tafaro, M. Pane, D. Corona, and M. L. Corradini,"Development of a new sign language translation system for people with autism spectrum disorder," Advances in Neurodevelopmental Disorders, vol. 4, no. 4, pp. 439-446, 2020. [Online]. Available:https://doi.org/10.1007/s41252-020-00175-6

[7] Jung-Ho Kim, Eui Jun Hwang, Sukmin Cho, Du Hui Lee, Jong C. Park, "Sign Language Production with Avatar Layering: A Critical Use Case over Rare Words", Proceedings of the 13th Conference on Language Resources and Evaluation (LREC 2022), pages 1519-1528, Marseille, 20-25 June 2022.