Enhancing Word-Level Translation of American Sign Language Using Modified 3D Convolutional Networks

Jadon Geathers
Stanford University
geathers@stanford.edu

Nikhil Suresh
Stanford University
ncsuresh@stanford.edu

Abstract

Real-time translation of American Sign Language (ASL) has a been a technical problem of interest, as it provides the opportunity to assist millions of deaf people in communicating with non-speakers more effectively. Several studies propose visual recognition techniques using deep learning that have already shown promise as potential solutions to the problem. This study, in particular, uses the World-Level American Sign Language dataset to approach the translation task from ASL to English. We take inspiration from the 3D Convolution Model developed by Li et al. [9], identifying and resolving a mathematical error in the evaluation metric presented by the authors. We then propose novel adaptations to the model through Self-Attention and Squeeze-and-Excitation mechanisms, which better capture contextual, long-range dependencies between image frames and produce more accurate translations.

1. Introduction

The art of language translation has long been a topic of interest for computer scientists, linguists, and sociologists alike, as it is a problem that lies at the heart of social interconnectedness. Providing fully accurate translations remains an unsolved problem, especially with non-vocalized languages such as American Sign Language (ASL).

There are several existing methods of translation between ASL and English. Many methods perform the simpler task of character-level translation, but word-level translation–which involves a multitude of physical elements like facial expression–proves more difficult. However, with word-level translation, several interesting challenges arise. For one, the meaning of signs depends on the complex combination of gestures, movements, and expressions, and subtle differences in these aspects can easily translate improperly. Additionally, context is an important factor as signs may have multiple counterparts that depend on usage. Consequently, these errors are propagated further in small-scale datasets.

In this project, we approach the word-level translation task between ASL and English from a computer vision perspective. As inputs for the models, we use video data containing signs from the Word-Level American Sign Language (WLASL) dataset. These videos are monocular RGB recordings collected from the internet, showcasing different signers performing various ASL words in near-frontal views. As a baseline method, we use the 4 models described in the WLASL paper [9] as they have been tested on the same dataset we use.

We will quantitatively evaluate the performance of our model against existing approaches proposed by the WLASL creators. We compare the precision of each baseline model and our model at various subsets of the dataset, the top-1, top-5, and top-10 precision (where top-k references the top k nearest words). Qualitatively, we visualize heatmaps to gain insights into semantic similarities, assess the strengths of our model, understand the performance trajectory, address failures cases, and conduct a holistic analysis of our model within the context of ASL.

2. Related Works

To provide more context for this study, we take from pre-existing insights on word-level and sentence-level translation.

There have already been several approaches to translating American Sign Language and other sign languages in general. One popular approach used for translation is the use of recurrent neural networks (RNNs), which we believe underscore the importance of context and continuity in signed language. For example, [8] creates an application prototype for learning ASL by using an RNN for an embedded sign language recognition mechanism. Similarly, [3] performs sign language translation for Chinese sign language, using an RNN-based model to map extracted video features to sentence-level labels.

Another popular approach is that of convolutional neural networks (CNNs) for both sign language translation and

recognition. [7] attempted to embed a CNN within a hidden Markov model with success, although the computational inefficiency of training the model poses a challenge. On the other hand, [11] uses an entirely convolutional structure to perform Indian sign language translation in selfie mode, achieving high recognition accuracy. Despite training costs, authors from both papers highlight the promising performance of CNNs within the problem setting.

In our paper, we take inspiration from the convolutional approaches to sign language translation. First, the paper by Li et al. on word-level deep sign language recognition provides the Word-Level American Sign Language (WLASL) dataset that we will be using in this project to evaluate the performance of our deep learning models [9]. Moreover, the authors compare several deep learning models for word-level sign recognition, including a baseline using VGG and GRU architectures, as well as 3D convolution networks and a novel pose-based temporal graph convolutional network (Pose-TGCN) which captures spatial-temporal dependencies in human pose trajectories. We use their 3D-CNN architecture, which they found to produce the highest accuracy in translation among all their models tested, as is described further in the technical approach section.

The paper by Huang et al. also employs the use of 3D CNNs in capturing the spatial-temporal features directly from raw videos [4]. However, for this architecture, elements like color information, depth clue, and body joint positions are inputted into the 3D CNN to integrate color, depth and trajectory information. This model provides insights into the optimization of certain model parameters and features, but because it was adapted and tested on small-scale datasets, it may lack the generalizabiilty we wish for our model to have when used in practical settings.

Finally, we also inspect a significantly different approach presented by Fang et al. in their paper on Non-Intrusive Word and Sentence-Level Sign Language Translation [2]. While the other models focus on word-level translation, their model DeepASL targets both by using infrared light as the sensing mechanism. In doing so, they opt for a novel hierarchical bidirectional deep recurrent neural network (HB-RNN) for word-level translation and a probabilistic framework based on Connectionist Temporal Classification (CTC) for sentence-level ASL translation. In this project, we focus on word-level translation, but it would be insightful to compare RNN models to CNNs in not only their fundamental approaches but the extent and accuracy to which they can translate ASL.

We expect several challenges with respect to creating an effective model that can recognize sign language. Importantly, [5] identifies that one critical challenge is with recognizing features not just of the hands but of facial expressions, involving minors details with eye brows or slight eye widening, and other subtle body motions. This adds a layer of complexity when modeling as it increases the diversity of ways in which an individual can sign a single gesture. Another challenge identified by [6] is in being able to distinguish a sign from several different perspectives, whether it is a close up of only the hands or an angled perspective of the person, for instance. We suspect that the breadth of the WLASL dataset we use will be highly useful in adapting to these challenges.

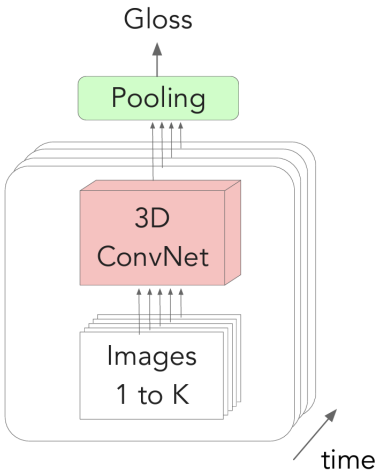

Figure 1. The architecture of the baseline 3D convolutional network used for video classification. Image taken from [9].

3. Technical Approach

We create a pipeline that allows for real-time word-level translation of ASL. Initially, we convert ASL image sequences to text. Our ASL-to-English word-level translator will be modeled as a 3D convolutional network, which allows us to handle the continuity present in the time dimension for the video segments. These segments will be discretized into images on a per-frame basis to serve as input to the model, as illustrated in Figure 1.

3.1. 3D Convolutional Model

To develop and train the translator, we use the World-Level American Sign Language (WLASL) video dataset [9] and implement the 3D convolutional model as developed by the creators of WLASL. We primarily investigate this model, as it demonstrates better performance on the classification task.

The model processes video frame sequences through a series of 3D convolutional layers, max-pooling layers, and inception modules to extract spatio-temporal features. Initially, 3D convolutions with large kernel sizes capture comprehensive features from the input frames, which are then followed by max-pooling to reduce spatial dimensions. Furthermore, the model enhances feature extraction with con-

volutions and inception modules, which are designed to capture multi-scale features by applying multiple convolution operations with different kernel sizes in parallel. Finally, it concludes with average max-pooling and a final convolution to produce the logits for classification. This model structure can be visualized in a simplified form via Figure 1. Performance is ultimately evaluated using binary cross entropy loss and the top-k accuracy metric, which is explained in Section 5.2.

3.2. Self-Attention and Squeeze-and-Excitation

To augment the performance of the convolutional network, we explore the addition of two attention mechanisms, namely

- **Self-Attention.** By implementing self-attention, we enable the model to focus on different parts of the input video and dynamically weigh the importance of each part. For our ASL-to-English translator, this is useful for allowing the model to understand long-range dependencies between different video frames, which aids in capturing the complexity of poses, movements, and hand gestures.

  The self-attention mechanism computes the attention score using the following formula:

  $$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$

  where Q, K, and V represent query, key, and value matrices respectively, similar to standard attention mechanisms, and d_k is the dimension of the key vectors.

  This mechanism allows the model to focus on different parts of the input sequence and thus helps the model better understand the context in which each sign occurs. As mentioned before, it is inserted after the inception blocks to capture long-range dependencies in the feature maps.

- **Squeeze-and-Excitation (SE).** This technique prompts the model to focus primarily on the important features in the video while de-emphasizing the relevance of others. This is achieved by inspecting the inter-channel dependencies, then performing a rescaling of the weight matrix along the channel dimension. Rescaling the weights improves the model's attention towards key motions, consequently yielding more accurate translations. Squeeze-and-excitation has the additional advantage of being less computationally expensive than self-attention.

  We calculate the SE mechanism as follows:

  $$\text{SE}(x) = \sigma(W_2\text{ReLU}(W_1 F_{\text{avg}}(x))) \cdot x$$

  where F_{avg} is the global average pooling layer, W_1 and W_2 are fully connected layers, σ is the sigmoid activation function, and $\cdot$ denotes element-wise multiplication. This process scales the input features based on their importance. Like self-attention, the SE block is integrated within each inception module to emphasize important features at various depths in the network.

In addition to testing the performance of the techniques, we alter the existing hyperparameters such as the batch size, dropout rate, learning rate, and learning rate decay, as illustrated in Section 5.2. Given that there is a diverse array of individuals signing within the dataset, dropout is particularly important for ensuring that the model is generalizable.

4. Dataset and Features

For this project, we use the Word-Level American Sign Language dataset (WLASL) [9] which consists of over 20,000 videos with each containing one sign in ASL. The 119 signers signed 2,000 different words in ASL with each sign being performed by at least 3 individuals in order to include for inter-signer variations and allow for generalizability of the trained sign recognition models. The WLASL dataset only contains videos from the near-frontal position to achieve the highest quality, as people typically communicate in frontal perspective.

In creating the dataset, the authors used YOLOv3 detection tool to identify and isolate the body boundaries of each of the signers in the video which helps to standardize videos across different filming setups. Additionally, it contains annotations for dialects that are commonly found in ASL. We chose this dataset amongst others as it is the standard for use as a large scale word-level dataset and contains three times as much data as the next largest dataset, the American Sign Language Lexicon Video Dataset [1].

Other common word-level ASL datasets, such as the Purdue RVL-SLLL ASL Database and Boston ASLLVD, were also considered for this project, but they proved unsuitable for large class training as they were sizably smaller than the WLASL with many words only having a few examples [10] [1]. Figure 2 demonstrates an example from the dataset with the time-series frames from the video data, where two different signers sign the word "scream."

In the study that created the WLASL dataset, the authors conducted testing on datasets of different sizes, selecting top-k words for dataset subset sizes of 100, 300, 1000, and 2000. In this project, because of limitations in computation with only one GPU, we are unable to elect to evaluate on the WLASL2000 subset and instead opt for the WLASL100 subset. However, with on average 10.5 samples for each word, we are able to properly train the model for reproducible results.

Figure 2. Two signers signing the word "scream" in different exaggerations and manners. Image taken from [9].

5. Results

We now present our results, assessing the validity of published results, demonstrating the top-k precision metrics of our various models, and conducting a qualitative analysis of our findings.

5.1. Discrepancy in Per-Class Accuracy Calculation in Published Work

As one principal result, we first take note of a mathematical error in the paper presented by [9]. The authors claim that they report the accuracy of the 3D convolutional model in their paper. However, the authors calculate the *precision* of the model in their codebase instead of *accuracy*, leading us to suspect that the true accuracy differs from the results published in the paper.

To see this error, we define by TP, FP, TN, and FN the number of true positive, false positive, true negative, and false negative predictions. Here, positive indicates that the example is predicted to belong to the top-k classes, whereas negative indicates that the example is predicted to not belong to the top-k classes. True and false refer to the correctness of the prediction, and i refers to the ith class. We note that, as per the codebase, the authors attempt to calculate the per-class accuracy as

$$\frac{TP_i}{TP_i + FP_i}.$$

However, this is not calculating the per-class accuracy but rather the per-class precision. While precision is a highly useful metric for providing a better understanding of how often predictions for the positive class(es) are correct, it is an entirely different metric from accuracy, which evaluates the correctness of predictions for all classes. Accuracy should, instead, be calculated as

$$\frac{TP_i + TN_i}{(TP_i + FP_i) + (TN_i + FN_i)},$$

thus accounting for all true predictions made by the model, not only true positives.

The published figures are not in themselves numerically incorrect, but rather provide discrepant labels that may mislead readers regarding their interpretation of the results. We correct this discrepancy by calculating the precision as the authors report in their paper, which provides a better understanding of intra-class accuracy since precision is a measure of the accuracy of positive predictions.

5.2. Experimental Hyperparameters and Evaluation Metrics

To assess the performance of our model, we evaluate the precision of the model on a test dataset of 2,000 video samples. As briefly discussed in Subsection 5.1, we evaluate three different types of precision, being the top-k precision metrics, where k is either 1, 5, or 10. This metric signifies whether the ground truth word lies within the top k predicted words for the example, with $k = 1$ referencing the word itself. We also reference the semantic similarity of words, where the notion of closeness is determined by measuring the cosine similarities of all pairs of word vectors within the dataset. The cosine similarity between word vectors A and B is calculated as

$$\text{Cosine Similarity} = \frac{A \cdot B}{||A||||B||}.$$

Thus, the words closest to the ground truth words are the word whose cosine similarities are the largest.

In addition, we carefully tuned our hyperparameters for optimization. We selected the Adam optimizer, a standard, robust, and versatile optimizer that is applied within a variety of problem settings. Furthermore, we worked with a relatively small mini-batch size of 6. This small batch size allows for more accurate albeit slower training, but since the dataset was relatively small and contained a subset of 100 classes, execution of the code with this batch size still proved computationally efficient. We used a learning rate of 10^{-3} with a decay of 10^{-8} to allow the model to overcome plateaus that were often experienced during training. Lastly, we applied dropout with a probability of $p = 0.25$ to aid in regularizing the model.

5.3. Model Performance

We highlight the performance on the WLASL100 subset of our data (consisting of the top-100 most common words) via the top-k precision metrics as follows:

Model	Top-1	Top-5	Top-10
3DConvNet	0.22	0.35	0.46
3DConvNet+SelfAttn	0.23	**0.42**	**0.55**
3DConvNet+SE	**0.24**	0.40	0.53

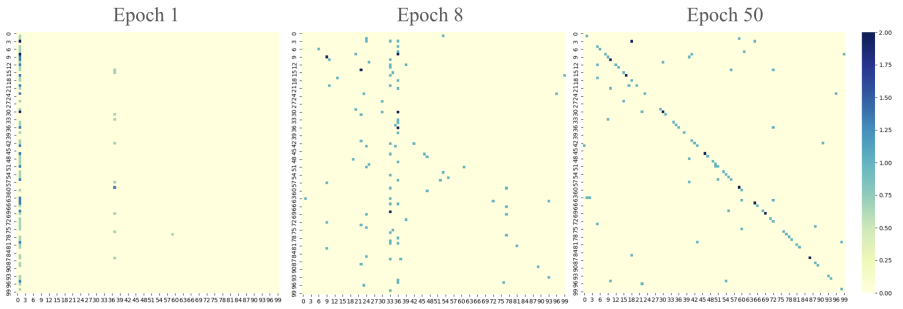

Figure 3. Confusion matrices (heatmap) of the 3D convolutional model during training for epochs 1, 8, and 50.

Ground Truth:
"candy"

Prediction:
"cool"

Figure 4. Individual signing the ground truth word (candy) versus the predicted word (cool) during test time.

Ground Truth:
"dog"

Prediction:
"thin"

Figure 5. Individual signing the ground truth word (dog) versus the predicted word (thin) during test time.

We can see that for each value of k in the top-k precision metrics, the addition of self-attention and squeeze-and-excitation blocks improves the precision. In the case of squeeze-and-excitation blocks, we surmise that the precision improves because they aid in modelling inter-channel dependencies and re-weight channels to focus on more important features, which is useful in the multi-modal case of sign language. Moreover, for self-attention blocks, we suspect that the precision improves because self-attention aids in capturing the long-range dependencies between frames, since one sign lasts the length of the entire video example. It is likely that self-attention does better than squeeze-and-excitation because the latter may struggle with spatial dependencies.

Note that the increase in precision is more pronounced for the top-5 and top-10 precision metrics. This highlights how, although squeeze-and-excitation and self-attention can improve the classification precision for the ground truth, the improvements are most pronounced in the model's ability to better capture general semantic relationships between words.

5.4. Analysis

The convolutional follow interesting trajectories in training when classifying examples. We illustrate this by investigating a heatmap of several different epochs on the validation set for the 3D convolutional model, as depicted by Figure 3.

Notice how, as the number of epochs increase, the prediction of correct words expectedly increases as evidenced by the increasingly darker diagonal. However, we also notice that the heatmap starts to illustrate semantic similarities between words, allowing us to see hints of what the semantically nearest words may be. For instance, looking at Class 3 during Epoch 50, which corresponds to the word "before," we see that one of its nearest often predicted neighbors is "last" in Class 65. "Last" happens to be semantically close to "before" with a cosine similarity of 0.678, which is the third closest word to "before" in the WLASL100 dataset.

We make another interesting observation during the early to middle stages of training. As observed in the heatmap during Epoch 8, we see that Class 33, which corresponds to the word "blue" in American Sign Language, is often incorrectly predicted for a wide range of words. When we look at how "blue" is signed in Figure 6, we notice that the right hand is placed in an open and relaxed configuration that serves as a general template for many other signs, like "fine," which has a close appearance to "blue." The model overgeneralizes in the early to middle stages of training and predicts "blue" since it has similar positional foundations to other words in the vocabulary.

Additionally, in classifying examples, our models encounter several interesting failure cases. One failure case for all models is that of the class for "candy," which is frequently mispredicted as "cool" during testing. We can see the reason for the failure case by inspecting the hand posi-

Figure 6. Individual signing “blue” (top left), “fine” (top right), “who” (bottom left), and “candy” (bottom right). These examples are all predicted to be ”blue” in early-to-mid stage training.

tions illustrated in Figure 4, in which the hand is up near the right side of the face in both images, which are both in a fist. Another failure case is that observed for the class “dog,” which is mispredicted as “thin.” In Figure 5, we can see that the hands are oriented with the pointer finger and thumb out. Although the hand positions differ subtly, the models struggle to detect the differences between them. This highlights the difficulty in feature engineering and in managing the complexity of the input that is inherent to ASL videos.

5.5. Discussion

Although squeeze-and-excitation and self-attention blocks have shown to be useful in classification, there is certainly substantial room for further improvement, especially considering that this is a model that has a real-world application, as is the case with all models designed for sign language translation.

One limitation that we feel our models have encountered is that, despite the addition of more complex layers, the feature space of the videos is quite large, and videos feature a lot of complexity in the form of ethnicity, body type, clothing, hand positioning and size, and even background color. While this level of representation is absolutely necessary to guarantee an equitable model, our lack of attention to the feature space made making substantial improvements difficult. Perhaps one solution to this problem could be drawing more focus on the individual by using a form of facial tracking and hand tracking so that the model is not unnecessarily distracted by irrelevant and frequently changing features.

Another limitation that our models have encountered is that the parameter space of self-attention and squeeze-and-excitation blocks is particularly large. Although we desired to use a small batch size of only 6 for training regardless, using batch sizes any larger resulted in out-of-memory errors on a computational device with a standard disk quota. Therefore, our proposed modification of the model, while effective from a numerical perspective, lacks the computational efficiency necessary for easy accessibility with respect to training. One solution would be to further explore the optimization of these layers or running the model of more advanced hardware, although the latter option is not always feasible.

6. Conclusion and Future Work

In this paper, we highlighted the mathematical error presented by Li et al. in computing the accuracy. We resolved the issue by adjusting the accuracy metric to be mathematically consistent with its definition and computing the *precision* as the authors may or may not have intended to do. We then implemented adjustments to the existing 3D convolutional architecture proposed by the authors by adding two types of blocks at each point in the model: self-attention and squeeze-and-excitation blocks.

Self-attention blocks aiding in capturing the long-range dependencies present in sign language video, whereas the squeeze-and-excitation blocks emphasize the importance of relevant features like facial expressions and hand movements by rescaling the channel-wise weights. We illustrated how these blocks indeed augment the performance of the baseline model, but still run into interesting failure cases that were manually inspected in order to understand the similarities between the mispredicted class and the ground truth class.

In future works, we aim to test the architecture against the WLASL2000 collection–consisting of 2000 classes in total–when we obtain access to more computational resources. The goal of testing under this dataset is to assess the generalizability of our results to unseen data within the ASL lexicon. Additionally, with more computational resources, we will explore training with larger batch sizes that would be more feasible with more than one GPU. Lastly, we aim to address the limitations in our current architecture by considering the feature space, taking into consider facial expressions, hand tracking, and general body motions. This will enable us to represent the translational variants that are common among the diverse array of signers who use ASL.

References

[1] V. Athitsos, C. Neidle, S. Sclaroff, J. P. Nash, A. Stefan, Q. Yuan, and A. Thangali. The american sign language lexicon video dataset. *2008 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, pages 1–8, 2008. 3

[2] B. Fang, J. Co, and M. Zhang. Deepasl: Enabling ubiquitous and non-intrusive word and sentence-level sign language translation. In *Proceedings of the 15th ACM Con-*

ference on Embedded Network Sensor Systems, SenSys '17, New York, NY, USA, 2017. Association for Computing Machinery. 2

[3] L. Gao, H. Li, Z. Liu, Z. Liu, L. Wan, and W. Feng. Rnn-transducer based chinese sign language recognition. *Neurocomputing*, 434:45–54, 2021. 1

[4] J. Huang, W. Zhou, H. Li, and W. Li. Sign language recognition using 3d convolutional neural networks. In *2015 IEEE International Conference on Multimedia and Expo (ICME)*, pages 1–6, 2015. 2

[5] N. B. Ibrahim, H. H. Zayed, and M. M. Selim. Advances, challenges and opportunities in continuous sign language recognition. *Journal of Engineering and Applied Sciences*, 15(5):1205–1227, 2020. 2

[6] B. Joksimoski, E. Zdravevski, P. Lameski, I. M. Pires, F. J. Melero, T. P. Martinez, N. M. Garcia, M. Mihajlov, I. Chorbev, and V. Trajkovik. Technological solutions for sign language recognition: A scoping review of research trends, challenges, and opportunities. *IEEE Access*, 10:40979–40998, 2022. 2

[7] O. Koller, O. Zargaran, H. Ney, and R. Bowden. Deep sign: Hybrid cnn-hmm for continuous sign language recognition, 20160919 - 20160922. 2

[8] C. K. Lee, K. K. Ng, C.-H. Chen, H. C. Lau, S. Y. Chung, and T. Tsoi. American sign language recognition and training method with recurrent neural network. *Expert Systems with Applications*, 167:114403, 2021. 1

[9] D. LI, C. Rodriguez, X. Yu, and H. LI. Word-level deep sign language recognition from video: A new large-scale dataset and methods comparison. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, March 2020. 1, 2, 3, 4

[10] A. Martinez, R. Wilbur, R. Shay, and A. Kak. Purdue rvl-slll asl database for automatic recognition of american sign language. *Proceedings. Fourth IEEE International Conference on Multimodal Interfaces*. 3

[11] G. A. Rao, K. Syamala, P. V. V. Kishore, and A. S. C. S. Sastry. Deep convolutional neural networks for sign language recognition. In *2018 Conference on Signal Processing And Communication Engineering Systems (SPACES)*, pages 194–197, 2018. 2