Proceedings of the 12th Conference on Language Resources and Evaluation (LREC 2020), pages 6018–6024
Marseille, 11–16 May 2020

Sign Language Recognition with Transformer Networks

Mathieu De Coster[1], Mieke Van Herreweghe[2], Joni Dambre[1]
[1]IDLab-AIRO – Ghent University - imec, [2]Ghent University
[1]Technologiepark-Zwijnaarde 126, Ghent, Belgium, [2]Blandijnberg 2, Ghent, Belgium
{mathieu.decoster, mieke.vanherreweghe, joni.dambre}@ugent.be

Abstract

Sign languages are complex languages. Research into them is ongoing, supported by large video corpora of which only small parts are annotated. Sign language recognition can be used to speed up the annotation process of these corpora, in order to aid research into sign languages and sign language recognition. Previous research has approached sign language recognition in various ways, using feature extraction techniques or end-to-end deep learning. In this work, we apply a combination of feature extraction using OpenPose for human keypoint estimation and end-to-end feature learning with Convolutional Neural Networks. The proven multi-head attention mechanism used in transformers is applied to recognize isolated signs in the Flemish Sign Language corpus. Our proposed method significantly outperforms the previous state of the art of sign language recognition on the Flemish Sign Language corpus: we obtain an accuracy of 74.7% on a vocabulary of 100 classes. Our results will be implemented as a suggestion system for sign language corpus annotation.

Keywords: sign language recognition, deep learning, corpus annotation

1. Introduction

Sign language recognition (SLR) is a complex problem. Sign languages are, after all, complex visual languages. Generally, one can say that sign languages have five parameters. A sign is distinguished by hand shape, hand orientation, movement, location, and non-manual components such as mouth shape and eyebrow shape. However, these parameters do not necessarily fully identify signs: two signs can have the same execution but different meaning. Furthermore, identical signs are often executed differently based on several factors, such as age, gender, the dominant hand, and dialects. Additionally, there is a high degree of co-articulation in sign languages: both hands can produce different signs at the same time.

SLR is typically tackled using machine learning approaches. Deep learning in particular has proven to be very powerful for tasks such as image classification (Krizhevsky et al., 2012) and neural translation (Vaswani et al., 2017). Deep learning algorithms require large datasets in order to learn meaningful representations that generalize well to unseen data, especially for complex problems such as SLR. While large video corpora are available for several sign languages, they consist mostly of unlabeled data. Labeling the sign language corpora is a time-consuming process that requires the annotator to know sign language and its specific phonetic and phonological properties. As a consequence, the portion of a video corpus that is labeled grows only slowly.

Larger datasets exist (Chai et al., 2014; Huang et al., 2018; Vaezi Joze and Koller, 2019), but several consist of recordings of persons performing signs in repetition (Ronchetti et al., 2016; Ko et al., 2018). These datasets are often not representative of real world sign language, as they contain artificial repetitions of isolated signs (Bragg et al., 2019). Because the accuracy - the measure that is most commonly used to assess the performance of sign classification systems - is saturated on such datasets when using deep learning methods (Konstantinidis et al., 2018; Ko et al., 2018), it is more challenging and interesting to perform SLR on real sign language data. This also paves the way for sign language translation in the future.

The question can be posed if a deep learning system can be used to speed up the annotation process of sign language corpora, in order to obtain more labeled data. This could for example be done by creating a suggestion system for corpus annotators, that provides a list of likely glosses given a selected video fragment of a sign. In this work, we use a Long Short-Term Memory (LSTM) network as a baseline. We then present and compare three methods based on the transformer network architecture, that consistently outperform this baseline. The methods will be applied in the creation of the proposed suggestion system.

2. Related Work

Three sub-domains can be distinguished within SLR: isolated SLR, also known as sign classification, continuous SLR, and sign language translation. In isolated SLR, each sample corresponds to a single sign. In continuous SLR, samples contain one or more signs, and the task is to locate and recognize the signs. Sign language translation is translation from sequences of signs to sentences in a written language, such as English. In this work, we focus on isolated SLR specifically, but give an overview of methods used for SLR in general, because methods can be applied to several sub-domains.

The success of Hidden Markov Models (HMMs) in automatic speech recognition motivated their use by SLR researchers (Vogler and Metaxas, 1997; Starner et al., 1998; Bauer and Kraiss, 2001). In recent years, however, the domain has largely moved towards deep learning, because of its performance in related domains (in particular computer vision and again, speech recognition). Spatio-temporal models such as 3D Convolutional Neural Networks (CNNs) have been used for continuous SLR (Pigou et al., 2017; Jing et al., 2019). Hybrid architectures which combine spatial models (2D CNNs) with temporal models (LSTMs or HMMs) also yield good results for continuous SLR (Koller et al., 2016b; Koller et al., 2017; Ye et al., 2018) and sign language translation (Cihan Camgoz et al., 2018).

Previous work has attempted to classify isolated signs in sign language corpora using learned features and 2D CNNs (Pigou et al., 2014; Pigou et al., 2016). However, sign language corpus datasets are small for the problem complexity, and cross-domain learning is required to obtain adequate accuracy values. Even then, there is room for improvement. Recently, OpenPose was introduced as an open-source human pose estimation system (Cao et al., 2017). OpenPose extracts relevant information for sign language (i.e., position of important body parts) and can be used as a feature extractor (Konstantinidis et al., 2018; Ko et al., 2018). This is quite similar to the approach taken by researchers before such generic pose estimation systems were available (Ong and Bowden, 2004; Charles et al., 2014). Currently, OpenPose is the only full-body pose estimation technique and is therefore an obvious choice for SLR, which relies not only on hand shape, location and orientation, but also on non-manual components such as mouth shape. Other pose estimation techniques exist, but only recognize, for example, body keypoints (Fang et al., 2017) or hand keypoints (Mueller et al., 2018).

3. Deep Learning Background

SLR is a spatio-temporal problem. Methods from computer vision and natural language processing are typically applied to tackle it.

3.1. Convolutional Neural Networks

The most popular computer vision algorithm in deep learning is the CNN. Krizhevsky et al. (2012) showed that CNNs can outperform engineered feature extractors. Since then, they have been ubiquitous in image and video recognition tasks.

3.2. Attention and Transformers

Transformer networks, introduced by Vaswani et al. (2017), obtain state of the art performance for many natural language processing tasks (Vaswani et al., 2017; Devlin et al., 2019). These networks consist of repeated multi-head attention blocks and point-wise feed-forward layers.

A query, key and value matrix are constructed from the input features through trainable linear transformations:

$$Q = XW^Q,$$
$$K = XW^K,$$
$$V = XW^V,$$

where W^Q, W^K and W^V are trainable weight matrices. Then, scaled dot-product attention (Vaswani et al., 2017) is applied. For this, the query and key matrices are used to calculate an attention weight matrix, which is multiplied with the value matrix to obtain the attended output O. This is computed as

$$O = \text{softmax}\left(\frac{QK^\top}{\sqrt{d_e}}\right)V. \quad (1)$$

The denominator, with d_e the dimensionality of the input features, is used to scale the dot products such that the softmax is not saturated for large embeddings (Vaswani et al., 2017).

Multi-head attention applies attention several times in parallel on sub-regions of the input space. Every attention calculation is performed in a so-called attention head. Each of the n_h heads looks at a subset of the original input: for $X \in \mathbb{R}^{T\times N}$ (T: sequence length, N: number of features), Q^h, K^h and V^h are elements of $\mathbb{R}^{T\times d_e}$ in head h, with $d_e = N/n_h$.

The outputs of all heads in a layer are concatenated, normalized using layer normalization (Ba et al., 2016), and processed by a residual point-wise feed-forward network as input for the next layer. Transformers typically consist of several of these layers. An illustrated example of a multi-head attention layer is shown in figure 1.

3.3. Video Transformer Networks

Recently, CNNs and transformers have been combined to perform action recognition on videos (Kozlov et al., 2019). Specifically, they use a 2D CNN which learns vector representations in latent space from input frames as a feature extractor. The extracted features are used as input to a network consisting of 4 stacked 8-head attention layers. The trainable weight matrices are initialized using the Glorot normal distribution (Glorot and Bengio, 2010).

The network outputs predictions per frame. These are averaged across time to obtain the final clip prediction. Kozlov et al. (2019) call their network architecture the Video Transformer Network (VTN). They note that VTNs do not lead to optimal performance in action recognition, which is currently obtained using 3D CNNs (e.g., the network by Tran et al. (2018) for the Sports-1M dataset (Karpathy et al., 2014)). However, because of the lower number of parameters, VTNs require less computation time and are less prone to overfitting than 3D CNNs. The latter is particularly important for sign language datasets which are typically smaller than action recognition datasets.

4. Experiments

4.1. Data

As the goal of this work is to present an annotation tool for sign language corpora, we use such a corpus as dataset to train and evaluate our models. In particular, we evaluate our method on the Flemish sign language (VGT) corpus (Van Herreweghe et al., 2015).

The Corpus VGT consists of 140 hours of video data. Each video has a spatial resolution of 960 by 540 pixels and a temporal resolution of 50 frames per second (FPS). We consider 100 classes, corresponding to 104 glosses. Some classes correspond to two glosses rather than one because they are visually indistinguishable. If two data classes are visually very similar but have distinct labels, the neural network will be confused. For an annotation suggestion tool, both glosses are valid suggestions: they can simply both be shown at the same time. In total, we obtain 18730 samples from 67 native signers.

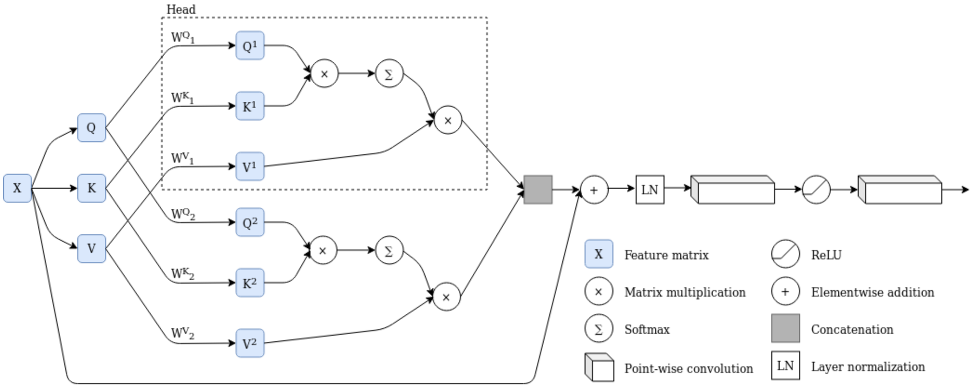

Figure 1: A multi-head attention layer with 2 heads.

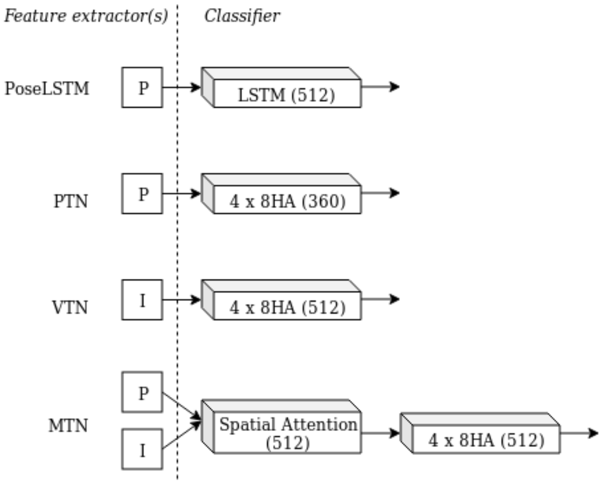

Figure 2: The network architectures of the different methods. The classifiers (on the right) obtain their input features from one or more feature extractors (on the left). “P” represents OpenPose and “I” represents a 2D CNN.

4.2. Methodology

We explore four methods for isolated SLR. Each method is based on a combination of one or more feature extractors and a classifier. The feature extractor is either OpenPose (Cao et al., 2017) or a 2D CNN pre-trained on ImageNet (Deng et al., 2009) that is fine-tuned on sign language data during the classification task. The classifier architecture is either an LSTM, or a deep multi-head attention network like the one shown in figure 1. An overview of the four methods is shown in figure 2. We train and evaluate the methods on the same dataset, but tune hyperparameters individually in order to obtain optimal performance for each of the methods. We apply each method to the problem of isolated sign language recognition and compare how they perform. They are described in detail in the next sections.

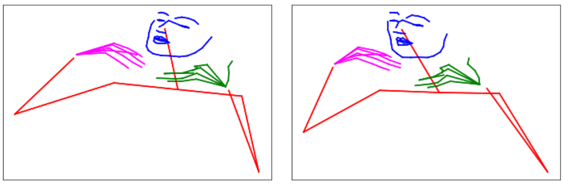

Figure 3: An example of the output of the pre-processing and data augmentation pipeline used for OpenPose keypoints. On the left: the original keypoints. On the right: the transformed keypoints.

We split the data into a train, validation and test set. The training set consists of 70% of the original dataset, the validation set of 10% and the test set of 20%. This yields 13077 training samples, 3743 test samples, and 1910 validation samples. The neural networks are trained on the training set, tuned on the validation set, and evaluated on the test set.

We use random temporal cropping to select 16 frames as a form of temporal data augmentation for all four methods. If a sample is longer than 16 frames, a random start index is chosen, and 16 consecutive frames are used as input to the network. If a sample is shorter than 16 frames, it is padded by looping the sample. Finally, if the sample length is exactly 16 frames, the sample is used in its entirety as input to the network. We find that this performs similarly to using zero-padded variable length sequences and the shorter sequence length reduces computation time during training. At inference time, the entire clip is evaluated using a sliding window approach: non-overlapping sequences are chosen from each sample, and the accuracy is averaged across these windows of all samples.

All neural networks are trained and evaluated using PyTorch 1.3 (Paszke et al., 2017) on two Nvidia GeForce GTX 1080 Ti GPUs.

4.2.1. PoseLSTM

As a first experiment, we consider using LSTM networks with OpenPose features as a baseline to be able to evaluate the impact of multi-head attention on the accuracy of the neural network.

We use OpenPose as a fixed feature extractor. For every frame, OpenPose extracts 137 keypoints. 25 keypoints represent the body pose, 70 are facial keypoints, and there are 21 keypoints per hand representing the hand pose.

Every keypoint is a triplet (x, y, c), where x and y are rational numbers representing the 2D coordinates of the keypoint, and c is the confidence of OpenPose in the correctness of this keypoint. We use entire triplets as input features. The lower body is not in frame and is not relevant for sign language, so we decide to drop those keypoints. The facial keypoints from the body model are also removed, because that information is present in the keypoints of the face model. We keep 8 keypoints for the body and 120 in total per frame. As spatial pre-processing, we rotate the pose such that the shoulders are horizontal to account for seating position, and we standardize the body pose such that the length of the neck is 1. For data augmentation, we perform the following transformations. First, we introduce Gaussian noise on the keypoints, i.e., translating every keypoint by (x, y), where x and y are sampled from $\mathcal{N}(0, 0.005)$. Secondly, we randomly rotate both hands separately up to 20 degrees using the wrist keypoints as pivot points. These values were empirically found. An example result of this pre-processing and data augmentation pipeline is shown in figure 3.

These features are used as input to an LSTM with 512 hidden units with a positive forget gate bias initialization (Jozefowicz et al., 2015). We use the Adam optimizer (Kingma and Ba, 2015) with initial learning rate $\lambda = 1\mathrm{e}{-3}$, $\beta_1 = 0.9$, $\beta_2 = 0.999$, $\epsilon = 1\mathrm{e}{-8}$ and weight decay with a penalty of $1\mathrm{e}{-4}$. Learning rate decay is used to reduce the learning rate by a factor 10 whenever the validation accuracy does not improve by at least 0.01 for 10 epochs. The mini-batch size is set to 64. The network is trained for 100 epochs with early stopping.

4.2.2. Pose Transformer Network

The Pose Transformer Network (PTN) uses the keypoints extracted by OpenPose as input for a multi-head attention classifier with embedding size equal to the number of input features, which is 360.

We use the same settings as for the PoseLSTM with regards to the optimizer and learning rate decay.

4.2.3. Video Transformer Network

The third method uses the architecture proposed by Kozlov et al. (2019). The feature extractor is a framewise 2D CNN (in our case ResNet-34 (He et al., 2016)) that transforms each frame into a 512-dimensional vector. The classifier is a multi-head attention network with embedding size 512.

We use the standard pre-processing and data augmentation techniques from action recognition. Multi-scale cropping (Wang et al., 2015) is applied to obtain RGB images of 224 by 224 pixels. The images are then normalized by subtracting the mean μ of the ImageNet dataset on which the network is pre-trained and dividing by the standard deviation σ, with $\mu = (0.485, 0.456, 0.406)$ and $\sigma = (0.229, 0.224, 0.225)$. As in the other methods, temporal random cropping is used as temporal data augmentation to obtain 16 frames per sample.

We also use the Adam optimizer with the hyperparameters as described for the PoseLSTM. The initial learning rate differs: it is $\lambda = 1\mathrm{e}{-4}$. The mini-batch size is set to 16.

Method	Top-1	Top-3	Top-5	Top-10
PoseLSTM	54.55%	66.61%	72.75%	79.81%
PTN	61.73%	75.77%	81.12%	87.63%
VTN	73.39%	85.25%	89.19%	92.62%
MTN	74.70%	86.11%	89.81%	93.37%

Table 1: Top-n accuracy values for different models on the unseen test set.

4.2.4. Multimodal Transformer Network

The fourth and final method combines the OpenPose keypoints and the learned features as input for the classifier. In order to reduce the dimensionality of the feature space, which would otherwise be equal to $512 + 360 = 872$, we apply spatial scaled dot-product attention to learn which features are relevant for a given frame. We assume that there is quite some redundancy within the keypoints of a single frame, as well as between keypoints and RGB data. This dimensionality reduction technique allows the network to learn how to remove redundancy on a frame by frame basis. We modify the attention calculation in equation 1 in order to attend to spatial rather than temporal information:

$$\begin{aligned} Q &= PW_Q, \\ K &= FW_K, \\ V &= XW_V, \\ O &= V\text{softmax}\left(\frac{QK}{\sqrt{872}}\right), \end{aligned}$$

where P is the keypoint feature matrix, F is the image feature matrix and X the matrix of all input features. We obtain 512 features per frame, which is the same as for the VTN method. Each of these 512 features is a linear transformation of the original 872 input features, with the factors given by the attention weights.

The pre-processing and data augmentation approach is identical to the approach taken for the previous methods ($\lambda = 1\mathrm{e}{-4}$). The hyperparameters are initialized identically to the VTN method.

4.3. Evaluation

In a suggestion tool for corpus annotation, the number of suggestions could depend on the preferences of the annotator and is not fixed. Therefore, we look at top-3, top-5 and top-10 suggestions, as done by Pigou et al. (2016). We report on the top-n accuracy, with $n \in \{1, 3, 5, 10\}$. This is defined as the frequency with which the ground truth label is in the top n of network predictions.

5. Results

5.1. Comparison of the Architectures

The top-1, top-3, top-5 and top-10 accuracy measures per model are shown in table 1. The difference between PTN and PoseLSTM indicates that transformers outperform LSTM networks when used with OpenPose features. This suggests that transformers are also capable of outperforming LSTM networks on tasks other than machine translation, for which they were designed.

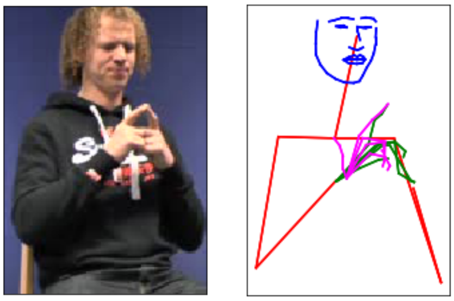

Figure 4: An example of a case where OpenPose fails. The keypoints of the left hand (green) are noisy and at incorrect locations. Meanwhile, the RGB frame is of reasonable quality and shows the hand in the correct position.

We notice a large difference in accuracy values between the PoseLSTM and PTN on the one hand and the other methods on the other hand. This indicates that OpenPose features on their own are not significantly powerful, as it is possible to learn better features directly from the data. We suspect that there is too much motion blur in the available data for OpenPose to detect keypoints with sufficient accuracy, and that information is therefore lost that is still present in the RGB data (e.g., keypoints are completely missed while the RGB data still gives a hint of what could be the correct hand shape). We present a failure case in figure 4, which clearly shows how OpenPose extraction may reduce the quality of the input features. A possible remedy for this problem would be to fine-tune OpenPose while training the network for the SLR task, rather than using it as a fixed feature extractor. This is beyond the scope of this work and left for future research.

We notice a smaller difference between the VTN and MTN than between the PTN and VTN. We suspect that the 2D CNN in the networks is already capable of extracting most of the relevant information by itself. However the increase in accuracy suggests that the OpenPose keypoints add information which can be useful for discriminating between classes.

5.2. Error Analysis

We now look at the confusion matrix of the MTN network (figure 5), more specifically at two pairs of signs. The largest confusion (38% misclassified) occurs between the signs for “BRIEF” (letter) and “ENVELOPPE” (envelope). This is a minimal pair in VGT: the signs differ only in a single parameter, in this case mouthing. A possible improvement for future work is to add side-channels to the neural network that predict the sign parameters. This is similar to the works of Cooper et al. (2012) and Koller et al. (2016a). The second largest confusion (36% misclassified) is between two variants of the sign “VAN” (of). The first variant is the general use of the preposition, while the second variant more specifically refers to the first person (“my”). These are similar in the signing of the dominant hand, but distinct in the non-dominant hand (which is why they are not grouped in a single class). Distinguishing between these two variants may be possible with context, which is absent in the domain of isolated SLR. A continuous SLR model might be able to reduce the confusion.

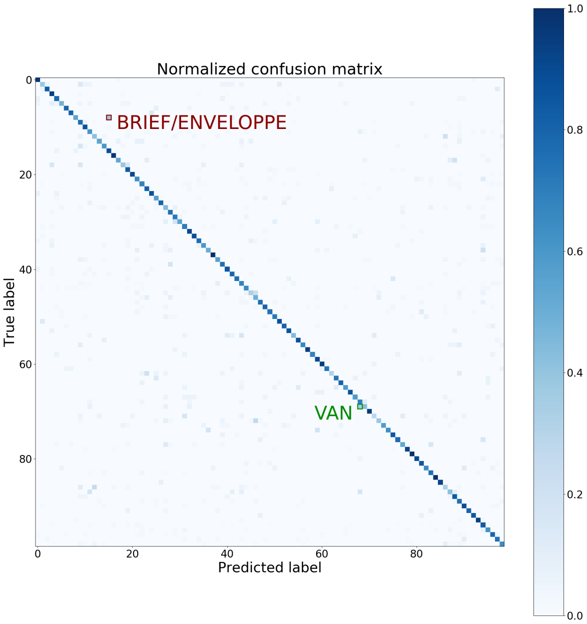

Figure 5: Normalized confusion matrix for the MTN. The two largest confusions have been indicated in red and green.

5.3. Impact of Dataset Size

Pigou et al. (2016) use an older version of the Corpus VGT with only 12599 labeled samples and obtains top-1, top-3 and top-5 accuracy values of 39.3%, 60.3% and 69.9% respectively. The top-10 accuracy is not explicitly reported. In this work, we use an updated version with 18730 labeled samples. In order to compare both works, we investigate the impact of this dataset update. To do this, we measure the absolute increase in top-n accuracy for each of the methods. The average absolute increases in top-1, top-3, top-5 and top-10 accuracy are 9.31%, 8.33%, 7.83% and 6.89% respectively. It is clear from these results that increasing the dataset size has a large impact on the accuracy on unseen data.

We now show that further increases in dataset size will likely lead to further improvements in accuracy. Figure 6 shows a learning curve for the MTN. The validation accuracy curve has not yet converged, which indicates that more data, or more data augmentation, would lead to better performance. In fact, the curve suggests that the vocabulary size will be able to be increased, as isolated SLR will be a solved problem for vocabularies of 100 glosses with double the amount of labeled data. This illustrates the importance of speeding up corpus annotation for SLR.

6. Conclusion

This work presents four network architectures for SLR: three based on transformer networks and one based on LSTMs. Pigou et al. (2016) used end-to-end deep learning to extract features from video data and classify signs based on these features. Initial results in this work

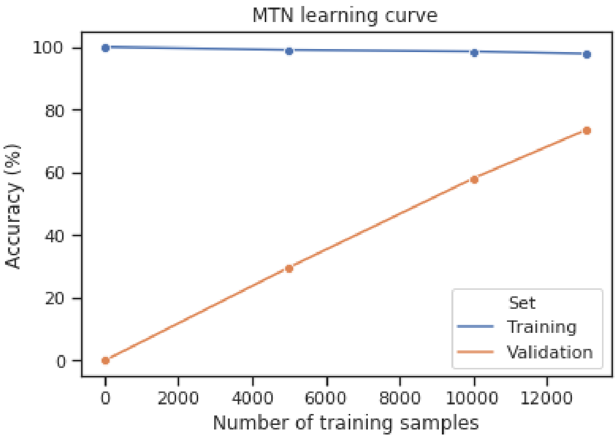

Figure 6: Learning curve for the MTN.

indicate that a network trained using features extracted by the pose estimation technique OpenPose is able to significantly outperform this previous work. However, by using state of the art techniques for computer vision (pre-trained CNNs) and sequence learning (transformers), end-to-end deep learning is able to extract features that are more salient than OpenPose keypoints. Future work on SLR in sign language corpora must focus on extracting salient features from the available data.

By using attention to combine both feature sets (learned features and OpenPose keypoints), the network accuracy can further be improved beyond what is possible when considering either feature set in isolation. Our best method, which we name the "Multimodal Transformer Network", obtains 74.7% accuracy on the unseen test set for a vocabulary of 100 classes.

Finally, we show that increasing the dataset size leads to increased performance. The proposed methods can be used in a suggestion tool for sign language corpus annotation. Such a tool has the potential to speed up the annotation process, which will lead to the availability of larger datasets for SLR.

In future work, we will investigate continuous SLR and focus on creating models that are capable of understanding sign language.

7. Bibliographical References

Ba, J. L., Kiros, J. R., and Hinton, G. E. (2016). Layer normalization. *arXiv preprint arXiv:1607.06450*.

Bauer, B. and Kraiss, K.-F. (2001). Towards an automatic sign language recognition system using subunits. In *International Gesture Workshop*, pages 64–75. Springer.

Bragg, D., Koller, O., Bellard, M., Berke, L., Boudreault, P., Braffort, A., Caselli, N., Huenerfauth, M., Kacorri, H., Verhoef, T., Vogler, C., and Morris, M. R. (2019). Sign language recognition, generation, and translation: An interdisciplinary perspective. In *ASSETS 2019*. ACM, October. Best Paper Award.

Cao, Z., Simon, T., Wei, S.-E., and Sheikh, Y. (2017). Realtime multi-person 2d pose estimation using part affinity fields. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7291–7299.

Chai, X., Wang, H., and Chen, X. (2014). The devisign large vocabulary of chinese sign language database and baseline evaluations. *Technical report VIPL-TR-14-SLR-001. Key Lab of Intelligent Information Processing of Chinese Academy of Sciences (CAS), Institute of Computing Technology, CAS*.

Charles, J., Pfister, T., Everingham, M., and Zisserman, A. (2014). Automatic and efficient human pose estimation for sign language videos. *International Journal of Computer Vision*, 110(1):70–90.

Cihan Camgoz, N., Hadfield, S., Koller, O., Ney, H., and Bowden, R. (2018). Neural sign language translation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7784–7793.

Cooper, H., Ong, E.-J., Pugeault, N., and Bowden, R. (2012). Sign language recognition using sub-units. *Journal of Machine Learning Research*, 13(Jul):2205–2231.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. (2009). Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota, June. Association for Computational Linguistics.

Fang, H.-S., Xie, S., Tai, Y.-W., and Lu, C. (2017). Rmpe: Regional multi-person pose estimation. In *ICCV*.

Glorot, X. and Bengio, Y. (2010). Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pages 249–256.

He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778.

Huang, J., Zhou, W., Zhang, Q., Li, H., and Li, W. (2018). Video-based sign language recognition without temporal segmentation. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

Jing, L., Vahdani, E., Huenerfauth, M., and Tian, Y. (2019). Recognizing american sign language manual signs from rgb-d videos. *arXiv preprint arXiv:1906.02851*.

Jozefowicz, R., Zaremba, W., and Sutskever, I. (2015). An empirical exploration of recurrent network architectures. In *International Conference on Machine Learning*, pages 2342–2350.

Karpathy, A., Toderici, G., Shetty, S., Leung, T., Sukthankar, R., and Fei-Fei, L. (2014). Large-scale video classification with convolutional neural networks. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 1725–1732.

Kingma, D. P. and Ba, J. (2015). Adam: A method for stochastic optimization. In Yoshua Bengio et al., editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Ko, S.-K., Son, J. G., and Jung, H. (2018). Sign language recognition with recurrent neural network using human keypoint detection. In *Proceedings of the 2018 Conference on Research in Adaptive and Convergent Systems*, pages 326–328. ACM.

Koller, O., Bowden, R., and Ney, H. (2016a). Automatic alignment of hamnosys subunits for continuous sign language recognition. *LREC 2016 Proceedings*, pages 121–128.

Koller, O., Zargaran, O., Ney, H., and Bowden, R. (2016b). Deep sign: Hybrid cnn-hmm for continuous sign language recognition. In *Proceedings of the British Machine Vision Conference 2016*.

Koller, O., Zargaran, S., and Ney, H. (2017). Re-sign: Re-aligned end-to-end sequence modelling with deep recurrent cnn-hmms. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4297–4305.

Konstantinidis, D., Dimitropoulos, K., and Daras, P. (2018). A deep learning approach for analyzing video and skeletal features in sign language recognition. In *2018 IEEE International Conference on Imaging Systems and Techniques (IST)*, pages 1–6. IEEE.

Kozlov, A., Andronov, V., and Gritsenko, Y. (2019). Lightweight network architecture for real-time action recognition. *arXiv preprint arXiv:1905.08711*.

Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105.

Mueller, F., Bernard, F., Sotnychenko, O., Mehta, D., Sridhar, S., Casas, D., and Theobalt, C. (2018). Ganerated hands for real-time 3d hand tracking from monocular rgb. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 49–59.

Ong, E.-J. and Bowden, R. (2004). A boosted classifier tree for hand shape detection. In *Sixth IEEE International Conference on Automatic Face and Gesture Recognition, 2004. Proceedings.*, pages 889–894. IEEE.

Paszke, A., Gross, S., Chintala, S., Chanan, G., Yang, E., DeVito, Z., Lin, Z., Desmaison, A., Antiga, L., and Lerer, A. (2017). Automatic differentiation in PyTorch. In *NIPS Autodiff Workshop*.

Pigou, L., Dieleman, S., Kindermans, P.-J., and Schrauwen, B. (2014). Sign language recognition using convolutional neural networks. In *European Conference on Computer Vision*, pages 572–578. Springer.

Pigou, L., Van Herreweghe, M., and Dambre, J. (2016). Sign classification in sign language corpora with deep neural networks. In *International Conference on Language Resources and Evaluation (LREC), Workshop*, pages 175–178.

Pigou, L., Van Herreweghe, M., and Dambre, J. (2017). Gesture and sign language recognition with temporal residual networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3086–3093.

Ronchetti, F., Quiroga, F., Estrebou, C., Lanzarini, L., and Rosete, A. (2016). Lsa64: A dataset of argentinian sign language. In *XX II Congreso Argentino de Ciencias de la Computación (CACIC)*.

Starner, T., Weaver, J., and Pentland, A. (1998). Real-time american sign language recognition using desk and wearable computer based video. *IEEE Transactions on pattern analysis and machine intelligence*, 20(12):1371–1375.

Tran, D., Wang, H., Torresani, L., Ray, J., LeCun, Y., and Paluri, M. (2018). A closer look at spatiotemporal convolutions for action recognition. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 6450–6459.

Vaezi Joze, H. and Koller, O. (2019). Ms-asl: A large-scale data set and benchmark for understanding american sign language. In *The British Machine Vision Conference (BMVC)*, September.

Van Herreweghe, M., Vermeerbergen, M., Demey, E., De Durpel, H., and Verstraete, S. (2015). Het Corpus VGT. Een digitaal open access corpus van videos and annotaties van Vlaamse Gebarentaal, ontwikkeld aan de Universiteit Gent i.s.m. KU Leuven. `www.corpusvgt.be`.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L. u., and Polosukhin, I. (2017). Attention is all you need. In I. Guyon, et al., editors, *Advances in Neural Information Processing Systems 30*, pages 5998–6008. Curran Associates, Inc.

Vogler, C. and Metaxas, D. (1997). Adapting hidden markov models for asl recognition by using three-dimensional computer vision methods. In *1997 IEEE International Conference on Systems, Man, and Cybernetics. Computational Cybernetics and Simulation*, volume 1, pages 156–161. IEEE.

Wang, L., Xiong, Y., Wang, Z., and Qiao, Y. (2015). Towards good practices for very deep two-stream convnets. *arXiv preprint arXiv:1507.02159*.

Ye, Y., Tian, Y., Huenerfauth, M., and Liu, J. (2018). Recognizing american sign language gestures from within continuous videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 2064–2073.