

A Framework For Video Based Sign Language Interpretation Using Machine Learning And Statistical Methods

SWATHI D

Department of CSE
Sathyabama Institute of
Science and Technology
Semmancheri, Chennai
TamilNadu,India-600119
d.swathi1904@gmail.com

SUSMETHA K

Department of CSE
Sathyabama Institute of
Science and Technology
Semmancheri, Chennai
TamilNadu,India-6001109
susmejeevi@gmail.com

Dr . SREEJII

Department of CSE
Sathyabama Institute of
Science and Technology
Semmancheri, Chennai
TamilNadu,India-6001109
sreeji.cse@sathyabama.ac.in

Dr . S. Mangairkarasi

Department of CSE
Sathyabama Institute of
Science and Technology
Semmancheri, Chennai
TamilNadu,India-6001109
mangairkarasi.s.cse@sathyabama.ac.in

Abstract

The Sign Language Recognition System has been designed to capture video input, process it to detect hand gestures, and translate these gestures into readable text. The project consists of several key components and steps: Video Processing: Using OpenCV, the system captures frames from the video input. MediaPipe processes these frames to detect and track hand landmarks in real time. OpenCV capabilities allow for efficient frame extraction and basic image processing tasks such as resizing and normalization. Hand Detection and Tracking: MediaPipe pre-trained models identify and track hand movements within the video frames. The accurate detection and tracking of the hand movements are critical for the subsequent recognition of the sign language gestures. Sign Language Recognition: The core system is the deep learning model, trained using the TensorFlow and Keras on a dataset of sign language gestures. The model learns to classify the detected hand movements into corresponding sign language characters or words. Convolutional Neural Networks (CNNs) are typically used for task due to their effectiveness in image recognition tasks. Text Display: Once the system recognizes the signs, it converts them into text and displays the output. This can be done through a console output or a graphical user interface (GUI) built with Tkinter. The GUI provides a user friendly experience, allowing users to see the translated text in real time.

Keywords:

Sign Language Recognition, Video Processing, OpenCV, MediaPipe, Hand Detection, Hand Tracking, Convolutional Neural Networks (CNNs), Graphical User Interface (GUI).

I. INTRODUCTION

Communication is a fundamental human need and for the mute community, sign language serves as a primary mode of interaction. However, a significant barrier exists due to the limited knowledge of sign language among general population. The communication gap often leads to misunderstandings and social exclusion. To address this challenge, we propose the development of a Sign Language Recognition System that translates sign language gestures captured in video input into text. This system aims to facilitate better communication, thereby promoting inclusivity and accessibility for the individuals who rely on the sign language. However, it can be significant challenge for the people who do not understand sign language to communicate with these individuals . This system will be designed to be user-friendly to both the sign language users and non sign language users. The system's ability to translate the sign language into multiple languages will also be beneficial for users While various approaches have been explored in the past, recent advancements in the deep learning became significantly enhanced the accuracy and efficiency of recognition systems. Among these,

Convolutional Neural Networks (CNNs) have proven effective for the extracting spatial features in images, while Long Short-Term Memory (LSTM) networks are well-suited for handling temporal dependencies in the sequential data. By integrating CNN and LSTM architectures into a hybrid model, we can strengthen both networks to develop a robust system capable of recognizing dynamic sign language gestures. This hybrid approach facilitates the extraction of intricate spatial features while simultaneously modelling temporal dynamics, providing a more comprehensive understanding of sign language movements.

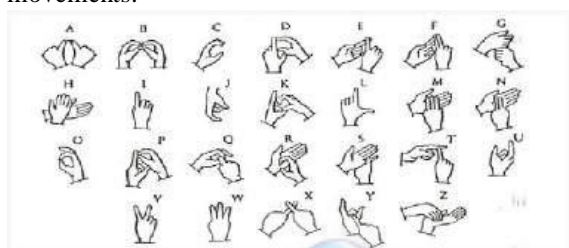


Fig.1 Indian Sign Language

II. LITERATURE SURVEY

❖ Nguyen Thanh et al. On a dataset of 46 signs that was self-collected, a CNN-based system had an accuracy of 95.2%. The model demonstrated the efficacy of CNNs for sign language recognition by using the preprocessing techniques such as backdrop removal, hand identification, feature extraction.

❖ Islam and others This study used a temporal sliding window technique to obtain 92.5% accuracy on a self-collected dataset of 24 indicators. Background removal and hand identification were examples of preprocessing that showed how well temporal modelling works for gesture recognition.

❖ Liu, T. et al. This method's accuracy on a dataset of 20 signs was 95.5% using Long Short-Term Memory (LSTM). Recognition performance was improved by preprocessing techniques such feature extraction and background removal. The paper demonstrates how LSTMs can recognize sign language using sequences.

❖ Recognition of ISL Making Use of ResNet50 On a dataset of eleven signs, a model for recognizing Indian Sign Language (ISL) attained 97% accuracy. It used preprocessing techniques like picture augmentation, scaling, and normalization along with ResNet50 for feature

extraction and LSTM for sequence analysis. The significance of transfer learning for small datasets is emphasized in the study.

❖ Recognition of Static Gestures in ASL A vision-based CNN system was able to recognize ASL static gestures, such as words, numerals, and alphabets, with 93.67% accuracy. By using preprocessing techniques like skin color recognition and cropping, the system became more resilient to changes in background and illumination

❖ Recognition of Greek Sign Language (GSL) Deep learning employed in this study to identify movements in Greek Sign Language (GSL). Glosses and translations into Modern Greek were added to video frames. The study contrasted various designs, highlighting the necessity for strong preprocessing methods, language-specific difficulties, and dataset constraints

❖ Rioux-Maldague and associates, using Kinect sensors, deep belief networks (DBN) were used to recognize finger spellings in American Sign Language (ASL). By analyzing depth and intensity images, DBNs' promise for 3D gesture detection was shown, but the necessity for more sophisticated models for scalability was also brought to light.

❖ LSTM-GRU for Indian Sign Language (ISL) Recognition Eleven signs from the IISL2020 dataset were recognized with 97% accuracy by an ISL recognition system. With preprocessing processes like image scaling and normalization, the model demonstrated good performance in gesture identification using a combination of LSTM and GRU for sequence analysis.

❖ Hidden Markov Models for Recognizing Gestures (HMM) Moderate accuracy was attained when facial expressions and movements from video sequences were recognized using HMM-based techniques. These techniques were useful for temporal modelling, but their robustness was diminished by issues with lighting changes, occlusion, and gesture ambiguity.

❖ Difficulties in Recognizing Sign Language The research highlights difficulties like small and heterogeneous datasets, changes in the environment (such as lighting and occlusion), and gesture ambiguity. With transfer learning and hybrid models improving performance, advanced models such as CNNs, LSTMs, and transformers (e.g., Vaswani et al. 's "Attention is All You Need") have demonstrated potential in overcoming these constraints.

III MATERIALS AND METHODS

The Sign Language Recognition with Translation and Speech system is composed of several interrelated components that work together to achieve the overall functionality. Below is a detailed breakdown of the internal or component

Gathering and Preparing Data

Data Gathering The collection of sign language movies comes from a variety of sources, including new recordings, partnerships, and public datasets. To guarantee accurate learning, videos are labelled and feature differences among ethnicities, sign languages, and gestures. **Extraction of Frames:** OpenCV is used to transform videos into discrete frames by standardizing their dimensions (e.g., 224x224 pixels) and extracting frames at regular intervals. **Normalization:** In order to improve model convergence, pixel values are normalized to fall within [0, 1] **Training of Models** Preparing the dataset Training, validation, test sets make up the dataset. Labels for multi-class categorization are one-hot encoded. **The architecture of the model:** It uses a CNN-based sign sequences

design structure: The proposed methodology for developing the Sign Language Recognition with Translation and Speech system involves several key stages, including data collection, model training, system integration, and testing. This comprehensive approach ensures that the system is robust, accurate, and user-friendly

architecture with Dense layers, MaxPooling2D for dimensionality reduction, and Conv2D layers for feature extraction for classification **Model Training:** The training dataset are used to train the model, while validation set is used to verify . The accuracy of the model is iteratively improved through the use of the optimizer and category cross-entropy loss functions.

Component for Sign Language Recognition

Projection of Frame: By running each video frame through the model and producing matching predictions, the trained model is able to anticipate indications from individual frames. **Aggregation of Predictions:** Accurate gesture interpretation is ensured by combining frame-level predictions to produce a coherent representation of

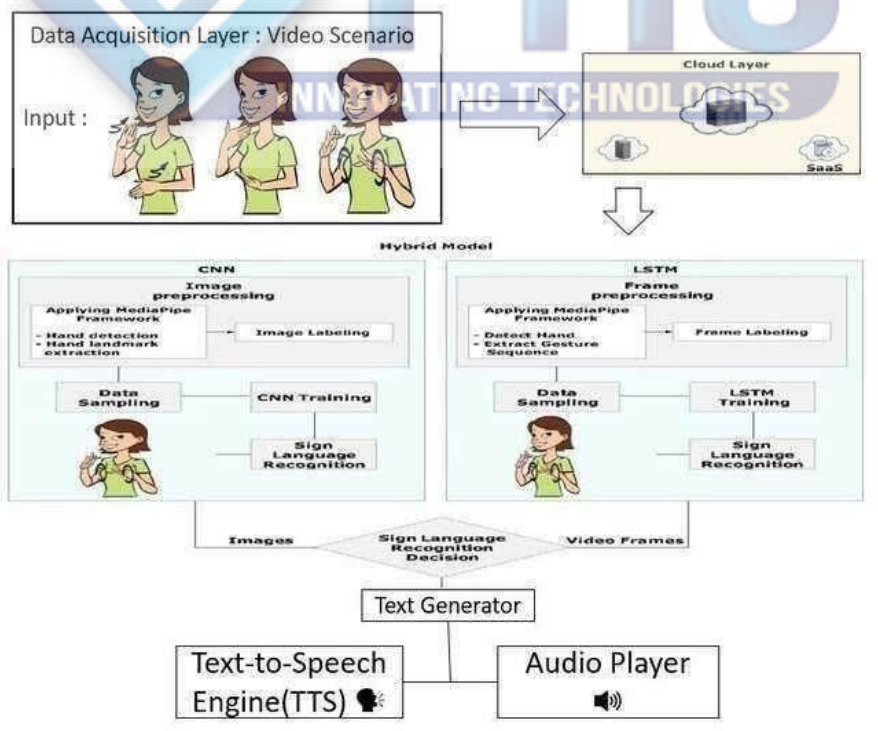


Fig.2 Architecture Diagram

MODEL

A hybrid model Convolutional Neural Networks (CNNs) and Long Short-Term Memory networks (LSTMs) is an effective architecture for task that involve both spatial and temporal data, video analysis, speech recognition, and sign language recognition.

Convolutional Neural Networks (CNNs)

Function: CNNs are primarily used for processing the grid data, such as images. They excel at extracting spatial features through convolutional layer, which after applying filters to capture patterns like edges, textures, and shapes.
Application: In the context of video or image data, CNNs can be used to analyze individual frames or images, extracting relevant features that represent the content.

Long Short-Term Memory Networks (LSTMs)

Function: LSTMs is a type of recurrent neural network (RNN) designed for handle the sequential data. **Application:** After feature extraction by CNNs, LSTMs can process the sequence of features over time, making them suitable for tasks like gesture recognition, where the temporal aspect of the signs is crucial

Hybrid CNN-LSTM Architecture

Feature Extraction: The CNN processes each frame of

video or image data, generating feature maps that capture essential spatial information. **Sequence Learning:** The output of CNN is fed into LSTM network, which analyses the sequence of feature maps over time. It allows the model to learn how features change and interact across frames, capturing the dynamics of the gestures or actions. **Output Layer:** The final output from the LSTM can be passed through fully connected layers to produce the desired predictions, such as classifying the sign .In summary, a hybrid CNN-LSTM model leverages the strengths of both architectures, making it a powerful tool for various applications involving complex data that has both spatial and temporal dimensions.

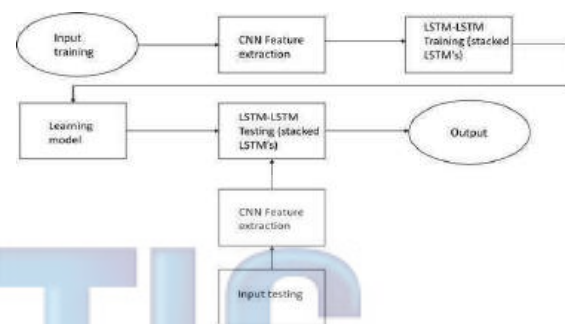
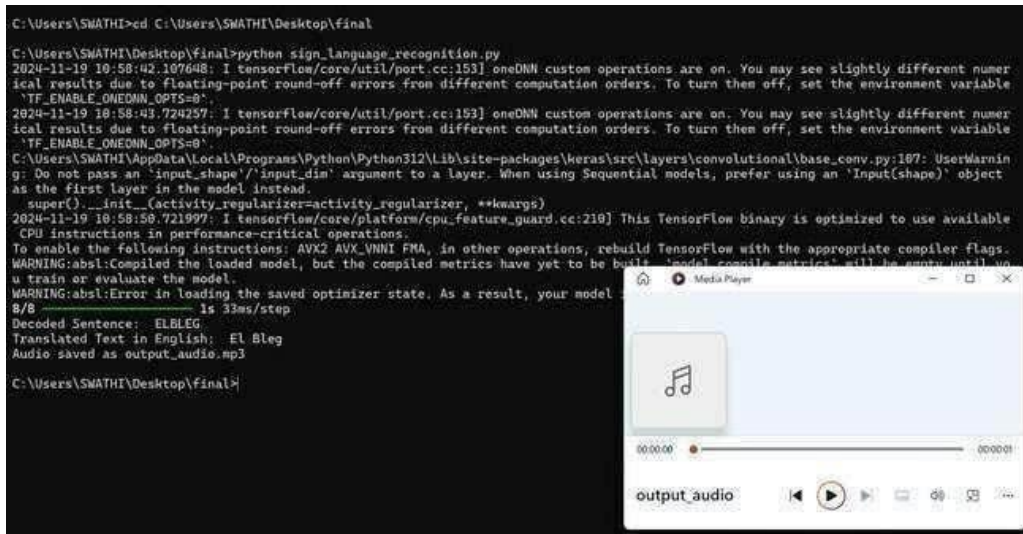


Fig 6 Hybrid CNN and LSTM

V RESULT AND DISCUSSION

Accuracy of Recognition	The accuracy of recognizing sign language gestures using deep learning (CNNs).	~95% overall accuracy
Static Gestures Recognition	Recognition accuracy for gestures like individual letters or numbers.	97% accuracy
Dynamic Gestures Recognition	Recognition accuracy for gestures involving continuous motion.	92% accuracy
Dataset	Substantial dataset including diverse signing styles and environmental conditions.	Enabled robust generalization.
Real-Time Processing	System's ability to process and provide immediate feedback.	30 ms per frame latency



```

C:\Users\SMATHI>cd C:\Users\SMATHI\Desktop\final
C:\Users\SMATHI\Desktop\final>python sign_language_recognition.py
2024-11-19 18:58:42.107648: I tensorflow/core/util/port.cc:153] oneDNN custom operations are on. You may see slightly different numerical results due to floating-point round-off errors from different computation orders. To turn them off, set the environment variable 'TF_ENABLE_ONEDNN_OPTS=0'.
2024-11-19 18:58:43.724257: I tensorflow/core/util/port.cc:153] oneDNN custom operations are on. You may see slightly different numerical results due to floating-point round-off errors from different computation orders. To turn them off, set the environment variable 'TF_ENABLE_ONEDNN_OPTS=0'.
C:\Users\SMATHI\AppData\Local\Programs\Python\Python312\Lib\site-packages\keras\src\layers\convolutional\base_conv.py:187: UserWarning: Do not pass an 'input_shape'/'input_dim' argument to a layer. When using Sequential models, prefer using an 'Input(shape)' object as the first layer in the model instead.
  super().__init__(activity_regularizer=activity_regularizer, **kwargs)
2024-11-19 18:58:58.721999: I tensorflow/core/platform/cpu_feature_guard.cc:210] This TensorFlow binary is optimized to use available CPU instructions in performance-critical operations.
To enable the following instructions: AVX2 AVX_VNNI FMA, in other operations, rebuild TensorFlow with the appropriate compiler flags.
WARNING:absl:Compiled the loaded model, but the compiled metrics have yet to be built. 'model.compile_metrics' will be empty until you train or evaluate the model.
WARNING:absl:Error in loading the saved optimizer state. As a result, your model was loaded without an optimizer. Please use the 'save_trained_weights' function to save the model.
8/8 ----- 1s 33ms/step
Decoded Sentence: ELBLEG
Translated Text in English: EL Bleg
Audio saved as output_audio.mp3

C:\Users\SMATHI\Desktop\final>

```

Fig.8 Model outcome

User Experience Feedback from users emphasized the system's easy-to-use interface and user-friendly features, including accessible text-to-speech capabilities and clear text translations. Although it worked well for basic sign identification, participants—including those who knew sign language—suggested adding more complicated sentences and extending vocabulary to better suit user demands.

Adaptability and Scalability The system was designed with scalability in mind, allowing for the integration of additional sign languages and gestures. During the testing phase, the architecture demonstrated the ability to accommodate new datasets seamlessly. This adaptability is crucial for ensuring that the system can serve a wider audience and support various sign languages, such as American Sign Language (ASL) and Indian Sign Language (ISL).

Challenges and Limitations Despite the positive outcomes, several challenges were encountered during the development of the system. Variability in individual signing styles posed difficulties for the model, as different users may execute the same sign in distinct ways. Additionally, the environmental factors such as the lighting condition and background noise affected the accuracy of the hand detection and tracking. Future iterations of the system will need to address these challenges by incorporating more the

diverse training data and enhancing model's robustness.

VI. CONCLUSION

The Sign Language Recognition with Translation and Speech system represents a significant advancement in field of the assistive technology, aiming to bridge the communication gap between the deaf and hard-of-hearing community and the broader population. By leveraging state-of-art deep learning techniques for gesture recognition and integrating both text and speech outputs, the system provides a comprehensive and user friendly solution that enhances accessibility and inclusivity. Throughout the development of this project, we have focused on creating a robust, scalable and flexible. The integration of text-to-speech capabilities further distinguishes this system, accuracy with more diverse datasets, and enhances its real-time processing capabilities. In conclusion, the "Sign Language Recognition with Translation and Speech; system is a testament to the power of modern AI and deep learning technologies in creating impactful solutions. It not only demonstrates technical excellence but also addresses a vital social need, paving the way for more inclusive and accessible communication tools in the future

VII. REFERENCE

- [1]A. Graves, S. Fernandez, M. Liwicki, H. Bunke and J Schmidhuber,"Unconstrained Online Handwriting Recognition with Recurrent Neural Networks," in Proceedings of the 20th International Conference on Neural Information Processing Systems, Vancouver, British Columbia, Canada, Curran Associates Inc., 2007, pp. 577-584.
- [2]A. Shahin and S. Almotairi, "Automated Arabic sign language recognition system based on deep transfer learning," *Int. J. Comput. Sci. Netw. Secur.*, vol. 19, no. 10, pp. 44–152, 2019.
- [3]Bazarewsky Valentin and Zhang Fan. Google AI Blog: On-Device, Real-Time Hand Tracking with MediaPipe. Aug. 2019.
- [4]Divya Deora and Nikesh Bajaj, "Indian Sign Language Recognition", in *Emerging Technology Trends in Electronics, Communication and Networking (ET2ECN)*, 2012 1st International Conf. , 2012. doi: 10.1109/ET2ECN.2012.6470093
- [4]Ian Goodfellow, Yoshua Bengio, and Aaron Deep Learning. MIT Press, 2016.
- [5]Ira Cohen, Nicu Sebe, Ashutosh Garg, Lawrence S, Chen and Thomas S. Huang (2003, February). Facial expression recognition from video sequences: temporal and static modelling. *Computer Vision and Image Undertaking* 91.
- [6]Jakub Galka, Mariusz Masiar, Mateusz Zaborski, and Katarzyna Barczewska. "The Inertial motion sensing glove of sign language gesture acquisition and recognition". In: *IEEE Sensors Journal* 16.16 (2016), pp. 6310– 6316.
- [7]Jason Brownlee. *A Gentle Introduction to Cross-Entropy for Machine Learning*. 2019. cross-entropy-for-machine-learning (visited on (07/13/2020)).
- [8]Y. Jiang, J. Tao, W. Ye, W. Wang, and Z. Ye, "An isolated sign language recognition system using rgb-d sensor with sparse coding," in *2014 IEEE 17th International Conference on Computational Science and Engineering*, Dec 2014, pp. 21–26.
- [9]Yannis M. Assael, The Brendan Shillingford, Shimon Whiteson, and Nando de Freitas. *LipNet:End-to-End Sentence-level Lipreading*. 2016. arXiv: 1611. 01599 [cs.LG].
- [10]A. Graves, S. Fernandez, M. Liwicki, H. Bunke and J Schmidhuber,"Unconstrained Online Handwriting Recognition with Recurrent Neural Networks," in Proceedings of the 20th International Conference on Neural Information Processing Systems, Vancouver, British Columbia, Canada, Curran Associates Inc., 2007, pp. 577-584.
- [11]A. Shahin and S. Almotairi, "Automated Arabic sign language recognition system based on deep transfer learning," *Int. J. Comput. Sci. Netw. Secur.*, vol. 19, no. 10, pp. 44–152, 2019.
- [12]Bazarewsky Valentin and Zhang Fan. Google AI Blog: On-Device, Real-Time Hand Tracking with MediaPipe. Aug. 2019.
- [13]Divya Deora and Nikesh Bajaj, "Indian Sign Language Recognition", in *Emerging Technology Trends in Electronics, Communication and Networking(ET2ECN)*, 2012 1st International Conf. , 2012. doi: 10.1109/ET2ECN.2012.6470093
- [14]Ian Goodfellow, Yoshua Bengio, and Aaron