Less is More: Techniques to Reduce Overfitting in your Transformer Model for Sign Language Recognition

Joe Huamani-malca
PUCP
Lima, Peru
huamani.jn@pucp.edu.pe

Gissella Bejarano
Marist College University
New York, USA
gissella.bejarano@marist.edu

Abstract

Sign language recognition (SLR) in deep learning is a challenging task due to the need for interpreting human body movements, including detailed hand movements and facial expressions. Recent research has focused on using keypoint body landmarks and transformer models to improve SLR performance. However, these models can face overfitting issues due to the need for more available datasets. To address these problems, we analyze three Peruvian sign language (LSP) datasets for SLR. Additionally, we apply several techniques to reduce overfitting in the Spoter model, a transformer-based architecture for SLR. The results of these techniques reveal that the data-based techniques improve generalization and reduce overfitting in transformer-based models for SLR.

1. Introduction

Sign language is an important form of communication for the deaf community. Machine learning technology has advanced the ability to communicate through text, voice, and images, but sign language recognition is still a challenge. This is due to the need to capture and interpret facial expressions and detailed hand movements, as well as the vast vocabulary and variation of signs [10, 15]. Recent research has leveraged human action recognition (HAR) technology to develop sign language recognition systems [20].

Recent SLR research has focused on body keypoint landmarks [1, 14, 23] to reduce computational requirements, but this requires a pre-trained body keypoint estimator. On the other side, transformer models and their variations are gaining popularity due to their successful performance in tasks like HAR [16, 18, 22] also used and SLR [4, 7, 27, 31]. Nevertheless, overfitting and limited datasets remain challenges for transformer models in SLR. Hence, it is crucial to address these issues and enhance model generalization. Our work contributes by analyzing three PSL datasets for sign language recognition and unifying them into a comprehensive dataset. Additionally, we enhance Spoter model metrics by addressing overfitting with several techniques.

2. Related work

This section overviews the state-of-the-art sign language recognition (SLR) models. Prominent models include Convolutional Neural Networks (CNNs), Long Short-Term Memory Networks (LSTMs), and Transformers [2, 4, 6, 19, 29]. Other recent works have explored the use of attention mechanisms, multi-modal fusion, and transfer learning for improving the performance of SLR models [8, 11, 12, 24].

Deep learning models often face the challenge of overfitting during training. Several techniques can mitigate this issue, such as regularization, early stopping, learning rate variation, label smoothing, dropout, weight decay, and batch normalization [5, 17, 21, 28]. Data-related techniques, such as outlier removal, handling missing values, data augmentation, noise reduction, and data smoothing [9, 13, 26, 30], can also be applied. Combining these techniques can improve the generalization performance of deep learning models for SLR tasks.

3. Data Analysis

This section explains the data analysis we performed to clean, preprocess and prepare the data for the SLR model.

3.1. Datasets

We utilize three distinct datasets: AEC, PUCP-DGI156, and PUCP-DGI305. Each comprises a unique set of classes, with more than 50% classes overlapping between datasets.

AEC: This dataset is created from two 30-minute videos of the Educational Peruvian TV show called "Aprendo en casa" (Learning at home). Two different interpreters appear in the dataset. The video frames of 29.9 fps were cropped 220x220 pixel that focus on the interpreter's small corner square, and the signs were segmented based on the spoken words in the program [2].

PUCP-DGI156: This dataset comprises 27 videos of 29.9 fps in which 19 deaf Peruvian signers tell stories. These videos were recorded in various settings, including classrooms. The recordings in this dataset have not been standardized, so some have a noisy background, different signers' camera-distance, and some show zoom-in and zoom-out effects. This dataset is annotated and reviewed by the same annotator.

PUCP-DGI-305: this dataset has 1920x1080 resolution videos of deaf Peruvian signers making sentences in 29.9 fps. The videos in this dataset are standardized, meaning they have a white background and were recorded using the same camera distance. An LSP native annotated this dataset; then it has reviewed by a deaf and followed by a linguistic master student who knows LSP. Finally, a linguistic expert standardized the labels used in the dataset. This dataset is still growing and not publicly available to date.

The combinations of these three LSP datasets for more than 22 instances per class allow us to use 50 classes and a stratified split of 80% for training and 20% for validation has been applied. The distributions of instances per class are shown in Figure 1. The training distribution of the multi-dataset comprises approximately 28.42% from the AEC dataset, 51.05% from the PUCP-DGI156 dataset, and 20.58% from the PUCP-DGI305 dataset.

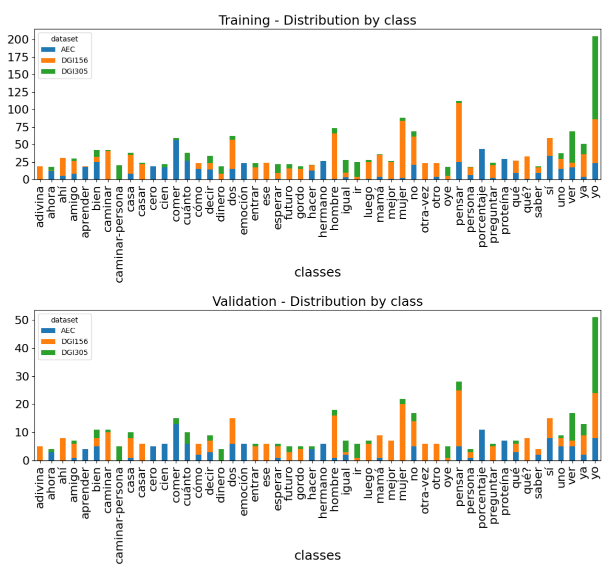

Figure 1. distribution of the number of instance per classes of the train (top) and the validation (bottom) for the mix of the three datasets, all classes have more than 20 instances

We preprocessed these three datasets using the PeruSil framework to create a continuous dataset[1] [2] to obtain isolated sign videos. Next, we used the ConnectingPoints repository [2] to extract the keypoint landmarks of the signer from each video. This repository uses a pre-trained pose estimation model from Google called "Mediapipe Holistic.". Our experiment involved a total of 54 keypoints, comprising 42 from the hands, 7 from the upper body pose, and 5 from the face.

4. Spoter model

The SPOTER model is a transformer-based architecture for SLR from a sequence of keypoint landmarks data. Similarly, as presented in [3]. It has 6 encoder and decoder layers, 9 heads, 2048 feed-forward dimensions, and 108 hidden dimensions. The input is transformed into one dimension feature vector before feed the model. The architecture includes a customized transformer decoder layer that omits the repeated self-attentional operation found in the standard implementation. The model employs a linear layer at the end to make class predictions. The model uses the Adam optimizer and the cross-entropy loss function. The main goal of the Spoter model authors is to create a pre-trained model that is lightweight and can learn quickly.

5. Reduce overfitting - Data

In this section, we will describe the techniques applied to the data to reduce overfitting and their respective results.

5.1. Cross Validation

To evaluate the performance of our model and identify any potential issues with the dataset, we employed a stratified k-fold cross-validation method: the dataset was divided into five parts, with similar proportions of instances per class in each split. For each experiment, we took a different fold as validation split.

The model performed well across all folds, as shown in Figure 2, with a validation accuracy of 51% (Table 1). However, the validation loss started to increase while train accuracy approaches 100%, suggesting overfitting. Our analysis showed that this overfitting was not due to any particular fold.

5.2. Data cleaning

This is a critical step in the machine learning pipeline that is often overlooked but can have a significant impact on the performance of the model. While it is not a technique for reducing overfitting, it is an essential step that must be taken to ensure that the model is training correctly. This can include removing duplicates, correcting misspellings, and removing outliers.

In our case, we checked miss-annotated videos with an LSP expert. Some videos were out of phase and did not

[1] https : / / github . com / gissemari / PeruvianSignLanguage

[2] https://github.com/JoeNatan30/ConnectingPoints

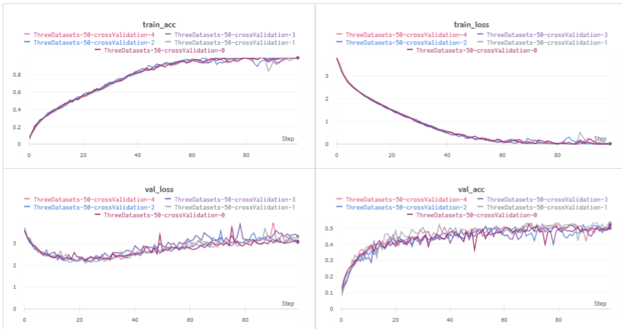

Figure 2. Training and validation metrics for five experiments, each corresponding to a cross-validation fold. Top-left: training accuracy. Top-right: training loss. Bottom-left: validation loss. Bottom-right: validation accuracy

represent the sign. We also removed videos with less than 3 frames. This resulted in a different distribution of instances in the dataset. With the AEC dataset seeing a reduction of 17.42% and the PUCP-DGI156 dataset seeing a reduction of 3.73%. Despite these reductions, the accuracy improved from 51% to 55% (Table 1). However, the validation loss remained similar to that of the bottom-left plot shown in Figure 2, indicating that further improvements may be necessary.

5.3. Reduce the dataset

Analyzing datasets from different sources can be challenging due to imbalanced classes and quality variations. To address these issues, we analyzed each dataset separately to determine which ones contribute to a robust model.

After testing each dataset using the spotter model, we found that the PUCP-DGI156 dataset had an accuracy of 25%, which is lower than the result obtained from AEC and PUCP-DGI305, which is 60% and 61% respectively. Additionally, the PUCP-DGI156 dataset showed unusual loss behavior, with losses increasing faster than the other datasets, for example AEC dataset, as seen in Figure 3.

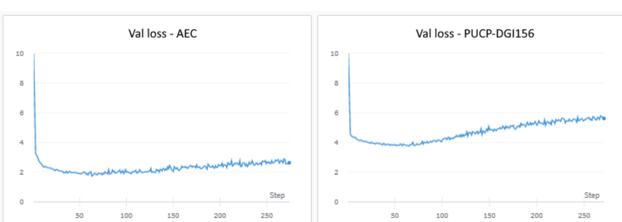

Figure 3. Comparison of validation loss between the AEC and PUCP-DGI156 datasets. The validation loss of AEC is on the left, and the validation loss of PUCP-DGI156 is on the right

We removed the PUCP-DGI156 dataset and combined the remaining datasets. This resulted in a new dataset of 50 classes, as shown in Figure 4, with 61.64% of AEC and the rest of PUCP-DGI305. Testing the new dataset generated an improvement in accuracy from 55% to 68% (Table 1).

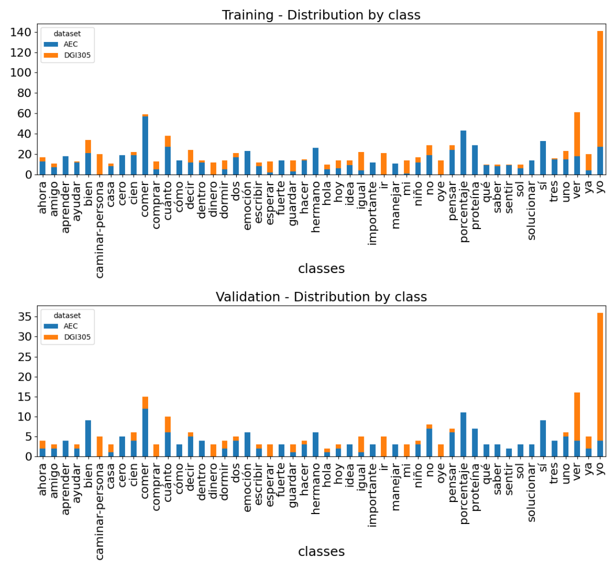

Figure 4. distribution of the number of instance per classes of the train (top) and the validation (bottom) of the mix of AEC and PUCP-DGI305, all classes have more than 9 instances

Technique	Acc
Cross-validation	50.93 %
Data-cleaning	55.07 %
Reduce dataset	68.21 %
Data Augmentation	68.93 %

Table 1. Cumulative Accuracy of Data-Oriented Techniques for Overfitting Reduction.

5.4. Data augmentation (AUG)

We used the data augmentation techniques proposed by [3] to increase the diversity of the training samples and improve the robustness of the trained model. We randomly applied each technique to each training instance with a probability of 50%.

Rotation: Rotates the image around its center by a random angle within a specified range. This technique helps the model learn to recognize signs at different orientations.

Shear squeeze: Deforms the image along the x and y axes by helping the model learn to recognize signs whose signers are tall or wide.

Shear perspective: Simulates a 3D perspective distortion in 2D images, and help the model learn to recognize signs from different viewpoints.

Joint rotation: Variates the angle between two consecutive joints. This helps the model learn to recognize different arms movements.

The data augmentation experiments show a small increment in accuracy while maintaining similar loss metrics. This could be due to the limited range of variation values used.

6. Reduce overfitting - Training process

After applying overfitting reduction techniques to the data, this section will explore the effect of applying overfitting reduction techniques to the training process.

6.1. Class weighting (CW)

Class weighting is a technique used to assign different importance to each class during training. In our experiment, we used the inverse of the proportion of each class to determine the weight of each class.

$$ClassWeight(r) = \frac{1}{N_{\text{class (r)}}}$$

6.2. Model complexity reduction (MCR)

One of the techniques used to deal with overfitting is to reduce the number of trainable parameters of the model. In our experiments, we reduced the number of trainable parameters in the model by reducing the feedforward dimension from 4096 to 256. This resulted in a 58.25% reduction in trainable parameters.

6.3. Label smoothing (LS)

Label smoothing is a regularization technique that reduces overfitting by making the model less confident in its predictions. It does this by adding a small amount of noise to the one-hot encoding for the labels [25]. This diffuse way of learning makes the model less likely to memorize the training data too closely, which can improve its ability to generalize to new data

7. Results

The experiments in this section were done after reducing the dataset. We experimented with no techniques (baseline), and then with one technique at a time. We also experimented with grouping some techniques to see the impact on loss and accuracy. Figure 5 shows that the train and validation loss behavior is similar for most techniques. However, for the two experiments where label smoothing was applied, the training loss plateaus instead of increasing, while the validation loss continues to decrease.

The Table 2 shows that combining techniques can improve accuracy by 1%. Using class weight and reducing model complexity both lead to lower train loss, and this effect keeps their behaviour when both techniques are used together. Although label smoothing shows a lower value compared to the others, we consider that this value may change with more epochs.

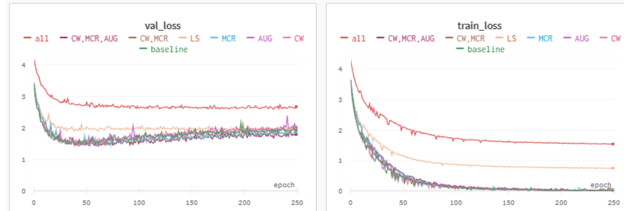

Figure 5. Comparison between the training and validation loss of the baseline and of each of the techniques

Technique	Train loss	Val loss	Acc (top1)
Baseline	0.025	1.806	68.21 %
CW	0.006	1.802	68.93 %
AUG	0.025	1.868	68.93 %
MCR	0.007	1.768	68.21 %
LS	0.749	1.940	67.50 %
MCR,CW	0.014	1.807	67.86 %
MCR,CW,AUG	0.072	1.617	69.29 %
all	1.606	2.615	69.29 %

Table 2. Top-1 accuracy of Baseline (after reduce the dataset) and the use and combination of each reduce overfitting technique

8. Conclusion

In conclusion, SLR is an important research topic, especially for improving communication with the deaf community. Recent advancements in SLR technology have shown promising results, but the limited availability of training datasets remains a challenge.

This paper details three dataset for isolated sign recognition, do an analysis of combining these dataset, and proposes data-oriented and model-oriented techniques for reducing overfitting in the Spoter model. Experiments show that data-oriented techniques are more effective than model-oriented techniques.

Although model-oriented techniques have less impact, they can be beneficial as the size of the dataset grows. We also have to mention that these experiments did not consider larger epoch numbers, so it is possible that some techniques, such as label smoothing, could have better results due to the way they modify the loss behavior.

We hope that this research will be useful to researchers looking to optimize their SLR transformer models. By improving the accuracy and efficiency of SLR models, we can contribute to the accessibility of the deaf community and make it easier for deaf people to communicate with others.

References

[1] Sholikhatul Amaliya, Anik Handayani, Muhammad Akbar, Heru Wahyu, Osamu Fukuda, and Wendy Kurniawan. Study on hand keypoint framework for sign language recognition. pages 446–451, 10 2021. 1

[2] Gissella Bejarano, Joe Huamani-Malca, Francisco Cerna-Herrera, Fernando Alva-Manchego, and Pablo Rivas. PeruSIL: A framework to build a continuous Peruvian Sign Language interpretation dataset. In *Proceedings of the LREC2022 10th Workshop on the Representation and Processing of Sign Languages: Multilingual Sign Language Resources*, pages 1–8, Marseille, France, June 2022. European Language Resources Association. 1, 2

[3] Matyáš Boháček and Marek Hrúz. Sign pose-based transformer for word-level sign language recognition. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV) Workshops*, pages 182–191, January 2022. 2, 3

[4] Necati Camgoz, Oscar Koller, Simon Hadfield, and Richard Bowden. Sign language transformers: Joint end-to-end sign language recognition and translation. 03 2020. 1

[5] Tianlong Chen, Zhenyu Zhang, Sijia Liu, Shiyu Chang, and Zhangyang Wang. Robust overfitting may be mitigated by properly learned smoothening. In *International Conference on Learning Representations*, 2021. 1

[6] Runpeng Cui, Hu Liu, and Changshui Zhang. A deep neural framework for continuous sign language recognition by iterative training. *IEEE Transactions on Multimedia*, PP:1–1, 07 2019. 1

[7] Mathieu De Coster, Mieke Van Herreweghe, and Joni Dambre. Sign language recognition with transformer networks. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 6018–6024, Marseille, France, May 2020. European Language Resources Association. 1

[8] Mathieu De Coster, Mieke Van Herreweghe, and Joni Dambre. Isolated sign recognition from rgb video using pose flow and self-attention. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pages 3441–3450, June 2021. 1

[9] Jianglin Huang, Yan-Fu Li, and Min Xie. An empirical analysis of data preprocessing for machine learning-based software cost estimation. *Information and Software Technology*, 67:108–127, 2015. 1

[10] Nada Ibrahim, Hala Zayed, and Mazen Selim. Advances, challenges, and opportunities in continuous sign language recognition. *Journal of Engineering and Applied Sciences*, 15:1205–1227, 12 2019. 1

[11] Songyao Jiang, Bin Sun, Lichen Wang, Yue Bai, Kunpeng Li, and Yun Fu. Sign language recognition via skeleton-aware multi-model ensemble. *arXiv preprint arXiv:2110.06161*, 2021. 1

[12] Songyao Jiang, Bin Sun, Lichen Wang, Yue Bai, Kunpeng Li, and Yun Fu. Skeleton aware multi-modal sign language recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, 2021. 1

[13] Younsik Kim, Dongjin Oh, Soonsang Huh, Dongjoon Song, Sunbeom Jeong, Junyoung Kwon, Minsoo Kim, Donghan Kim, Hanyoung Ryu, Jongkeun Jung, Wonshik Kyung, Byungmin Sohn, Suyoung Lee, Jounghoon Hyun, Yeonghoon Lee, Yeongkwan Kim, and Changyoung Kim. Deep learning-based statistical noise reduction for multidimensional spectral data. *Review of Scientific Instruments*, 92:073901, 07 2021. 1

[14] Sang-Ki Ko, Jae Gi Son, and Hyedong Jung. Sign language recognition with recurrent neural network using human keypoint detection. *Proceedings of the 2018 Conference on Research in Adaptive and Convergent Systems*, 2018. 1

[15] Oscar Koller, Jens Forster, and Hermann Ney. Continuous sign language recognition: Towards large vocabulary statistical recognition systems handling multiple signers. *Comput. Vis. Image Underst.*, 141:108–125, 2015. 1

[16] Bing Li, Wei Cui, Wen Wang, Le Zhang, Zhenghua Chen, and Min Wu. Two-stream convolution augmented transformer for human activity recognition. In *AAAI Conference on Artificial Intelligence*, 2021. 1

[17] Amitha Mathew, Amudha Arul, and S. Sivakumari. *Deep Learning Techniques: An Overview*, pages 599–608. 01 2021. 1

[18] Vittorio Mazzia, Simone Angarano, Francesco Salvetti, Federico Angelini, and Marcello Chiaberge. Action transformer: A self-attention model for short-time pose-based human action recognition. *Pattern Recognition*, page 108487, 2021. 1

[19] Ilias Papastratis, Kosmas Dimitropoulos, Dimitrios Konstantinidis, and Petros Daras. Continuous sign language recognition through cross-modal alignment of video and text embeddings in a joint-latent space. *IEEE Access*, 8:91170–91180, 2020. 1

[20] Razieh Rastgoo, Kourosh Kiani, and Sergio Escalera. Sign language recognition: A deep survey. *Expert Syst. Appl.*, 164:113794, 2021. 1

[21] Leslie Rice, Eric Wong, and J. Zico Kolter. Overfitting in adversarially robust deep learning. In *International Conference on Machine Learning*, 2020. 1

[22] Yoli Shavit and Itzik Klein. Boosting inertial-based human activity recognition with transformers. *IEEE Access*, PP:1–1, 04 2021. 1

[23] Kun Su, Xiulong Liu, and Eli Shlizerman. Predict & cluster: Unsupervised skeleton based action recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020. 1

[24] Suharjito Suharjito, Narada Thiracitta, and Herman Gunawan. Sibi sign language recognition using convolutional neural network combined with transfer learning and non-trainable parameters. *Procedia Computer Science*, 179:72–80, 01 2021. 1

[25] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jonathon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of*

IEEE Conference on Computer Vision and Pattern Recognition,, 2016. 4

[26] Luke Taylor and Geoff Nitschke. Improving deep learning with generic data augmentation. In *2018 IEEE Symposium Series on Computational Intelligence (SSCI)*, pages 1542–1547, 2018. 1

[27] Anirudh Tunga, Sai Nuthalapati, and Juan Wachs. Pose-based sign language recognition using gcn and bert. pages 31–40, 01 2021. 1

[28] Bingzhe Wu, Zhichao Liu, Zhihang Yuan, Guangyu Sun, and Charles Wu. Reducing overfitting in deep convolutional neural networks using redundancy regularizer. In *International Conference on Artificial Neural Networks*, 2017. 1

[29] Qinkun Xiao, Minying Qin, and Yuting Yin. Skeleton-based chinese sign language recognition and generation for bidirectional communication between deaf and hearing people. *Neural Networks*, 125, 02 2020. 1

[30] Jordan Yeomans, Simon Thwaites, William S. P. Robertson, David Booth, Brian Ng, and Dominic Thewlis. Simulating time-series data for improved deep neural network performance. *IEEE Access*, 7:131248–131255, 2019. 1

[31] Kayo Yin and Jesse Read. Better sign language translation with STMC-transformer. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 5975–5989, Barcelona, Spain (Online), Dec. 2020. International Committee on Computational Linguistics. 1