Word-Level Sign Language Recognition using Deep Learning Methods and Beyond

Sahil Surapaneni, Yuting Liao, Carmen Giger, Vaarun Muthappan, Jiaqi Wu

1. Introduction

Effectively using American Sign Language (ASL) is a pertinent problem for the roughly 600,000 deaf people in the United States [1]. Communicating through sign language for them is necessary but still difficult. For example, ineffective communication with their health care providers has led to health care avoidance among the deaf population [11]. Hence, there is a significant need in the wider community for a way to accurately understand sign language.

Error rates in computer vision have decreased significantly since 2010 [4], hence computer vision models may provide an effective way to translate sign language for people with no familiarity in sign language. However, many existing sign language models are out of date and in-comprehensive [2]. ASL also depends on body movements and motions, which many models trained on images do not capture, so it is important to make a model for videos. Therefore, we explore several models in this paper to find an effective solution for the problem of interpreting ASL.

Accurate computer vision models for recognising ASL could also be applied to other sign languages through transfer learning or using similar models, making the task more important as over 430 million people worldwide have 'disabling' hearing loss, with this number set to increase to 700 million by 2050 [10].

In our work, we decided to explore the ways in which machine learning models in video-related recognition tasks can be applied to recognizing sign language words. We also develop a baseline model to compare the performances to.

2. Related Work

Most of the works developed to address this issue have been contact-based (such as using sensor gloves) or vision-based systems. The latter is cheaper and can be expanded upon using deep learning, so we explored relevant papers that used vision-based machine learning algorithms and built our approaches upon those.

2.1. Recurrent Neural Networks

A recurrent neural network (RNN) is a type of network that uses sequential data or time-series data. RNNs are commonly used for language translation and image captioning for temporal problems. In work published by Sarfaraz Masood et al. [8], they implemented two approaches that input data into a conventional neural network (CNN) to either produce predictions or convolved features about corresponding frames that were then pooled and fed into an RNN. The model that used the pooling had over 95% accuracy and showed that CNN along with RNN can be successfully used to train and classify sign language gesture videos.

2.2. Frame Attention Networks

Frame attention networks (FAN) can be used to highlight some discriminated frames in an end-to-end framework automatically. In a paper using FANs for facial expression recognition [3], researchers were able to show that their FAN with self-attention weights on individual features has superior performance to other CNN-based methods for facial expression recognition. Compared to the hand movements involved in sign language, though, facial features generally stay in a fixed position. In contrast, hand movements should involve more spatial information to be captured and remembered.

2.3. 3D Convolution Networks

3D convolution networks are similar to 2D ones; however, they also have spatio-temporal filters. They can be used as an alternative to RNNs to capture data with high-level sequence variation such as a gesture. In a paper by Carrera and Zisserman [5] the two explore and compare a series of models, including 2D CNNs with an RNN, 3D CNNs, different fusion, and pre-training methods. The best-performing model was trained on the ImageNet data set and pre-trained on the Kinetics data set. This showed that (1) pre-training could be beneficial, (2) training metrics from one set can be transferred over to an extent for further training on similar data sets, and (3) Spatio-temporal feature extractions from videos improve the performance of action classification.

3. Approach

3.1. Dataset

We used the Word-Level American Sign Language (WLASL) dataset obtained from the official github [6], which consists of sign videos from youtube and other sources. The videos are of relative high quality, with the signer standing in the center and facing directly at the camera. We chose this dataset since it is the largest dataset for Word-Level American Sign Language Recognition. However, we found many of the videos they used in their original paper were now missing from the dataset, which became a source of issue for us. For consistency sake, we used the data for the top 100 appearing glosses (words in ASL). In their original dataset, they had 2038 videos across the 100 glosses, while only 991 were available when we downloaded. the dataset.

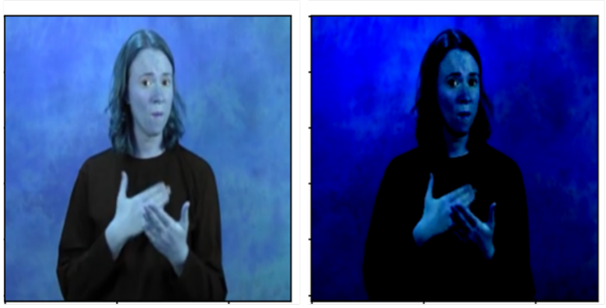

Figure 1. The original frame and the preprocessed frame

3.2. Deep-Learning Models

We divided our work into 2 main parts: building the models on the pre-extracted feature using ResNet34, and building the models on the video frames directly.

Our attempted models can be further divided into these categories:

1. Pure RNN model using different layers of LSTM and different directionality.

2. RNN model with 1-2 attention layers at different location (e.g., applying attention to all hidden states after the first LSTM layer, applying attention to hidden states after the last LSTM layer).

3. Pure attention model consisting of stacks of attention blocks. Each block multiplies the inputs with a weight matrix, followed by an attention matrix A, where Aij represents the importance of jth frame when computing for ith frame, inspired by the original paper [6].

4. 3D convolution network following an inception architecture that's trained from scratch, without preloaded weights trained from ImageNet dataset. The I3D architecture strictly follows the work by Carreira and Zisserman [5].

For each model, we explored the best combination of hyperparameters, such as the depth of LSTM layers, the hidden size, and the activation functions.

We used cross entropy loss function for a multiclass classification, with a learning rate of 0.0001 - 0.001, a batch size of 16 - 32 and the number of epochs ranging from 100 to 300 depending on the speed of the convergence of different models. The optimizers we used were Adam and AdamW. Due to the limited amount of videos we were able to download, we manually shuffled the videos and split them into 85% train, 10% validation, and 5% test sets.

Our training procedure involved a forward pass on the batched inputs, followed by the loss calculated on the outputs with softmax activation. At the end of each epoch, we calculated the validation accuracy for the model.

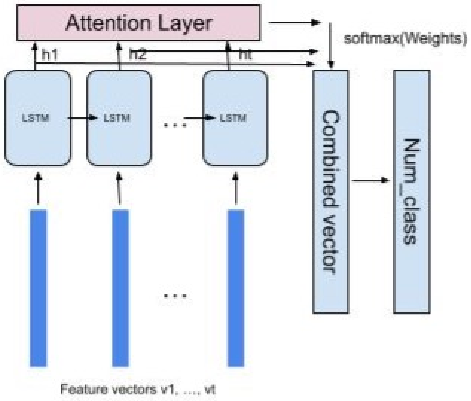

Figure 2. An example of RNN with Attention architecture

An illustration - In Fig 2, the pre-extracted feature vectors using ResNet34 are loaded into the dataset and fed into the LSTM layer, whose hidden states at each time point are fed into an attention layer to compute weights assigned to each hidden vector. These weighted hidden states sum to a combined vector with the same size as the hidden state. Finally, we feed the vector into a fully connected layer whose output dimension equals to the number of class (100).

3.3. Baseline Model

Due to the relatively lesser amount of videos available to us as compared to the original dataset, we wanted to compare these deep learning methods to a method that does not rely on training models, but rather attempts recognition on solely the geometry of the tensor space the videos exist in. The inspiration for this method was the paper ”Action Classification on Product Manifolds” published in 2010 [7]. The action videos are decomposed into three factor manifolds using Higher Order Singular Value Decomposition (HOSVD), the product of which comprises the product manifold.

HOSVD on an N order tensor is done as

$$A = S \times_1 V^1 \times_2 V^2 ... \times_n V^n \tag{1}$$

where S is a core tensor of singular values and V^n is the orthogonal matrix spanning the column space of the n unfolded matrix, and $\times_k$ is mode-k multiplication. Each V^n is a factor manifold of the video. The difference between two factor manifolds is the canonical angle θ. The canonical angles can be calculated recursively by:

$$\begin{aligned} \theta_k &= \cos^{-1}(x_k^\top y_k) \\ x_0 &= V_A, y_0 = V_B \end{aligned} \tag{2}$$

Gesture recognition is achieved by choosing the classification that minimizes the chordal distance between A and B, defined as:

$$dist_{chordal}(A, B) = \|\sin(\Theta)\|^2 \tag{3}$$

Where Θ is a vector of the canonical angles between the three factor manifolds of A and B. This process (outlined in Figure 3, from [9]) is described in [7], which describes a nearest-neighbor classifier based on classifying A as the same class as whichever B minimizes this chordal distance(Eq.3). Originally intended for human gesture recognition, we wanted to see how it would perform for sign language recognition.

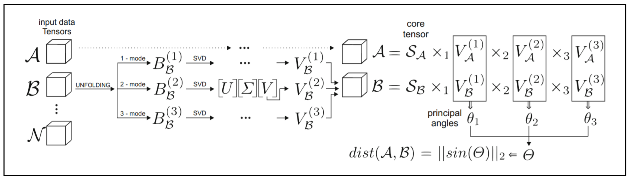
input data
Tensors
$\mathcal{A}$
$\mathcal{B}$
$\mathcal{N}$
UNFOLDING
1 - mode
2 - mode
3 - mode
$B_{\mathcal{B}}^{(1)}$
$B_{\mathcal{B}}^{(2)}$
$B_{\mathcal{B}}^{(3)}$
SVD
SVD
SVD
$[U][\Sigma][V]$
$V_{\mathcal{B}}^{(1)}$
$V_{\mathcal{B}}^{(2)}$
$V_{\mathcal{B}}^{(3)}$
core
tensor
$\mathcal{A} = \mathcal{S}_{\mathcal{A}} \times_1 V_{\mathcal{A}}^{(1)} \times_2 V_{\mathcal{A}}^{(2)} \times_3 V_{\mathcal{A}}^{(3)}$
$\mathcal{B} = \mathcal{S}_{\mathcal{B}} \times_1 V_{\mathcal{B}}^{(1)} \times_2 V_{\mathcal{B}}^{(2)} \times_3 V_{\mathcal{B}}^{(3)}$
principal
angles
θ_1
θ_2
θ_3
$dist(\mathcal{A}, \mathcal{B}) = ||sin(\Theta)||_2 \Leftarrow \Theta$

Figure 3. Action classification on product manifold schema *X*, *Y* and temporal dimension *T*.

4. Experiment

Metrics - We used the top-30 accuracy on the preselected validation and test video datasets as our metric. Considering the insufficient amount of training data, we decided to use top-30 accuracy as our metric, which is a quantitative measure. We regard this appropriate because the meaning of words can be closely related, so that the incorrect classification of words can still result in relatively accurate interpretation as long as their classification is close to the actual label.

4.1. Results

Among different types of models we have attempted, we have selected to include the models with relatively satisfactory top-30 accuracy in this paper. They are pure RNN, RNN with attention, and pure attention models.

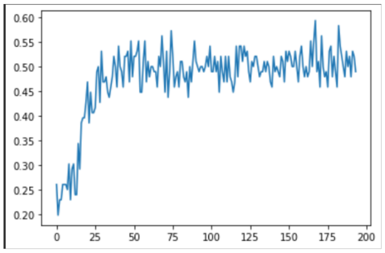

Figure 4. Top-30 validation accuracy for RNN with Attention model, which stabilizes at around 55%

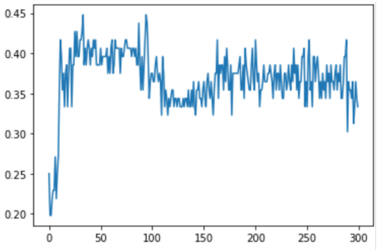

Figure 5. Top-30 validation accuracy for pure RNN model, which stabilizes at around 37%

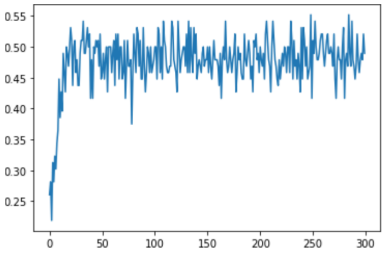

Figure 6. Top-30 validation accuracy for pure attention model, which stabilizes at around 50%

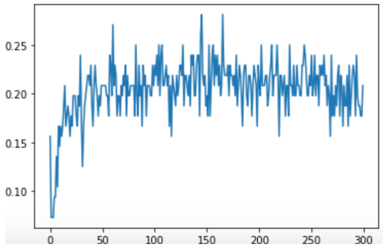

Figure 7. The best Top-10 validation accuracy was from pure attention model

By comparison we found the Baseline Method using Product Manifold did not perform very well on the WLASL top-100 gloss data set with a **top-1 accuracy of 9% and a top-30 accuracy of just 27%**. Due to the fact that this method does not rely on extensive training, it took 4 min 32 sec of Wall Time to factor all the videos and classify the test data. We rightly hypothesized that this method would perform poorly for word-level sign language recognition since the difference between different words are just small changes in hand motion and body position. The product manifold method performs best when the differences between actions are very different geometrically (i.e jumping up and down vs running across the screen). This goes to prove why sign-language recognition is such an important topic of research in computer vision currently. It is a complicated problem that cannot be solved trivially, yet has a very real impact on millions of people world wide.

5. Implementation

We slightly tweaked the existing preprocessing procedures from the github of the WLASL dataset [6] to get our dataset. The preprocessing involved reading in each video frame and then scaling and croping them to (224, 224, 3).

All the deep-learning models tested in this paper were written from scratch. The inspiration for the models we used came from a variety of papers published on Sign Language Recognition and video classification.

The models were implemented in PyTorch, and the pre-trained models such as ResNet34 were obtained from the torchvision models.

The baseline product manifold method was written from scratch in Python based on the algorithm outlined in the original paper about it [7].

6. Discussion

As shown by the results section, all models tested have a higher performance than the baseline model. More specifically, RNN with attention layers produces better performance than pure RNN models, which suggests that focusing more attention on specific time periods over the others can be beneficial to the final classification. Similarly, models that only use attention blocks perform better than pure RNN models. This shows that instead of remembering information from the previous time points, the ability to capture relationships among different frames may be more significant in video classification. This is further shown in Fig.7, where the pure attention mode produces the best performance on the top-10 validation accuracy among all the models ($>20\%$).

The I3D model failed to produce an accuracy better than simple guessing, hence this model is unsuitable for training the current dataset from scratch. A possible reason could be that the architecture was too deep and broad for such a small set of training videos. As suggested by Li *et al.* [6], a better application of the I3D model would be using the pre-trained weights trained on the ImageNet and Kinetics dataset, then fine-tune with the WLASL dataset.

6.1. Limitations

All the models were implemented and trained on Google Colab, meaning that there was a great limitation on the computing resources, including memory, GPU, and RAM. On top of that, the automatic timeout on this platform also prevented models to be trained for longer than 12 hours. As a result, we faced many challenges when training deeper and more complex models such as Pose-RNN and I3D, which were originally trained on 16 or 32 bit GPUs and would benefit from running over a larger number of epochs.

There was also been a lack of training videos due to unsuccesful downloads from the resource website. With around 700 videos only to train on 100 labels, all attempted models suffered from overfitting, as they performed well in their top-1 training accuracy, but poorly on the validation accuracy.

The use of existing feature extractors (ResNet34) instead of investigating more data pre-processing methods could have further prevented improvements in the results. For example, video samples from different individuals tend to have different characteristic lighting conditions and background colors - keeping the frames in RGB format without normalization might consequent in the models associating these unhelpful information with some labels. An attempted solution to this issue was to pre-process with optical flow. This method could be investigated further in future works.

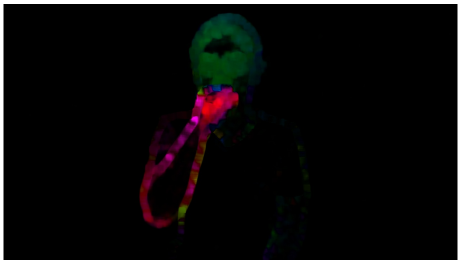

Figure 8. An example of optical flow image that captures the moving body parts

References

[1] Deaf employment reports, Jan 2021. 1

[2] Danielle Bragg, Oscar Koller, Mary Bellard, Larwan Berke, Patrick Boudreault, Annelies Braffort, Naomi Caselli, Matt Huenerfauth, Hernisa Kacorri, Tessa Verhoef, Christian Vogler, and Meredith Ringel Morris. Sign language recognition, generation, and translation: An interdisciplinary perspective. *CoRR*, abs/1908.08597, 2019. 1

[3] Kai Wang Yu Qias Debin Meng, Xiaojiang Peng. Frame attention networks for facial expression recognition in videos. 2019. 1

[4] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016. http://www.deeplearningbook.org. 1

[5] Andrew Zisserman Joao Carreira. Quo vadis, action recognition? a new model and the kinetics dataset. 2018. 1, 2

[6] Dongxu Li, Cristian Rodriguez, Xin Yu, and Hongdong Li. Word-level deep sign language recognition from video: A new large-scale dataset and methods comparison. In *The IEEE Winter Conference on Applications of Computer Vision*, pages 1459–1469, 2020. 2, 4

[7] Yui Man Lui, J. Ross Beveridge, and Michael Kirby. Action classification on product manifolds. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 833–839, 2010. 2, 3, 4

[8] Sarfaraz Masood, Adhyan Srivastava, Harish Chandra Thuwal, and Musheer Ahmad. Real-time sign language gesture (word) recognition from video sequences using cnn and rnn. 2018. 1

[9] Agnieszka Michalczuk, Kamil Wereszczy´nski, Jakub Segen, Henryk Josi´nski, Konrad Wojciechowski, Artur Bak, Slawomir Wojciechowski1, Aldona Drabik1, and Marek Kulbacki. Manifold methods for action recognition. pages 613–622, 2017. 3

[10] World Health Organisation. Deafness and hearing loss. 1

[11] Kathy M Pendergrass, Susan D Newman, Elaine Jones, and Carolyn H Jenkins. Deaf: A concept analysis from a cultural perspective using the wilson method of concept analysis development. *Clin. Nurs. Res.*, 28(1):79–93, Jan. 2019. 1