Video ConvNet-aided Sign Language Translation

Seung-Hoon Yi[1], Soh-Hyung Park[1], and Eunji Lee[1]

[1]Seoul National University, jaguar6182,lgb0324,leee4321@snu.ac.kr

Abstract

Sign languages use multiple and channels including gesture by hand, pose and facial expressions in communication. These are hard to take account in computational Sign Language Tranlstion(SLT). To solve this problem, multiple models, encompassing seq2seq and transformers had been widely proposed. In this paper, we tackle the SLT problem with aid of pretrained 3D convolution neural networks(CNNs) without any aid of bridging representations. We used RWTH-PHOENIX 2014 T dataset, one of the most commonly used dataset encompassing sign language videos, aligned transcription, and translations. Currently, the state-of-the art result of SLT task in this dataset demonstrates a BLEU-4 score of 13.41, with an end-to-end fashion training. In this paper, we propose a model with comparable performance of state-of-the art(SOTA) models with a BLEU score of 11.2, but with parameter size less than 1/5.

1. Introduction

Sign Languages are the primary communication medium of the Deaf. Sign Languages are distinct language systems which convey information through hand shape, facial expression, upper body posture, etc. Generally, sign languages are developed independently of spoken languages and have different linguistic rules compared to those of spoken languages. Hence, converting sign language and natural language one to another is an important task to bridge communication gaps with the deaf people. There have been various approaches to interpret sign video sequences into natural language text. This, especially Sign Language Recognition (SLR) or Sign Language Translation (SLT) is a challenging task in the field of computer vision since it involves interpreting several visual information such as body movements, facial expression into linguistic information.

Early works had focused on SLR approaches to interpret sign language into natural language. SLR methods mainly regards this task as a gesture recognition problem with the assumption that there exists one-to-one mapping between sign language and spoken language. Early works in SLR mainly focused on using hand-craft features with statistical modeling. More recently, extracting features from video with deep-learning method have achieved breakthrough in the field of continuous SLR. However, as we can see in Figure 1, when interpreting sign language glosses into spoken language, linguistic and grammar characteristics such as sentence length and word order are significantly different. So, it is challenging to precisely align sign language into spoken language with existing SLR methods.

As such, there have been Sign Language Translation approaches aiming at full translation dealing this task with an aspect of machine translation because one-to-one mapping between sign language and spoken language does not exist. Conceptual video-based methods were introduced in early SLT works. Recently, end-to-end approaches were introduced using attention-based Neural Machine Translation (NMT) models [4].

The biggest obstacle of video based continuous SLT research has been lack of suitable datasets to train models. Recently, Camgoz et al.[4] released the first continuous SLT dataset containing video segments, gloss annotations and spoken language translation, RWTH-PHOENIX-Weather-2014T (PHOENIX14T), which comprises glosses of popular SLR dataset RWTH-PHOENIX-Weather-2014 (PHOENIX14). The authors also approached translation task as a NMT problem, namely Sign2Text approaching the end goal of SLT without going via gloss annotation, Sign2Gloss2Text extracting gloss sequence (Sign2Gloss) and then approaching the task as a text-text problem (Gloss2Text).

More recently, d, Li et al. [15] focused on Sign2Text problem, namely TSPNet. By using features learned from video segments which encodes both spatial and temporal features with semantic hierarchical structure, TSPNet captures temporal information in sign gestures and has increased the BLEU-4 score from 9.58 [4] to 13.41. Camgoz et al. [7] proposed an end-to-end transformer based architecture jointly learning sign language recognition and trans-

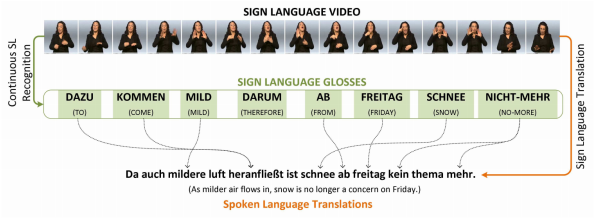

Figure 1. Difference between CSLR and SLT [4]

lation which is the current state-of-art on PHOENIX14T, namely Sign2(Gloss+Text). The authors used gloss annotations to train transformer encoders to learn spatial representation for SLT named Sign Language Recognition Transformers (SLRT) and then autoregressive transformer decoder named Sign Language Translation Transformers (SLTT) exploits this representation learned by SLRT increasing the BLEU-4 score from 9.58 [4] to 21.80[7]. However, as they are already transcription-guided, we aim to achieve comparable sign translation accuracy with current SOTA models, with relatively few parameters. In this point of view, we focused on optimizing translation model based on seq2seq with attention.

2. Related Works

2.1. Sign Language Recognition Models

The researches in SLR have been developed for decades, from sign segmentation to end-to-end sign language translation. They are divided by two main goals: Recognition and translation. The goal of sign language recognition is to detect and locate the signs so that the systems understand the information that sign language videos deliver. The goal of the latter is to translate sign language into natural language.

Initial works in SLR had focused on recognizing isolated sign gestures [17, 9, 19, 23]. Beyond recognizing only isolated signs, continuous SLR (CSLR) has emerged in order to apply to real-life signs. The studies have focused on recognizing sign glosses. However, such SLR using glosses have limitations since glosses, even manually provided by experts, can represent only a few frames in sign videos compared to actual actions involved in sign languages. With such failure of utilizing only glosses, other feature extraction methods have developed in SLR to extract visual information from sign videos. Recently, convolutional neural networks (CNNs) [21, 22] or pose estimation technique [12, 8] have been widely used as feature extractors.

2.2. Sign Language Translation Models

The research in sign language translation (SLT) is challenging because sign languages have their own grammatical and semantic structures. Sign language and natural language cannot be converted to each other as one-to-one mapping. The models which generate spoken language from sign videos in an end-to-end manner are called Sign2Text. Glosses could also be used as an intermediate representation in SLT research; therefore, to reach the end goal of SLT, one may go through two consecutive steps, Sign2Gloss and Gloss2Text. The CSLR models are used for the first step, Sign2Gloss, and the output of the CSLR models are used for text-to-text translation, which is the second step Gloss2Text.

Camgoz et al. [4] achieved the translation process without glosses, using attention-based NMT. Their work was the first end-to-end learning which enabled deriving text from the sign videos. The results showed that, since Sign2Text networks had problem of long term dependencies, Sign2Gloss2Text networks resulted in better performance than Sign2Text networks. Subsequently, Camgoz et al.[7] used similar attention-based encoder-decoder architecture and supplemented it by adding Positional Encoding to Word and Spatial Embeddings. Despite the high performances of [4] and [7], they lose temporal information among the frames. In that context, Yin and Read [24] achieved state-of-the-art performance using Spatial-Temporal Multi-Cue (STMC) Network. They obtained quality tokenization by translating ground truth glosses and the Sign2Gloss2Text task using STMC-Transformer was followed by Gloss2Text experiment. Although the performance of Sigh2Gloss2Text surpassed that of Gloss2Text in this study, gloss annotations seem to play a critical role as intermediate representation in SLT.

On the other hand, Li et al. [15] used video features in Sign2Text network instead of gloss annotations. The video features were extracted with 3D-CNN I3D as video segment representation. This approach improved capturing temporal information of sign videos. In the current study, we apply the encoder-decoder architecture, substituting the spatial embeddings for 3D-CNN features.

3. Embeddings

3.1. Video Embeddings

We consider the SLT task as a machine language translation task, and designed a model which receives visual features of the given videos as inputs, and returns a sequence of sign language, translated to plain text. To fully utilize the advantage of multiple visual clues, it is possible to append a simple CNN under the encoder layer and obtain features. However, it cannot learn temporal clues that are expected to be useful in translation. As we target to learn from spatio-temporal features extracted from the whole video, we decided to use 3D CNN models that can learn both spatial and temporal features and summarize within a small dimension. Visual features were first

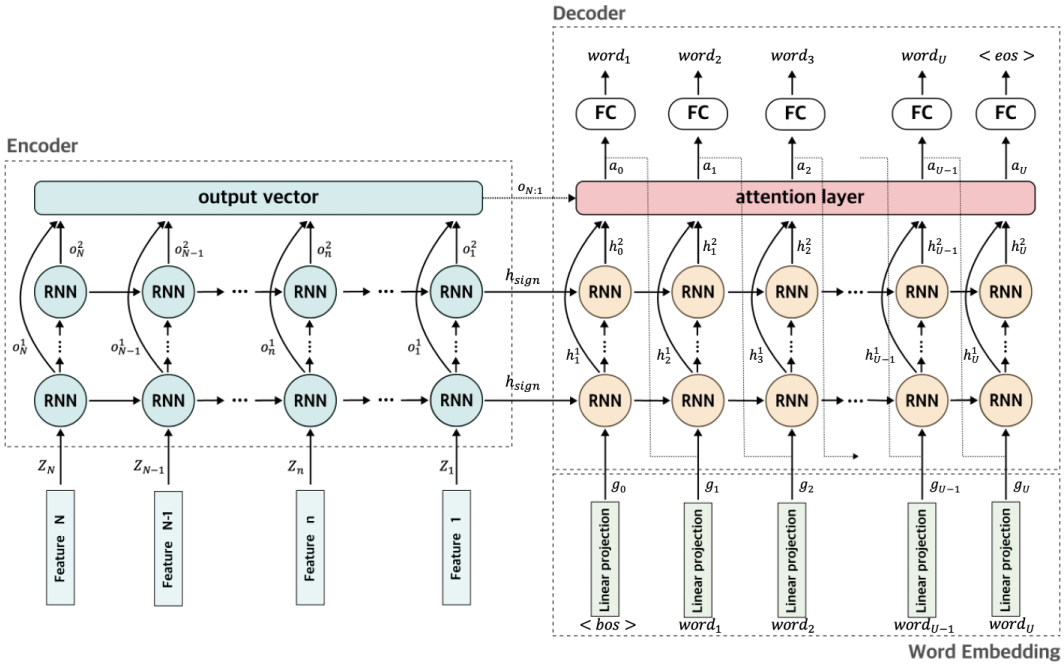

Figure 2. The overall encoder-decoder structure. Numbers of layers of both encoder and decoder are the same. The entire output of the encoder and the entire hidden state joins in the attention process.

extracted from the last few layers of pretrained 3D CNN, with temporal span of 16 frames. In the first step, we tried to use I3D[5] to extract features. But as the original model was only trained from the kinetics dataset, we utilized the open source from TSPNet[15]. Here the authors fine-tuned only the last layer with their WLASL[14]dataset, a word level sign language dataset. The final feature outputs have dimension of 1024, and expected to learn meaningful data from the given frames.

3.2. Word Embeddings

PHOENIX14T dataset supports video frames and video-wise translations. Every word in the translation text corpus was transformed into one-hot vectors and then projected in 512-dimension embedding space by a linear embedding layer. We also tried to exploit word vectors from fasttext[3], a pre-trained language model which incorporates subword tokenizer. However, to cover rare words, the size of the language model have to be excessively large and makes it difficult to restore a complete text from the vector sequence. Thus, we embed word vectors from scratch and trained in an end-to-end fashion to prevent such problems.

3.3. The Dataset

We trained our model using PHOENIX14T dataset[13], a sign language video with translation and aligned glosses(sign annotations) comprising one or multiple sentences. Translation of all videos contains one or more sentences, forming a text corpus with 2887 unique words and gloss corpus with 1066 unique sign words. The size of the dataset are each 7069, 642, 519 for training, test, validation set. Before training, video length over 314 frames were truncated, as this length results a length of 150 after extracting features. Videos which have shorter length than this were zero-padded after the final frame. This was possible since only 24 videos over 7096 in the training set exceeds this threshold. Therefore the loss from the error by omitting frames could be much smaller than the risk of the vanishing gradient when we set the length of our model as the maximum length of the training videos.

4. Implementation details and Results

Framework and Architecture : We used Tensorflow [1] to build our network. To test whether Seq2Seq models are able to learn from video features, we started from a similar model as Camgoz et al. [4] did. 2 to 4 layers of RNN, either LSTM [10] or GRU [6] are stacked in both the encoder and the decoder, and the encoder was tested in a bidirectional or in a unidirectional structure as shown in Figure 1. To obtain output tokens using the overall context, we applied bahdanau attention [2] between the encoder and the decoder. After passing through a linear layer which returns the softmax score of our given words in the corpus, beam search was applied to obtain the full text of the sign language video as well as to maximize conditional proba-

bility of the entire text.

Learning Rate and Optimizer : All training processes used piecewise constant learning rates that decays halfway with predefined boundaries with the initial learning rate of 5e-3 to 5e-5. Adam optimizer [11] was used for the mini-batch gradient descent [16], using categorical cross entropy as a loss metric.

Training Protocol : We evaluate the model changing batch size from 8 to 64, and applied Xavier initialization in every layer to achieve fast training. Every layer has a dropout rate of 0.2 to alleviate overfitting of the model. After training 12 epochs and when the loss becomes relatively flat, early stopping was applied to retrieve the trained model before overfitting occurs.

Evaluation : To evaluate the quality of translation, we used BLEU [20] score metric for every model with the predicted translations of the test dataset.

4.1. The Encoder-Decoder Layer

We optimized the encoder-decoder structure in three ways. By the direction of state propagation(i.e. whether the model is unidirectional or bidirectional) in the encoder stage, numbers of the layer, and the type of the RNN cell(whether the used cell is LSTM or GRU). We observed that unidirectional structure with shallow layers are more suited for encoding the given spatio-temporal feature. This maybe due to the excessively large model complexity of former deep layer structures, as the input features contain meaningful information on itself. As we expected, the most prominent performance was found from the two layer unidirectional GRU structure.

Hereafter, Video features and word embeddings are notated as z_i, g_i each, and the outputs of the encoders are notated as o_i while superscripts represents from what layer is this output from, and subscripts represents sequence numbers. The encoder and decoder were both set up to have every hidden size of 512, and have the same layer numbers, while the sequence lengths are set to be 150 and 50 each. This is due to the high variation of sequence length in our dataset, and to alleviate gradient vanishing. Detailed explanation of comparison results will be provided in the following sections.

4.2. Attention mechanism

Attention mechanism was applied with the encoder outputs and decoder hidden states in every experiment. The attention layer performs bahdanau attention between every encoder outputs $o_{i\in[1:N]}^{j\in[1:2]}$ and specific decoder hidden state $h_t^{j\in[1:2]}$. First, the outputs of the encoder stage from each layer are concatenated to a vector notated as $O_{1:N}$, and form a weighted sum of the source sequence, which will provide us a vector for the representation of the encoder. This is notated as $\alpha_{(t)}$, which will be constructed as

$$\alpha_t = \sum_{i=1}^{N} \gamma_{i(t)} O_i \tag{1}$$

where γ_i is defined as the attention score expressed as,

$$\gamma_t = \frac{exp(score([h_t^1; h_t^2], O_i)}{\sum_{i=1}^{N} exp(score([h_t^1; h_t^2], O_i)} \tag{2}$$

and the score function is defined using the expression in [2].

$$score(A, B) = V^T tanh(W[A; B]) \tag{3}$$

$$a_t = tanh(W_c[\alpha_t; (h_t^1; h_t^2)]) \tag{4}$$

where W and V are parameters to be learned. Finally, to obtain the context vector or the attention vector a in equation (4), we again concatenate the hidden states in each layer of a certain timestep with the encoder representation, and project it into a smaller dimension. Here, the output dimension is equal to the dimension of O_i.

After the attention layer, we pass the attention vector a_t to a fully connected layer with softmax activation to get the probability map of unique vocabularies in our dataset. The index with the maximum probability will be the number of output vocabulary token at timestep t. We basically apply greedy search to predict the whole sentence, using only the token with maximum probability. However, greedy search does not always guarantee the maximum conditional probability of our output sentence. To maximize the prediction score, beam search was also used, beam search with beam width of k was used, utilizing tokens with top-k probability for every timestep.

4.3. Evaluation Metric

As mentioned above, we used BLEU score [20] as the metric of translation performance. BLEU is widely adopted in the domain of machine translation between one or multiple languages, as it takes into account the order of retrieved sequence as well as the numbers of the correct word tokens using N-grams. We report BLEU-1,2,3,4 scores of the best performing model to compare our models directly with previous works.

4.4. Quantitative Results

4.4.1 Unidirectional vs Bidirectional

We evaluate categorical cross entropy loss and BLEU-4 score of the generated translation using the test dataset to compare performance of unidirectional and bidirectional layers using LSTM cells. As shown in Table 1, unidirectional networks outperforms in both loss and BLEU score metrics. This maybe attributes from the nature of 3D convnets, which already encompasses spatial and temporal features, such that bidirectional structure halts the encoding of

the temporal cues from the features. Hence, the effects of depth or kind of the cell are evaluated using only the unidirectional structure. From here, we set the batch size and the initial learning rate as 32 and 5e-3, dropout rate as 0.2. Every loss and BLEU scores are from the test dataset.

4.4.2 LSTM vs GRU, and Layer Depth.

Table 2 lists the overall loss and BLEU-4 score from the test dataset with our trained models. As shown in our results, GRU-based Neural Nets(NNs) greatly outperforms the LSTM-based NNs, and shallower layers were more effective in translating video features. Therefore, we can conclude that the reduced number of parameters of GRU cell reduced overfitting and make the training more effective. Using only one layer could be another choice, but we obtained no meaningful results with a single layer seq2seq structure, with a drastic BLEU-4 score of under 4. This implies that the features itself requires substantial model complexity, which we could question about why deeper models over 2 layers could not give us a notable performance. Therefore, investigating about the gradient vanishing problems of deeper models seems to be crucial to understanding these phenomena.

(Direction, Number of layers)	Loss	BLEU-4 score
(Uni, 4)	0.8751	7.3820
(Bi, 4)	0.9214	5.9701
(Uni, 3)	**0.8310**	**8.0963**
(Bi, 3)	0.8998	7.0312

Table 1. Loss and BLEU-4 score with the propagation direction of the hidden state. LSTM cells were used for this experiment.

(Cell type, Number of layers)	Loss	BLEU-4 score
(LSTM, 4)	0.8751	7.3820
(LSTM, 3)	0.8310	8.0963
(LSTM, 2)	0.8081	8.8011
(GRU, 4)	0.8705	8.0356
(GRU, 3)	0.8322	8.9327
(GRU, 2)	**0.7971**	**10.261**

Table 2. Loss and BLEU-4 score with different RNN cells and layer depths. Using 2 unidirectional GRU structure outperforms Sign2Text translation of [4].

4.4.3 Effects of Batch Size

Huge batch sizes consumes less time in training, also generally good in finding the local minimum. However, as we are handling large but noisy datasets, updating gradient with large batches can increase risks to fall in through the local minimum. Therefore testing the model with various batch size is important in optimization, and as shown in Table 4, smaller batch sizes really do outperform models that trained with larger batch sizes. This is also consistent with Masters and Luschi(2018)[18] who quantitatively examined about the effects of smaller batch size. In this experiment, we used learning rate of 5e-5 to slow the update as to prevent the model to be overfitted in some easy clues in the early training phase. As the final outcome, 2-layer GRU models trained with batch size of 8 demonstrates the best performance, 2.15 lower BLEU-4 score than TSPNet[15], but 1.68 higher than the former seq2seq based Sign2Text model[4]. The comparison between examples from ground truth translation and predicted translation from the best performing model are listed in Table 3.

GT:	und nun die wettervorhersage für morgen dienstag den neunzehnten april (and now the weather forecast for tomorrow, Tuesday the nineteenth of april)
Ours:	und nun die wettervorhersage für morgen dienstag den achten mai (and now the weather forecast for tomorrow, Tuesday, the eighth of may)
GT:	am sonntag im nordwesten eine mischung aus sonne und wolken mit einigen zum teil gewittrigen schauern (on sunday in the northwest a mixture of sun and clouds with some thunderstorm showers)
Ours:	am sonntag im norden und westen einige schauer und kurze gewitter (on sunday in the north and west some showers and short thunderstorms)
GT:	im westen ist es freundlich (in the west it is friendly)
Ours:	sonst ist es recht freundlich (otherwise it is quite friendly)
GT:	am sonntag ziehen von nordwesten wieder schauer und gewitter heran (on sunday, showers and thunderstorms are approaching again from the northwest)
Ours:	am sonntag regnet es gebietsweise im flachland regnet es (on sunday it rains in areas in the lowlands it rains)
GT:	vom südwesten bis in die mitte bleibt es meist trocken (from the southwest to the center it usually stays dry)
Ours:	im übrigen land bleibt es weitgehend trocken (the rest of the country remains largely dry)

Table 3. Generated translations from our best performing models. (GT: Ground Truth, Ours: Predicted Sequence)

4.5. Comparison with our target model

CSLT can be achieved with various training methods, such as employing intermediate representations(gloss), or extracting human keypoints from the source video. Unfortunately, due to the low resolution, extracting meaningful keypoints were not an option. Hence we focus on direct translation from the video features, named as Sign2Text by

Batch size	BLEU-1	BLEU-2	BLEU-3	BLEU-4
8	**30.57**	**19.92**	**14.41**	**11.26**
16	27.86	18.80	14.03	10.77
32	26.44	18.72	13.50	10.26
64	26.28	17.92	13.14	10.09

Table 4. BLEU scores of the predicted sentence of the training dataset, with different batch size in training. The structure is the same with the one used in Table 2.

Model	Architecture	BLEU-1	BLEU-2	BLEU-3	BLEU-4	Num. of parameters
TSPNet[15]	Transformer	**36.10**	**23.12**	**16.88**	**13.41**	76.2M
S2T[4]	Seq2Seq	32.24	19.03	12.83	9.58	26.4M
Ours	Seq2Seq	30.57	19.92	14.41	11.26	**14.5M**

Table 5. BLEU scores, structures and numbers of parameter of compared models. Interestingly, our model generated sentences that are less accurate in word level, but more likely to form a correct N-grams with N greater than 1.

Camgoz et al.(2018)[4]. Nevertheless, we succeeded in performing beyond the previous model[4], with less numbers of parameters. Moreover, we could reach similar performance with feature based transformer model[15] with only 14.5 million parameters, nearly 1/5 of the cited paper. The results of each baseline models of feature-driven translation, their architectures, BLEU scores, and numbers of parameters are listed in Table 5. Ours even show relatively small variations in BLEU scores, which is thought to be our model had catch the whole context of the given video, leading relatively large error rate in word level(BLEU-1), but a more accurate predictions in 2 to 4-gram sentences.

4.6. Further Improvements

We constructed a transformer model which inputs are video embeddings from pretrained I3D model with 3 different span lengths, and outputs are translated text. Due to the pipeline error, we could not make any improvements.

5. Conclusion and Future works

We proposed a model which has relatively small numbers of parameters but comparable with leading feature-based translation models. As the inputs are 'video words' from a pretrained 3D CNN model, encoding and decoding such well-refined summary needs relatively shallow layer depth. This is consistant with the results using GRU were better than when using LSTM as our RNN cell. However, using only one layers could not get a notable performance, as the model could not learn complex representations of the given dataset, or due to the insufficient parameters of the attention layer. Batch size was also a crucial factor towards a more effective training. Results from it reveals us the importance of smaller batch size, contrary to numerous studies that had been conducted in the field of computer vision.

There also exists problems about achieving fully video-driven end-to-end SLT, since we still need pretrained models before we train our translation model. End-to-end training is also a problem. To handle this problem, most of the current the-state-of-the-art approaches in the field of SLT exploits Transformer based architecture which can resolve limitations of Seq2Seq models such as poor parallelization and gradient vanishing. Since we achieved comparable accuracy by optimizing Seq2Seq, we expect using well-optimized Transformer models could lead to further improvement.

Including gloss level annotation as a 'bridge' representations in the training process(Sign2Gloss2Text) demonstrated better performance compared to Sign2Text approach[4, 7]. Unfortunately, many approaches of Sign2Gloss2Text are difficult to train a end-to-end manner as required parameters are way larger than Sign2Text models. One possible extensions we can make to achieve state-of-the-art results in Sign2Text is to exploit custom 3D CNNs for the feature extraction. Applying this is expected to reduce the dependency of glosses as they could catch complex features both in the spatial and temporal domain at the same time. This is also consistent with our results that 3D CNN based feature-driven models outperform former 2D-CNN linked models. Also, since there is no guarantee for the maximum conditional probability obtained by greedy search in the decoder. We expect that better results can be obtained by applying beam search and optimizing beam width as to find the most probable translation of the given video. Linking video CNNs, Transformers, and HMMs will be our next step of this work.

References

[1] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, et al.

Tensorflow: Large-scale machine learning on heterogeneous distributed systems. *arXiv preprint arXiv:1603.04467*, 2016.

[2] D. Bahdanau, K. Cho, and Y. Bengio. Neural Machine Translation by Jointly Learning to Align and Translate. In *International Conference on Learning Representations (ICLR)*, 2015.

[3] P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov. Enriching Word Vectors with Subword Information. *Transactions of the Association for Computational Linguistics*, 5:135–146, 2017.

[4] N. C. Camgoz, S. Hadfield, O. Koller, H. Ney, and R. Bowden. Neural Sign Language Translation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[5] J. Carreira and A. Zisserman. Quo Vadis, Action Recognition? A New Model and the Kinetics Dataset. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4724–4733, 2017.

[6] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*, 2014.

[7] N. Cihan Camgöz, O. Koller, S. Hadfield, and R. Bowden. Sign Language Transformers: Joint End-to-End Sign Language Recognition and Translation. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10020–10030, 2020.

[8] M. De Coster, M. Van Herreweghe, and J. Dambre. Sign Language Recognition with Transformer Networks. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 6018–6024, 2020.

[9] K. Grobel and M. Assan. Isolated Sign Language Recognition using Hidden Markov Models. In *1997 IEEE International Conference on Systems, Man, and Cybernetics. Computational Cybernetics and Simulation*, volume 1, pages 162–167, 1997.

[10] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[11] D. P. Kingma and J. Ba. Adam: A Method for Stochastic Optimization. In *arXiv: 1412.6980*, 2017.

[12] S.-K. Ko, J. G. Son, and H. Jung. Sign Language Recognition with Recurrent Neural Network using Human Keypoint Detection. In *Proceedings of the 2018 Conference on Research in Adaptive and Convergent Systems*, page 326–328, 2018.

[13] O. Koller, J. /Forster, and H. Ney. Continuous sign language recognition: Towards large vocabulary statistical recognition systems handling multiple signers. *Computer Vision and Image Understanding*, 141:108–125, Dec. 2015.

[14] 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 *The IEEE Winter Conference on Applications of Computer Vision*, pages 1459–1469, 2020.

[15] D. Li, C. Xu, X. Yu, K. Zhang, B. Swift, H. Suominen, and H. Li. TSPNet: Hierarchical Feature Learning via Temporal Semantic Pyramid for Sign Language Translation. In *Advances in Neural Information Processing Systems*, volume 33, 2020.

[16] M. Li, T. Zhang, Y. Chen, and A. J. Smola. Efficient mini-batch training for stochastic optimization. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, page 661–670, 2014.

[17] R.-H. Liang and M. Ouhyoung. A Sign Language Recognition System using Hidden Markov Model and Context Sensitive Search. page 59–66, 1996.

[18] D. Masters and C. Luschi. Revisiting small batch training for deep neural networks, 2018.

[19] K. Murakami and H. Taguchi. Gesture Recognition using Recurrent Neural Networks. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '91, page 237–242, 1991.

[20] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu. Bleu: a Method for Automatic Evaluation of Machine Translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA, 2002.

[21] L. Pigou, S. Dieleman, P.-J. Kindermans, and B. Schrauwen. Sign Language Recognition using Convolutional Neural Networks. In L. Agapito, M. M. Bronstein, and C. Rother, editors, *Computer Vision - ECCV 2014 Workshops*, pages 572–578, 2015.

[22] J. Pu, W. Zhou, and H. Li. Iterative Alignment Network for Continuous Sign Language Recognition. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4160–4169, 2019.

[23] F. Yin, X. Chai, and X. Chen. Iterative Reference Driven Metric Learning for Signer Independent Isolated Sign Language Recognition. volume 9911, pages 434–450, 2016.

[24] K. Yin and J. Read. Better Sign Language Translation with STMC-Transformer. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 5975–5989, 2020.