Isolated Sign Language Recognition using Sequence Model

OUARDIGHI Omar

Abstract

Although speech is the most common form of communication, there are individuals who face obstacles due to hearing or speaking impairments. This can lead to significant communication barriers for people with these disabilities. Thankfully, the implementation of deep learning methods has the potential to minimize these barriers. This project aims to develop an isolated sign language recognition model that can identify signs made in landmark data extracted from raw videos of sign language gestures. The model will be trained using a combination of LSTM and GRU network and evaluated based on its accuracy in identifying the correct sign gestures. The developed model will be useful for developing mobile apps that can teach parents sign language so they can communicate with their Deaf children.

1 Introduction

Imagine living in a society where everyone can converse with one another without difficulty, regardless of their hearing or speaking abilities. Researchers and computer enthusiasts alike have been intrigued by this vision, which has sparked a rise in interest in sign language recognition (SLR).Sign languages are rich and complex visual languages, characterized by various parameters such as hand shape, orientation, movement, location, and even facial expressions. However, understanding sign language goes beyond these parameters. The same sign can carry different meanings depending on context, execution nuances, and even personal factors like age, gender, and dialect.

The majority of hearing parents who give birth to deaf children do not understand sign language. This might make it difficult for parents to talk to their kids, which can result in Language Deprivation Syndrome. Our goal is to develop a model that can precisely recognize sign motions in edited videos in order to solve this issue.

In this project, we aim to develop an isolated sign language recognition model that can identify signs made in landmark data extracted from raw videos of sign language gestures. This model will be trained using an LSTM and GRU layers and evaluated based on its accuracy in identifying the correct sign gestures. By developing this model , we intend to aid hearing parents in communicating with their Deaf children.

The developed model, we believe, will be able to reliably identify sign gestures in processed videos. If this hypothesis is correct, it will be a big step forward in assisting those with hearing or speech problems to communicate more successfully. This methodology will be especially useful for creating mobile apps that educate parents sign language so they can interact with their Deaf children.

Overall, we anticipate that by giving people with hearing or speech impairments a more effective way to communicate, our research will help to improve their quality of life.

2 Related work

Kothadiya et al. [2] proposed a deep learning-based model for detecting and recognizing sign language gestures, specifically for Indian Sign Language (ISL). The proposed model uses LSTM and GRU models and achieves 97% accuracy on 11 different signs from the IISL2020 dataset. One of the limitations of the work was that the evaluation was performed on a relatively small dataset, with only 11 ASL words. Our project will address this limitation by working on a bigger dataset with 250 words.

B. Fang et al. [6] presented DeepASL, a deep learning-based sign language translation technology that enables non-intrusive American Sign Language (ASL) translation at both word and sentence levels using infrared light as its sensing mechanism. The paper's main finding is that DeepASL achieved an average 94.5% word-level translation accuracy and an average 8.2% word error rate on translating unseen ASL sentences. One of the limitations of the work was that the evaluation was performed on a relatively small dataset, with only 11 participants and a limited number of ASL words and sentences.

Misael et al. [7] describes a study that implemented the SIBI (Sign System for Indonesian Gesture) gesture translation framework on a smartphone to improve its portability and accessibility. The study tested several approaches to improve the translation performance of the SIBI translation system, which used frozen models of three machine learning models. The final mobile SIBI gesture-to-text translation system achieved a word accuracy of 90.560%, a sentence accuracy of 64%, and an average translation time of 20 seconds.

Necati et al. [11] proposed a novel transformer-based architecture called "Sign Language Transformers" that simultaneously learns continuous sign language recognition and translation in an end-to-end manner without relying on gloss level representation. The main finding is that this joint approach leads to significant performance gains in both recognition and translation tasks on the PHOENIX14T dataset, outperforming previous approaches and even doubling the accuracy of some tasks .

De Coster et al. [12] presented a new method for recognizing sign language using a combination of OpenPose for human keypoint estimation and Convolutional Neural Networks for end-to-end feature learning. The proposed method outperforms the previous state-of-the-art approach in recognizing isolated signs in the Flemish Sign Language corpus, achieving an accuracy of 74.7% on a vocabulary of 100 classes. The researchers suggest that future work should focus on extracting salient features from sign language data.

3 Materials and Methods

3.1 Dataset

The dataset used in this project was created by The Georgia Institute of Technology and Deaf Professional Arts Network, you can find it here. it is composed of processed videos of sign language gestures, which were extracted using the MediaPipe holistic model. The landmark data was obtained from the videos and saved in 94477 parquet file, which contain the normalized spatial coordinates of the landmarks for each frame as shown in Figure 1.There are a total of 21 participants in training data and close to 4500 sequences per person. The dataset contains a total of 250 classes of sign language gestures, making it a challenging task to accurately recognize the gestures.

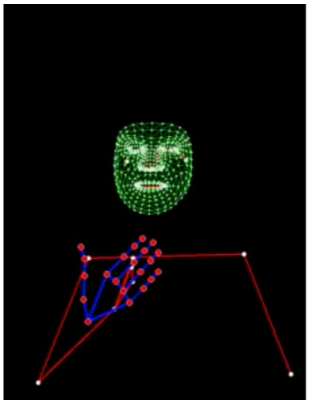

Figure 1: Visualisation of Attribute point on the body

3.2 Methodology

To keep it simple, we used the preprocessed tf.Dataset from tf.Dataset of Google ISL recognition data. The sequences are batched into ragged tensors of batch size 512. Our sign language recognition model takes as input sequences of 3D landmark coordinates that represent various body parts involved in sign language gestures, such as the nose, lips, pose, eyes, and hands. These landmarks are represented by a list of indices that can be refrenced by the Table 1.

Some isolated coordinate values missing from the dataset are represented by a placeholder value of '0.0' and then we concatenated them into one single tensor. Once the dataset was preprocessed, it was split into three sets: the training set, the validation set, and the test set. The dataset was divided in such a way that approximately 80% of the samples were allocated to the training set, while 10% were assigned to both the validation set and the test set

body region	Attribute point range
Nose	[1,2,98,327]
Lip	[0, 61, 185, 40, 39,...]
Left Pose	[513,505,503,501]
Right Pose	[512,504,502,500]
Left Eye	[263, 249, 390, 373, 374, 380,..]
Right Eye	[33, 7, 163, 144, 145, 153, 154, 155, 133,..]
Left Hand	468-489
Right Hand	522-543

Table 1: Landmark indices for different body regions

Our model architecture (Figure 2) is designed to effectively process the temporal dynamics and spatial information inherent in sign language gestures. It is composed of an input layer followed by LSTM and GRU layers for sequence processing. The input layer accepts variable-length sequences of landmark coordinates as its input. These sequences are then passed through an LSTM layer with 512 units, which captures the sequential patterns. To mitigate overfitting, we employ a dropout layer with a dropout rate of 0.5 after the LSTM layer. Next, the processed sequences are further refined using a GRU layer with 512 units. Another dropout layer with a dropout rate of 0.5

is applied after the GRU layer to enhance generalization. Finally, a dense layer with softmax activation generates the predicted probabilities for the different sign language classes.

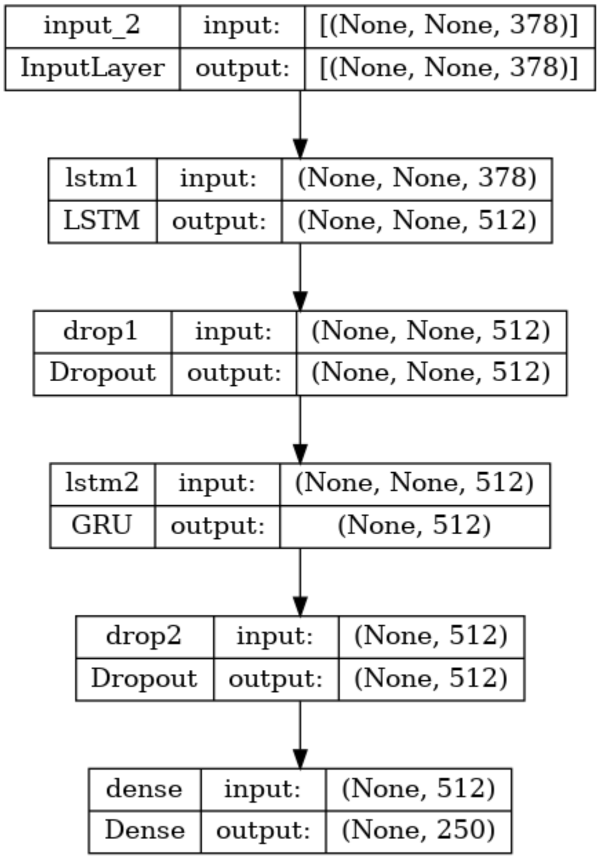

Figure 2: The LSTM-GRU model architecture.

To optimize the model during training, we use the Adam optimizer with a learning rate of 0.001. The training data is divided into batches of size 512 and the model is trained for a total of 100 epochs.

To further enhance the training process and prevent overfitting, we use two additional strategies. The Early Stopping callback monitors the validation accuracy and stops the training if no improvement is observed for a certain number of epochs (10). It also restores the weights of the best-performing model. The ReduceLROnPlateau callback reduces the learning rate by a factor of 0.1 if the validation accuracy does not improve for a certain number of epochs (3). These strategies help optimize the training process and improve the model's performance.

During training, the model evaluates its performance using multiple metrics. The main evaluation metrics used are accuracy and sparse top-k categorical accuracy.The sparse top-k categorical accuracy considers the predictions within the top-5 most probable classes. These metrics provide valuable insights into how well the model is performing on the sign language recognition task.

4 Results

Our model was trained for 56 epochs, before it was early stopped, with a learning rate reduced to 1.0e-09. During the training process, we monitored the loss and accuracy on both the training and validation sets. The loss decreased steadily over the epochs, indicating that the model was learning and converging towards better predictions. The accuracy, on the other hand, showed a gradual increase, indicating that the model was improving in its ability to recognize sign language gestures. In figure3 , We see that our model is overfitting with a marge of 16% hence other reguralization methods besides dropout are needed.

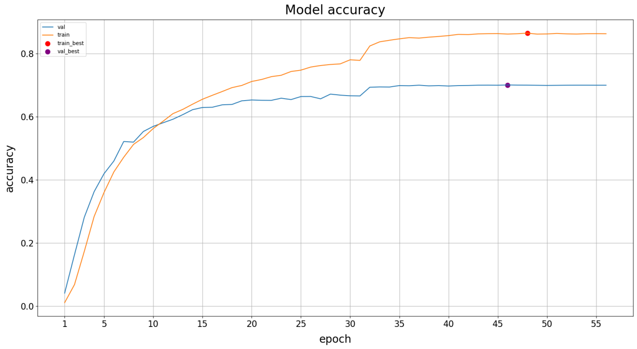

Figure 3: Model accuracy during training and validation

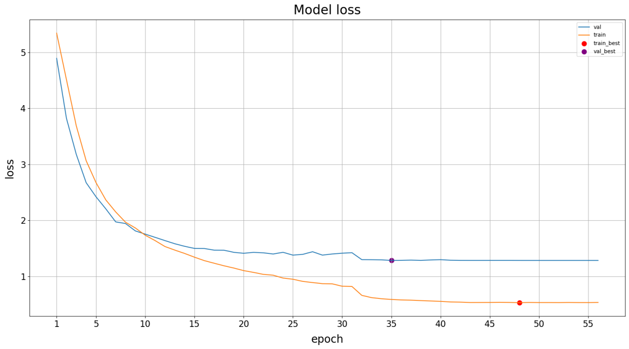

Figure 4: Model loss during training and validation

On the test set ,It achieved an accuracy of **70.5%** . While this accuracy is moderate, there is room for improvement to achieve higher recognition rates.

The sparse top-5 categorical accuracy was measured at **88.16%**. This metric considers the correct gesture prediction being within the top 5 predicted gestures. This indicates that the model has a reasonable understanding of the gestures and can often identify the correct gesture among the top predictions.

The classification report in Table2 provided insights into the performance of the model on specific classes. For example, class "brown" achieved high precision, recall, and F1-score values, indicating accurate recognition. on the other hand, the class "give" recorded the lowest precision, recall, and F1-score values. We also tracked the **AUC** for each class and the results were very high, between 0.96 and 1.0 which suggests that the model has excellent discriminative ability for a particular class.

	precision	recall	f1-score	support	sign ord
brown	1.00	0.94	0.97	51	31
fireman	0.93	0.93	0.93	29	82
find	0.89	0.93	0.91	43	78
bug	0.91	0.91	0.91	46	32
orange	0.91	0.91	0.91	44	162
gum	0.88	0.95	0.91	39	103
:	:	:	:	:	:
lips	0.38	0.39	0.38	36	134
sticky	0.32	0.46	0.38	26	206
beside	0.21	0.25	0.23	24	21
give	0.20	0.24	0.22	34	95
accuracy	0.70	0.70	0.70	9216	

Table 2: Classification report for all signs

5 Discussion and future directions

In analyzing the results obtained for the LSTM and GRU model with [512, 512] layers, we found that the results did not support our initial research hypothesis. This means that our model didn't perform as well as we expected. We can explore several possible explanations for this observed performance:

Firstly, it's possible that the complexity of sign language gestures was not adequately captured by the chosen model architecture. The LSTM and GRU models are commonly used for sequence modeling tasks, but they may not have been able to capture the subtle variations and temporal dependencies present in sign language gestures. To address this, we could consider increasing the complexity of the model by adding more layers or exploring alternative architectures such as the Transformer, which has shown promise in capturing long-range dependencies in sequences.

Another factor to consider is the availability and quality of the training data. In our case, the dataset used for training had an average of 378 rows per sign, indicating a balanced distribution of data. However, this may not have been sufficient to fully capture the diversity and variations within each sign. To improve the model's performance, we could consider augmenting the training data by applying techniques such as data augmentation. This involves creating additional training samples by applying transformations like flipping, rotation, and scaling to the existing data. Additionally, collecting a larger and more diverse dataset that covers a wider range of gestures, including different signing styles and variations across regions, cultures, and individuals, could enhance the model's ability to generalize and improve its performance.

It's also worth mentioning that hyperparameter tuning plays a crucial role in model performance. Parameters such as learning rate, batch size, optimizer, and regularization techniques can significantly impact the model's ability to learn and generalize. Exploring different combinations of hyperparameters through techniques like grid search or random search could help identify the optimal configuration for the models and potentially improve their performance.

By considering the complexity of sign language gestures, augmenting the training data, exploring diverse datasets, optimizing hyperparameters, and evaluating the models using multiple metrics, we can refine the model architecture and improve the performance for sign language gesture recognition.

6 Conclusion

In conclusion, our sign language recognizer model achieved an accuracy of 70.50% and a sparse top-k categorical accuracy of 88.16%. While there is room for improvement, these results show promise for recognizing sign language gestures using deep learning techniques. Future work should focus on increasing the dataset size, exploring advanced model architectures, and optimizing hyperparameters to enhance accuracy and reliability. This research contributes to the development of assistive technologies for the hearing-impaired and promotes better communication between hearing and non-hearing communities.

Code Repository: Isolated-Sign-Language-Recognition

References

[1] M. Al-Qurishi, T. Khalid and R. Souissi, ”Deep Learning for Sign Language Recognition: Current Techniques, Benchmarks, and Open Issues,” in IEEE Access, vol. 9, pp. 126917-126951, 2021, doi: 10.1109/ACCESS.2021.3110912.

[2] Kothadiya D, Bhatt C, Sapariya K, Patel K, Gil-González A-B, Corchado JM. Deepsign: Sign Language Detection and Recognition Using Deep Learning. Electronics. 2022; 11(11):1780. https://doi.org/10.3390/electronics11111780

[3] Oyedotun, Oyebade Khashman, Adnan. (2017). Deep learning in vision-based static hand gesture recognition. Neural Computing and Applications. 28. 10.1007/s00521-016-2294-8.

[4] Huang, Jie Zhou, Wengang Li, Houqiang Li, Weiping. (2018). Attention-Based 3D-CNNs for Large-Vocabulary Sign Language Recognition. IEEE Transactions on Circuits and Systems for Video Technology. PP. 1-1. 10.1109/TCSVT.2018.2870740.

[5] Kavarthapu, Dilip Mitra, Kaushik. (2017). Hand Gesture Sequence Recognition using Inertial Motion Units (IMUs). 10.1109/ACPR.2017.159.

[6] B. Fang, J. Co, and M. Zhang, “Deepasl: Enabling ubiquitous and non- intrusive word and sentence-level sign language translation,” in Proceed-ings of the 15th ACM Conference on Embedded Network Sensor Systems,2017, pp. 1–13.

[7] Jonathan, M., Rakun, E. (2022). Translating SIBI (Sign System for Indonesian Gesture) Gesture-to-Text in Real-Time using a Mobile Device. Journal of ICT Research and Applications, 16(3), 259-280. https://doi.org/10.5614/itbj.ict.res.appl.2022.16.3.5

[8] S. S. Kumar, T. Wangyal, V. Saboo, and R. Srinath, “Time series neural networks for real time sign language translation,” in 2018 17th IEEE In-ternational Conference on Machine Learning and Applications (ICMLA). IEEE, 2018, pp. 243–248.

[9] J. Huang, W. Zhou, H. Li, and W. Li, “Attention-based 3d-cnns for large-vocabulary sign language recognition,” IEEE Transactions on Circuits and Systems for Video Technology, vol. 29, no. 9, pp. 2822–2832, 2018.

[10] R. Cui, H. Liu and C. Zhang, ”Recurrent Convolutional Neural Networks for Continuous Sign Language Recognition by Staged Optimization,” 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 2017, pp. 1610-1618, doi: 10.1109/CVPR.2017.175.

[11] Camgoz, Necati Koller, Oscar Hadfield, Simon Bowden, Richard. (2020). Sign Language Transformers: Joint End-to-End Sign Language Recognition and Translation. 10.1109/CVPR42600.2020.01004.

[12] De Coster et al., (LREC 2020).Sign Language Recognition with Transformer Networks. (https://aclanthology.org/2020.lrec-1.737)