SIGN LANGUAGE RECOGNIZER

Dr Nancy Kumari, Harisankar G, Cherin Yacoob Wattacheril,

Neha R, Vinayak K Asish

Abstract

This project introduces a novel approach to sign language detection by employing Long Short-Term Memory (LSTM) neural networks to analyse sequential frame data rather than the conventional frame-by-frame analysis. Utilizing the advanced capabilities of LSTM, which is adept at processing time-series data, our model interprets complex gestures captured across multiple frames, providing a dynamic and fluid understanding of sign language.

The data collection for this project used the Media Pipe framework to capture human body movements. These key points were stored in NumPy arrays for efficient processing. In practical applications, the system captures real-time video streams, processes the last 30 frames through the trained LSTM model, and promptly outputs the gesture recognition results. This setup not only demonstrates the model’s capability in a controlled environment but also provides a foundation for real-time interaction and feedback systems.

The project has potential for expansion with the integration of NLP tools for gesture-to-speech translation and the development of web-based sign language education platforms, enhancing communication aids and paving the way for innovative learning tools in sign language.

Introduction

The emergence of sign language recognition represents a pivotal advancement in technology, offering transformative potential for communication accessibility to everyone. With over 300 sign languages utilized by around 70 million deaf individuals worldwide, the imperative for comprehensive solutions to bridge communication gaps is clear.

However, despite the popularity of communication technologies, the needs of sign language users have often been worsening barriers for the deaf community. Nevertheless, recent progress in sign language processing presents a promising opportunity to address these challenges. Integrating sign language recognition into mainstream technologies such as personal assistants like Siri and Alexa could facilitate seamless communication for deaf users, overcoming traditional voice-based interfaces. Furthermore, the potential of sign language processing extends beyond voice-activated services. By enabling automatic translation of sign language into written queries for search engines or real-time transcription of signed content, new avenues for accessibility and information retrieval can be unlocked. Yet, despite its promise, current research efforts in this field are often fragmented, hampered by disciplinary silos.

To tackle the challenges of sign language processing, we advocate for an interdisciplinary approach. By combining insights from Deaf studies, Convolutional Neural Network (CNN), Long Short term Memory (LSTM) , Machine Learning ,Human Computer Interaction (HCI) we can develop holistic solutions that cater to the diverse needs of the deaf community. This paper aims to comprehensively explore sign language recognition from an interdisciplinary perspective, identifying key challenges and proposing strategies for future research to foster inclusive technologies that empower deaf individuals and promote a more inclusive society.

Related Works

There is an urgent need to close the communication gap in the field of sign language recognition between spoken or written language users and deaf people. Some of them feel distant from this world because of this barrier. Scientists and Engineers are investigating new strategies that take advantage of developments in natural language processing, machine learning, and computer vision.

One notable aspect of recent research involves the use of deep learning techniques, particularly recurrent neural networks (RNNs) like Long Short-Term Memory (LSTM). These models excel at sequential data processing, making them well-suited for tasks involving temporal sequences such as sign language gestures captured in video frames.

By training these deep learning models on large datasets of sign language gestures, researchers aim to create robust recognition systems capable of accurately interpreting a wide range of signs. These systems can identify specific hand shapes, movements, and facial expressions associated with different signs, enabling them to transcribe sign language into written text or spoken language in real time.

Moreover, efforts are underway to enhance the accessibility and usability of sign language recognition systems. This includes the development of real-time recognition systems that can operate efficiently on low-resource devices such as smartphones or tablets. By making these technologies portable and widely available, they can empower deaf individuals to communicate more effectively in various settings, from everyday conversations to professional interactions. Additionally, researchers are exploring ways to improve the inclusivity of sign language recognition systems by considering the linguistic and cultural aspects of sign languages. This involves collaborating with experts in Deaf culture and linguistics to ensure that the systems accurately capture the nuances and subtleties of sign language communication.

Overall, the ongoing research in sign language recognition holds tremendous promise for fostering greater communication accessibility and inclusivity for the deaf community. By leveraging interdisciplinary expertise and innovative technologies, researchers are advancing towards the development of robust, user-friendly systems that facilitate seamless interaction between individuals using sign language and those who communicate through spoken or written language.

Creating The Dataset

In our pursuit of developing an effective sign language detector, the first step involved gathering a dataset for training purposes. Accomplish this, we turned to Python, a versatile programming language, in conjunction with OpenCV, a powerful computer vision library. This strategic combination empowered us to capture images of various sign language poses using a webcam, thereby laying the foundation for our subsequent labelling and model training endeavours. By harnessing the capabilities of OpenCV, we were able to access the webcam and capture images in real-time, ensuring a dynamic and representative dataset. Python's flexibility allowed us to implement custom scripts to streamline the image collection process, ensuring efficiency and accuracy.

Moreover, in the process of collecting images, we recognized the significance of Convolutional Neural Networks (CNNs) and Long Short-Term Memory (LSTM) networks in accessing and processing the acquired images. CNNs, with their ability to automatically learn hierarchical representations of features from raw pixel values, were instrumental in extracting meaningful features from the collected images. This feature extraction process was essential for identifying distinctive patterns and nuances in different sign language poses.

Simultaneously, LSTM networks played a crucial role in capturing the temporal dynamics of sign language gestures. By modelling the sequential nature of sign language sequences, LSTM networks enabled us to understand the fluidity and dynamics inherent in sign language communication. This temporal modelling capability was invaluable for interpreting the sequential progression of gestures over time and inferring the underlying meaning of sign language sequences.

In our project, we use a frame rate of 30 frames per second, capturing each gesture from multiple perspectives. This approach allows our Long Short-Term Memory (LSTM) model to learn from a dataset that represents the same action with subtle variations. By training with this level of detail, our model can accurately detect and interpret sign language gestures, even when there are slight differences in how the action is performed. This included capturing images in various lighting settings, from different angles, and against diverse backgrounds, ensuring our model could accurately recognize signs despite environmental variations. This comprehensive approach to data collection and model training was designed to ensure that our sign language recognition system is not only dependable in controlled environments but also effective in more unpredictable settings.

We worked with a dataset containing 20 different words in sign language, including 'c', 'cat', 'eat', 'father', 'fine', 'forget', 'hello', 'help', 'I love you', 'l', 'learn', 'more', 'mother', 'no', 'not', 'please', 'sad', 'sign', 'thanks', and ' where '. Our model has been trained to recognize these words in various contexts, allowing for flexibility in interpreting sign language. In developing our dataset, we aimed to create a robust collection of images and gestures that would be adaptable to real-world conditions. By focusing on these 20 words, we not only covered common expressions used in everyday communication but also incorporated essential terms that might be crucial in emergency situations.

The collected images were meticulously organized and stored in labelled directories, each corresponding to a specific sign language pose. This systematic approach not only facilitated the subsequent labelling process but also ensured the integrity and organization of our dataset. Each image served as a valuable data point, contributing to the richness and diversity of our training dataset.

Building Sign Language Detector Using TensorFlow

With the dataset prepared and necessary files generated, we embarked on the critical stage of training the TensorFlow model. This phase represents a pivotal step in our journey towards developing an accurate and robust sign language detector. Leveraging the dataset collected through meticulous image acquisition and labelling, we initiated the training process with the objective of optimizing model performance.

Throughout the training process, we meticulously monitored the model’s progress, thoroughly inspecting key performance metrics and observing its ability to accurately detect and classify sign language poses. This iterative approach allowed us to identify potential areas for improvement and fine-tune the model’s parameters accordingly. We continuously adjusted hyperparameters, such as learning rates, batch sizes, and optimization algorithms, to optimize the model’s performance on the training data.

 Model Recognizing the Data Points

A diagram of a computer

Description automatically generated with medium confidence

LSTM Architecture

Additionally, we employed techniques such as regularization and dropout to prevent overfitting and enhance the model’s generalization capabilities. Regularization techniques help mitigate the risk of overfitting by imposing constraints on the model’s weights, ensuring that it does not memorize the training data but instead learns meaningful patterns that generalize well to unseen data. Dropout, on the other hand, randomly deactivates a fraction of neurons during training, forcing the model to learn robust representations and reducing the likelihood of overfitting. Furthermore, we adopted strategies for data augmentation to enrich the training dataset and enhance the model’s ability to generalize to unseen variations in the input data. Techniques such as random rotations, translations, and flips were applied to artificially increase the diversity of the training samples, thereby improving the model’s robustness to variations in pose, lighting, and rch Pbackground. By iteratively fine-tuning the model on the training data and incorporating feedback from validation metrics, we aimed to achieve optimal performance in sign language detection. Our goal was to develop a model that not only demonstrates high accuracy on the training dataset but also generalizes well to unseen data, ensuring reliable performance in real-world applications. Through diligent experimentation and optimization, we endeavoured to unlock the full potential of the TensorFlow model, paving the way for effective sign language detection and communication accessibility.

Data Set Training and Testing

Training the model presented a challenge, especially with fluctuating accuracy during testing. Despite the model's ability to recognize signs and produce correct outputs, the accuracy varied with each iteration. To address this, we aimed to train the model as thoroughly as possible. We used a train-test split of 5%, which means that 95% of our dataset was used for training, while 5% was set aside for testing. The train-test split is a model validation technique that allows us to evaluate how well the model performs with new or unseen data.

We set the epoch to 500, allowing each word in our dataset to be processed through our algorithm 500 times. This extensive training ensures that the model can effectively learn and adapt to the various patterns and nuances in the sign language dataset. During training, we track two key metrics: categorical accuracy and categorical cross-entropy loss. Categorical accuracy calculates the percentage of predicted values (yPred) that correctly align with the actual values (yTrue) in one-hot label format. This metric helps us understand how often the model's predictions match the expected results. Categorical cross-entropy, also known as SoftMax loss, combines a SoftMax activation with a cross-entropy loss, designed for multiclass classification tasks. This loss function allows our Convolutional Neural Network to generate probabilities for each class, which is crucial for effective word recognition.

A graph of a graph

Description automatically generated with medium confidence

Categorical Accuracy

A graph of a graph

Description automatically generated with medium confidence

Cross entropy loss

During further testing, we consistently observed an accuracy of 1, suggesting that the model performed perfectly on the test dataset. However, while this high accuracy might seem promising, it raised concerns about overfitting. Overfitting occurs when a model becomes so tailored to its training dataset that it struggles to generalize to new, unseen data. This is often evidenced by a high accuracy during training or testing, but lower accuracy during real-world prediction.

In our case, the issue of overfitting became apparent when the model, despite achieving near-perfect accuracy during testing, exhibited a significant drop in accuracy when used for predictions, with only about 96% accuracy. This discrepancy highlighted that the model might have memorized the training data instead of learning general patterns that could be applied to new situations.

One probable cause of overfitting could be the small size of the training dataset. When the dataset is limited, the model might not be exposed to enough variability, leading it to learn specific examples rather than broader trends. This can cause the model to perform well on the training data while struggling with new data.

We are assessing th model’s performance by using the confusion matrix. It offers a thorough analysis of true positive (TP), true negative (TN), false positive (FP), and false negative (FN) predictions, facilitating a more profound comprehension of a model’s recall, accuracy, precision, and overall effectiveness in class distinction. The accuracy of the model is calculated by the help of confusion kmatrix the formula is

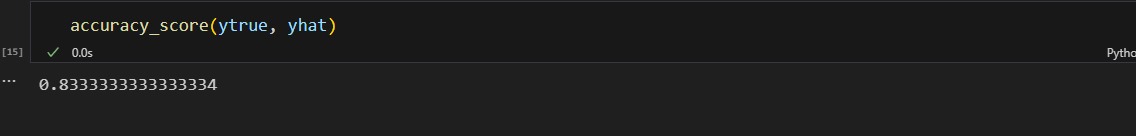
Accuracy = TP+TN

TP+TN+FP+FN

By utilizing these metrics, we evaluate the model's performance and identify areas for improvement during the training process. The combination of categorical accuracy and cross-entropy loss provides a comprehensive view of the model's learning efficiency and predictive accuracy, enabling us to fine-tune the model for optimal results in recognizing a wide range of sign language gestures.

Result Analysis

The model has been successful in predicting most signs correctly, achieving an accuracy rate of 88%. However, we have observed a consistent delay of about 1.2 seconds in the output response time. This delay could be attributed to the operating system environment, with our setup running on Windows. The cause of the delay is not yet fully understood, but it could be related to system-level processing, resource allocation, or other operational overheads.



While the accuracy rate is promising, the output latency might impact the real-time usability of the system.

Model Predicting the Sign

A diagram of a diagram

Description automatically generated

Conclusion

In this research paper, we explored the development of a sign language recognition system using a combination of Convolutional Neural Networks (CNNs), Long Short-Term Memory (LSTM), and various machine learning techniques. Our system was designed to bridge communication gaps between the deaf and hard-of-hearing community and those who use spoken or written language. The creation of a comprehensive dataset was central to our approach, involving meticulous image capture using Python and OpenCV, alongside a robust labelling process. We trained our model on a diverse set of 20 commonly used sign language words, incorporating real-time variations in lighting, angles, and background to ensure robustness and adaptability in real-world scenarios. The dataset's size and diversity contributed to the model's ability to recognize a range of sign language gestures.

Throughout the training process, we employed a train-test split of 5% and set the epoch to 500, allowing the model to learn from the data extensively. We monitored key metrics such as categorical accuracy and cross-entropy loss to assess the model's learning and performance. Despite achieving an accuracy rate of 88%, we encountered challenges with output latency and potential overfitting, indicating areas for further refinement. The observed 1.2-second delay in output response time could impact the system's real-time usability, and further investigation is needed to understand its cause and mitigate it. Overfitting was also a concern, as the model's high accuracy during training and testing did not always translate into consistent results during real-world predictions.

To address these challenges, we propose a combination of strategies, including data augmentation, regularization, and dropout, to improve the model's generalization capabilities. Additionally, exploring more efficient algorithms and optimizing the operating system environment could help reduce output latency and enhance the system's responsiveness. Despite these hurdles, the results of this study are encouraging, demonstrating the potential for AI-driven sign language recognition to foster greater communication accessibility and inclusivity. By continuing to refine the model and address the identified challenges, we aim to develop a robust system that effectively bridges communication gaps for the deaf and hard-of-hearing community.

Future work should focus on expanding the dataset to encompass a broader range of signs, exploring additional methods to reduce overfitting, and enhancing real-time performance. By addressing these areas, we can advance toward a more inclusive society where technology enables seamless communication for everyone.

References

1. Sujan, Vivek A., and Marco A. Meggiolaro. "Sign language recognition using competitive learning in the HAVNET neural network." *Applications of Artificial Neural Networks in Image Processing V*. Vol. 3962. SPIE, 2000.

2. Cooper, Helen, Brian Holt, and Richard Bowden. "Sign language recognition." *Visual Analysis of Humans: Looking at People*. London: Springer London, 2011. 539-562.

3. Lee, Hou-Hsien, Chang-Jung Lee, and Chih-Ping Lo. "Sign language recognition system and method." U.S. Patent Application 12/852,512, filed November 10, 2011.5.Karayılan, Tülay, and Özkan Kılıç. "Sign language recognition." *2017 International conference on computer science and engineering (UBMK)*. IEEE, 2017.

4. Pathan, Refat Khan, et al. "Sign language recognition using the fusion of image and hand landmarks through multi-headed convolutional neural network." *Scientific Reports* 13.1 (2023)

5. S Bragg, Danielle, et al. "Sign language recognition, generation, and translation: An interdisciplinary perspective." *Proceedings of the 21st International ACM SIGACCESS Conference on Computers and Accessibility*. 2019.

6.Bouamor, Houda, Juan Pino, and Kalika Bali. "Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing." *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*. 2023.

7. Zhou, Hao, et al. "Improving sign language translation with monolingual data by sign back-translation." *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2021.

|  |
| --- |
|  |

8. Wan Wang, Qingshan, et al. "Generalizations of wearable device placements and sentences in sign language recognition with Transformer-based model." *IEEE Transactions on Mobile Computing* (2024).

9. Guo, Dan, et al. "Sign Language Recognition." *Multimedia for Accessible Human Computer Interfaces* (2021): 23-59.

10. Srivastava, Sharvani, et al. "Sign language recognition system using TensorFlow object detection API." *International conference on advanced network technologies and intelligent computing*. Cham: Springer International Publishing, 2021.

11. Sunitha Nandhini, A., et al. "Sign language recognition using convolutional neural network." *Journal of Physics Conference Series*. Vol. 1916. No. 1. 2021