

Sign Language Interpreter Using Machine Learning

Aditya Thakur

Department of computer science and engineering, Galgotias University, Greater Noida, Uttar Pradesh, India adityathakur01201@gmail.com.

ABSTRACT

As technology advances each year, humanity continues to develop innovative tools aimed at improving lives, especially for individuals with disabilities. Our project focuses on easing communication for people with speech impairments by proposing a sign language interpreter that automatically converts sign language into text output. For those unable to speak, sign language serves as a primary mode of expressing ideas and emotions. However, this form of communication is challenging for many people to understand, making interaction difficult.

Recognizing sign language accurately is complex due to the variety of hand movements and gestures involved, and achieving high accuracy at a low cost is a significant challenge. Our solution proposes a system that leverages deep learning techniques, particularly Convolutional Neural Networks (CNNs), to recognize sign language gestures accurately. By using pre-trained hand-tracking models like MediaPipe, the system ensures robust detection in various environments

Keywords- Sign language interpreter, speech impairments, text output, hand gestures, Convolutional Neural Networks (CNN), MediaPipe

I. INTRODUCTION

Sign language is the primary mode of communication for individuals who cannot speak, using gestures to express messages. It combines hand shapes, orientations, movements of the hands, arms, or body, along with facial expressions and lip patterns. Common forms of sign language include American Sign Language (ASL), British Sign Language (BSL), and Indian Sign Language (ISL) [1]. According to the 2011 census, India has 1,998,535 individuals with speech disabilities [2]. Communication is an exchange of thoughts and feelings and should not be limited by words or speech. Sign language interpreters (SLI) are essential for converting sign language into spoken language [1]. In recent years, sign interpreters have become an important area of research and development, with various hand-movement tracking technologies emerging that differ in terms of accuracy, jitter, drift, and latency [3]. Generic automated methods for sign interpretation often rely on sensors, like optical fibres and flex sensors, to track hand and finger movements, mapping them to specific letters. Each country has a unique sign language, and signs for identical words can vary internationally [4]. While substantial research has been conducted on sign language interpreters in countries such as the USA, UK, Singapore, and Canada, India has seen minimal progress

in this field. Existing products in India struggle to match international standards in terms of user-friendliness.

II. LITERATURE REVIEW

Many studies have explored sign language recognition using various sensors and machine learning techniques. Here's an overview of recent work in this area.

Thakur et al. [25] proposed a real-time sign language detection system using Convolutional Neural Networks (CNNs) to produce spoken output. Their dataset primarily included the American Sign Language alphabet, and gestures were trained using the CNN VGG-16 model with Python tools such as OpenCV and skimage. The system achieved training loss and accuracy of 0.0259 and 99.65%, respectively, with 99.62% accuracy in testing, demonstrating the system's effectiveness in generating speech from detected gestures. This research highlights CNN's potential for real-time sign language interpretation, paving the way for further work to expand the system to more languages and gestures.

In a separate study, Shrenika and Madhu Bala [26] used a template matching algorithm for sign language recognition. This approach involved recording various hand gestures with a camera and then processing the images with algorithms, starting with pre-processing and edge detection to outline each sign's edges. The

template matching algorithm recognized the sign, displaying the corresponding text. The system effectively recognized basic static hand signs, suggesting template matching as a viable method for sign language recognition.

III. METHODOLOGY

The system's development involves five key stages, illustrated in Fig 1: data acquisition, image pre-processing, model training, model evaluation, and model deployment, where real-time predictions and results are generated. This chapter details the system's design specifications and the step-by-step approach to building the system.

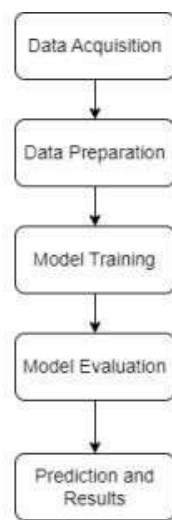


Fig. 1. Block Diagram representing the System Design

A. Data Acquisition

A dataset of images depicting static ASL alphabet signs was created. To capture the image dataset, a laptop webcam and the Python OpenCV library were used to record the signer's hand gestures. The HandDetector module from evzone helped isolate and crop the signer's hand, reducing the impact of background noise on model accuracy. Each hand image was then resized with OpenCV to a consistent 300 x 300 pixels. Dynamic signs "J" and "Z" were excluded, resulting in 24 image classes and a total of approximately 8,931 images, saved in JPG format.

B. Data Preparation

At this stage, the sign images underwent pre-processing using resizing and normalization techniques. Each RGB image was resized to 224x224 pixels to fit the model requirements, utilizing OpenCV's built-in functions. For

normalization, the 'to_categorical' function from Keras transformed the datasets into one-hot encoded vectors. Pixel intensity values were then scaled by dividing by 255, adjusting the range to achieve a mean of 0 and variance of 1, ensuring consistency in pixel value scales across the images.

C. The CNN Architecture

The model architecture was developed using the Keras framework, leveraging its flexibility and efficiency for constructing deep learning models. To optimize the hyperparameters of the Convolutional Neural Network (CNN), the validation dataset was employed. The input shape for the images was defined as (224, 224, 3), which signifies that the images have a resolution of 224x224 pixels and incorporate three color channels: Red, Green, and Blue (RGB). Utilizing RGB images provides crucial color information, enhancing the model's ability to differentiate between various signs and thereby improving the accuracy of sign language recognition.

The structure of the model consists of three convolutional layers, each designed to extract important features from the input images. These layers employ Rectified Linear Unit (ReLU) activation functions, which are widely used for introducing non-linearity into the model while maintaining computational efficiency. Additionally, a max pooling layer follows each convolutional layer, which helps to reduce the spatial dimensions of the feature maps. This pooling process not only decreases the computational load but also aids in retaining the most prominent features, contributing to better overall performance and robustness in recognizing sign language. By meticulously designing this architecture, the model aims to achieve high accuracy and reliability in real-time sign language interpretation.

D. Model Training

The model underwent training on the training dataset for a duration of 5 epochs with a batch size of 64. This number of epochs was chosen because the dataset was relatively small, so the smaller number of epochs was sufficient to achieve reasonable performance while minimizing the training time. The decision was also made to prevent overfitting to the training data which can be caused by prolonged training, resulting in poor performance on new and previously unseen data. The validation dataset was used to obtain optimal settings for the CNN hyper-parameters

E. Model Evaluation

To evaluate the model's performance on new, unseen data, the test dataset was utilized. A range of metrics—such as accuracy, precision, recall, and F1-score—were calculated using the test data to measure the model's effectiveness and reliability in sign language recognition.

F. Model Deployment

The model was implemented for real-time testing. The application utilized the device's camera to capture a live video feed and displayed the model's predictions on the screen. For effective sign detection, it is essential that the user's background is free of clutter and that there is adequate lighting. Once the hand is identified, the application will produce the corresponding text representation of the recognized sign.

IV. RESULTS

The overall training accuracy was determined to be 99.86% while validation accuracy was determined to be 99.94%. The test accuracy was determined to be 94.68%. Figure 5 exhibits the curves for training and validation accuracy and Fig 6. depicts the training and validation loss over 5 epochs. A confusion matrix and the calculated test accuracy, precision, recall, f1- score were obtained based on the test dataset using scikit learn library in Python. Accuracy is the most effective performance metric and is computed as the ratio of correct predictions to all predictions.

$$\text{Accuracy} = \frac{\text{number of correct predictions}}{\text{number of all predictions}} = \frac{TP + TN}{TP + FP + FN + TN}$$

V. CONCLUSION

The system was successfully developed, implemented, tested, and deployed. This paper highlighted the importance of sign language recognition systems for individuals who are deaf or hearing-impaired. It demonstrated the capability to accurately recognize sign language gestures in real-time. However, enhancing the model's effectiveness and reliability may require additional data collected under various environmental conditions. Currently, the system is limited to recognizing static ASL alphabet gestures and does not account for the dynamic and continuous aspects of sign language, which involve complex movements, facial expressions, and body language.

Future research should focus on extending the system's capabilities to recognize dynamic gestures for a more comprehensive mode of communication. The system has been specifically trained and evaluated on the ASL alphabet, but since sign languages differ across regions and countries, each possessing distinct vocabulary and grammar, the model's performance with other sign languages remains untested. Further exploration and adaptation are necessary to ensure its applicability across different sign languages.

VI. REFERENCES

The heading of the References section must not be numbered. All reference items must be in 8 pt font. Please use Regular and Italic styles to distinguish different fields as shown in the References section. Number the reference items consecutively in square brackets (e.g. [1]).

[1] Nair, A.V. and Bindu, V., "A review on Indian sign language recognition", *International Journal of Computer Applications*, 73(22), 2013.

[2] Sailee Brahme, "READING THE SIGNS: SIGNS LANGUAGE", April 21, 2017, available online at <http://forreadingaddicts.co.uk/culture/reading-signs-signlanguage/> 18395

[3] Republic World, Press Trust of India, "India's First Sign Language Dictionary Released", Mumbai, March 23, 2018

[4] Lucy Sweeney, "Gloves that convert sign language to speech 'to empower the deaf community'", *ABC News*, April 2016.

[5] Divyangjan, "Indian Sign Language Research and Training Centre (ISLRTC), Department of Empowerment of Persons with Disabilities, Ministry of Social Justice & Empowerment, Government of India, online available at <http://205.147.97.190/islrtcapp/>

[6] Sanjay Kumar, "First Indian Sign Language Dictionary of 3000 Words", *Press Information Bureau*, Government of India, Ministry of Social Justice & Empowerment, 23-March-2018

[7] Strong, M. and Prinz, P.M., "A Study of the Relationship Between American Sign Language and English Literacy", *The Journal of Deaf Studies and Deaf Education*, 2(1), pp.37-46, 1997

[8] Divya Deora, D. and Bajaj, N., "Indian sign language recognition", In *Emerging Technology Trends in Electronics, Communication and Networking (ET2ECN)*, 2012 1st International Conference on (pp. 1-5). IEEE, December 2012.