Skeleton-based Online Sign Language Recognition using Monotonic Attention

Natsuki Takayama[1][a], Gibran Benitez-Garcia[1][b] and Hiroki Takahashi[1,2]

[1] *Graduate School of Informatics and Engineering, the University of Electro-Communications, Japan*

[2] *Artificial Intelligence Exploration Research Center, the University of Electro-Communications, Japan*

Keywords: Monotonic Attention, Neural Networks, Skeleton-ased Sign Language Recognition.

Abstract: Sequence-to-sequence models have been successfully applied to improve continuous sign language word recognition in recent years. Although various methods for continuous sign language word recognition have been proposed, these methods assume offline recognition and lack further investigation in online and streaming situations. In this study, skeleton-based continuous sign language word recognition for online situations was investigated. A combination of spatial-temporal graph convolutional networks and recurrent neural networks with soft attention was employed as the base model. Further, three types of monotonic attention techniques were applied to extend the base model for online recognition. The monotonic attention included hard monotonic attention, monotonic chunkwise attention, and monotonic infinite lookback attention. The performance of the proposed models was evaluated in offline and online recognition settings. A conventional Japanese sign language video dataset, including 275 types of isolated word videos and 113 types of sentence videos, was utilized to evaluate the proposed models. The results showed that the effectiveness of monotonic attention to online continuous sign language word recognition.

1 INTRODUCTION

Sign language is a natural language commonly represented by several visual cues, such as hand motions and shapes, and non-manual signals that include posture, facial expression, gaze, and mouthing. From these characteristics, vision-based sign language recognition, which can estimate words from sign language videos, is an important subject of research to conduct machine translation of sign language to text. In recent years, continuous sign language word recognition (Koller et al., 2017; Huang et al., 2018; Pu et al., 2019; Cui et al., 2019; Zhou et al., 2020; Papastratis et al., 2020; Koller et al., 2020; Takayama et al., 2021b) and sign language translation (Camgoz et al., 2018; Camgoz et al., 2020; Guo et al., 2020; Zhou et al., 2021) based on deep neural networks (DNNs) have been proposed. In particular, sequence-to-sequence (Seq2Seq) models, which can directly learn projections between the input videos and sentences, have garnered attention owing to their high recognition performance.

Although various methods have been proposed, there is room for the improvement of sign language recognition in practical situations. For example, recognition systems should equip sign language detection (Moryossef et al., 2020) to handle streaming videos in an online situation. Moreover, several applications require online recognition, such as real-time communication systems. Figure 1 illustrates the difference between online and offline recognition. On the one hand, online recognition estimates words as quickly as the signer utters them. On the other hand, offline recognition waits for the entire sequence before starting the estimation. Users cannot know content until their partner's speaking is finished if offline recognition is applied in a system. This makes a system difficult to use for real-time communication. Conventional sign language recognition implicitly assumes an offline situation, and the investigation of online recognition techniques has not been well researched.

Owing to the aforementioned considerations, the skeleton-based online sign language recognition was investigated in this study. Skeleton-based sign language recognition has received growing interest (Kumar et al., 2019; De Coster et al., 2020; Li et al., 2020;

[a] https://orcid.org/0000-0002-4455-5820

[b] https://orcid.org/0000-0003-4945-8314

Takayama, N., Benitez-Garcia, G. and Takahashi, H.
Skeleton-based Online Sign Language Recognition using Monotonic Attention.
DOI: 10.5220/0010899400003124
In *Proceedings of the 17th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISIGRAPP 2022) - Volume 5: VISAPP*, pages 601-608
ISBN: 978-989-758-555-5; ISSN: 2184-4321

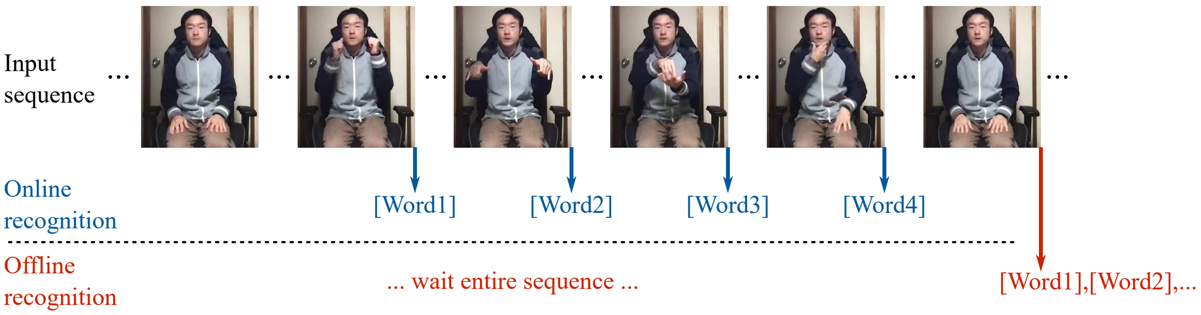


Figure 1: Difference between online and offline recognition. The arrows show yielding words by recognition models.

Jiang et al., 2021; Takayama et al., 2021b) in recent years because of its practical advantages. For example, the tracking points of the human skeleton are robust against scene variation, and the lightweight data reduces the learning and estimation time. Moreover, several state-of-the-art skeleton tracking methods can be applied to edge devices to achieve edge computing. These characteristics are essential for a production system using online recognition.

A combination of spatial-temporal graph convolutional network (STGCN) (Yan et al., 2018) and recurrent neural network with attention (RNNA) (Bahdanau et al., 2015) was employed as a base model. Subsequently, the base model was extended for online recognition by applying monotonic attention techniques. In this study, three types of monotonic attention: hard monotonic attention (HMA) (Raffel et al., 2017), monotonic chunkwise attention (MoChA) (Chiu and Raffel, 2018), and monotonic infinite lookback attention (MILK) (Arivazhagan et al., 2019), were examined.

The proposed models were evaluated using a conventional Japanese sign language (JSL) video dataset (Takayama et al., 2021b) that included 275 types of isolated sign language word videos and 113 types of continuous sign language word videos. Ultimately, the effectiveness of the proposed models for online sign language recognition was reported.

In the following sections, the term “individual word” and “continuous word” indicate “isolated sign language word” and “continuous sign language word”, respectively, to avoid redundant representations.

2 RELATED WORK

Standard sign language recognition is a combination of framewise feature extraction and temporal recognition. Previous methods have employed a combination of handcrafted features and statistical temporal recognition, such as the hidden Markov model (HMM) (Cooper et al., 2011; Forster et al., 2013). Currently, DNN has replaced the technical elements to improve recognition performance.

Koller et al. (Koller et al., 2017) proposed a continuous word recognition using a hybrid model based on the convolutional neural network (CNN), long short term memory (LSTM), and HMM. The superiority of data-driven feature extraction performed by the CNN-LSTM was demonstrated. This method has since been extended to multi-stream HMM based on hand shapes, mouthing, and entire bodies (Koller et al., 2020).

As end-to-end DNN approaches, CNN-LSTM (Cui et al., 2019), three-dimensional-CNN-LSTM with hierarchical attention (Huang et al., 2018), and three-dimensional-ResNet-LSTM (Pu et al., 2019) have been proposed for continuous word recognition. These methods incorporate state-of-the-art techniques derived from action recognition and natural language processing. Moreover, these methods employ stepwise training and multitask learning to improve the model generalization.

In addition to the continuous word recognition, a few research groups have attempted end-to-end sign language translation (Camgoz et al., 2018; Camgoz et al., 2020; Guo et al., 2020; Zhou et al., 2021) in recent years.

All the aforementioned methods implicitly premise offline situations in their recognition and translation, and their performance in online situations has not been evaluated.

3 CONTINUOUS SIGN LANGUAGE WORD RECOGNITION

Continuous word recognition can be modeled as a Seq2Seq learning problem. Let $\boldsymbol{X} = \{\boldsymbol{x}_1,\ldots,\boldsymbol{x}_t,\ldots,\boldsymbol{x}_T\}, \boldsymbol{x}_t \in \mathcal{R}^{100}$ and $\boldsymbol{Y} = \{y_1,\ldots,y_s,\ldots,y_S\}, y_s \in \{< start >, < end >, <$

$pad >, \boldsymbol{L}\}$ be an input feature sequence and a word sequence, respectively. T and S indicate the lengths of an input feature sequence and a word sequence, respectively. $\boldsymbol{x}_t$ is a set of tracking coordinates in the human skeleton extracted from the t_{th} video frame. OpenPose (Cao et al., 2021) was used as a human skeleton tracker in this study. The horizontal and vertical coordinates of the 50 points, including the nose, neck, arms, and hands, were employed in this study. Note that each coordinate was normalized using the average coordinates of the nose and the average length between the both shoulders. $\boldsymbol{L}$ is the set of words to be recognized. <start>and <end>are keywords that represent the start and end of the word sequence, respectively. <pad>is a padding keyword to ensure that the lengths of the word sequences are the same. The Seq2Seq models learn optimized projection $\boldsymbol{X} \rightarrow \boldsymbol{Y}$ through training.

3.1 Architecture Overview

The overview of the proposed model is shown in Figure 2. Figure 2 (a) and (b) show the framewise feature extraction module and overall model architecture, respectively. The proposed model comprises an encoder and a decoder. The encoder converts the input feature sequence into an abstracted hidden vector sequence. While the encoder converts the entire sequence of the input feature $\boldsymbol{X}_{1:T}$ into the hidden vector sequence $\boldsymbol{H}^e_{1:T}$ in an offline situation, it converts a chunk of the input feature sequence $\boldsymbol{X}_{t:t+u-1}$ into a chunk of the hidden vector sequence $\boldsymbol{H}^e_{t:t+u-1}$ in an online situation, where u is the size of the chunk. The decoder utilizes the encoder's hidden vector and input word sequence to predict the output word autoregressively.

The encoder converts the input feature $\boldsymbol{x}_t$ into a hidden vector $\boldsymbol{h}^e_t$ as

$$\boldsymbol{h}^e_t = \text{Encoder}(\boldsymbol{x}_t, \boldsymbol{h}^e_{t-1}), \qquad (1)$$

where the initial hidden vector $\boldsymbol{h}^e_0 = \boldsymbol{0}$. In the feature-extraction module, the input feature $\boldsymbol{x}_t$ is first normalized by one-dimensional masked batch normalization (MBN) (Takayama et al., 2021a). Next, the four cascaded STGCN layers apply graph convolution to the intermediate feature. STGCN layer applies spatial and temporal graph convolution according to a spatiotemporal graph. The proposed model employs the same graph definition as the conventional method (Takayama et al., 2021b). While the STGCN layer yields a feature map $\mathcal{R}^{C\times T\times J}$, the subsequent RNN layer requires a feature vector sequence $\mathcal{R}^{C\times T}$, where C and J are the dimensions of channels and joints. Hence, a linear transformation layer is applied to transform the feature map into a feature vector sequence. Note that TanhExp (Liu and Di, 2020) is employed as an activation function instead of the ReLU function in the proposed model. This modification slightly improved the recognition performance in this study. Finally, the feature vector sequence is converted into a hidden vector sequence using the RNN layer. In this study, the gated recurrent unit (GRU) (Cho et al., 2014) is used for the RNN layer.

The decoder estimates the word sequence autoregressively as

$$\hat{y}_s = \text{Decoder}(y_s, \boldsymbol{h}^d_{s-1}, \boldsymbol{c}_s), \qquad (2)$$

where $\boldsymbol{h}^d_s, \boldsymbol{h}^d_0 = \boldsymbol{0}$ and $\boldsymbol{c}_s$ are a hidden vector of the decoder's RNN layer and a context vector output by the attention layer, respectively. In the decoder, a one-hot vector representation of the word index is first converted into a four-dimensional feature vector through the word-embedding layer. In the training phase, the correct word y_s is entered into the word-embedding layer. In the test phase, the past estimated word $\hat{y}_{s-1}$ is entered into the layer. Simultaneously, the attention layer yields the context vector $\boldsymbol{c}_s$ using the encoder's hidden vector sequence $\boldsymbol{H}^e$ and the decoder's past hidden vector $\boldsymbol{h}^d_{s-1}$. The concatenated vector, including the context vector and output of the word-embedding layer, is entered into the RNN layer of the decoder. Finally, the hidden vector is transformed into the responses of each word using a linear transformation layer.

3.2 Monotonic Attention

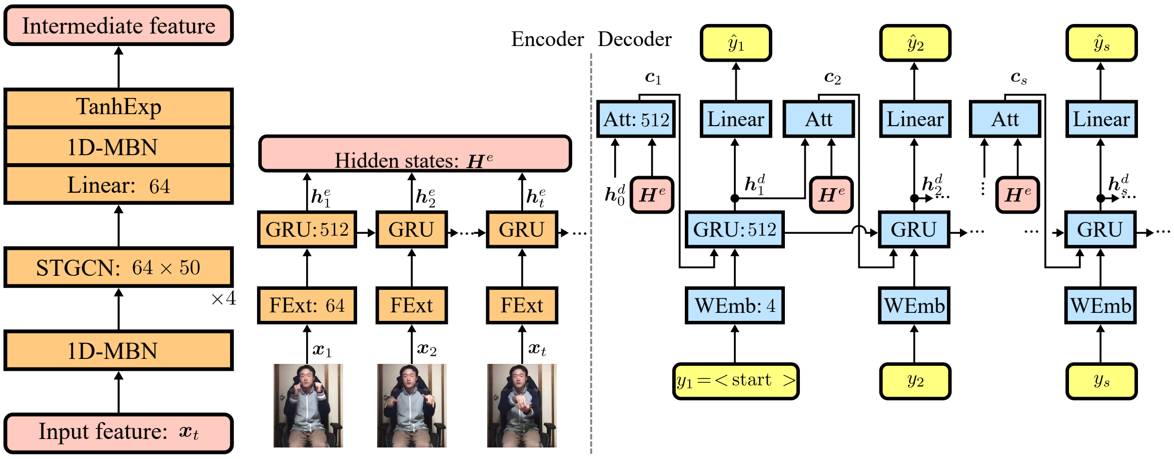
This section briefly introduces the computation of attention layers investigated in this study. The inference processes of these layers are focused on in this paper. For the computation in the training phase, please refer to the original papers (Bahdanau et al., 2015; Raffel et al., 2017; Chiu and Raffel, 2018; Arivazhagan et al., 2019).

3.2.1 Standard Soft Attention

The standard soft attention layer (Bahdanau et al., 2015) computes the context vector as follows: First, an energy value $e_{s,t}$ is calculated for each frame t at each output time step s as follows:

$$\begin{aligned} e_{s,t} &= \text{Energy}(\boldsymbol{h}^d_{s-1}, \boldsymbol{h}^e_t), \\ &= \boldsymbol{v}^{\mathrm{T}}\sigma(\boldsymbol{W}^d\boldsymbol{h}^d_{s-1} + \boldsymbol{W}^e\boldsymbol{h}^e_t + \boldsymbol{b}). \qquad (3) \end{aligned}$$

$\sigma(\cdot)$ denotes an activation function. The hyperbolic tangent was applied to compute the soft attention in this study. $\boldsymbol{W}^d \in \mathcal{R}^{d\times \dim(\boldsymbol{h}^d)}$, $\boldsymbol{W}^e \in \mathcal{R}^{d\times \dim(\boldsymbol{h}^e)}$,



(a) Feature extraction.

(b) Overall architecture.

Figure 2: Process overview. “FExt”, “WEmb”, and “Att” indicate feature-extraction, word-embedding, and attention layers, respectively. The number of layers indicates the dimensions of the output from the layer. The vocabulary determines the dimension of the decoder’s linear transformation layer.

$\boldsymbol{b} \in \mathcal{R}^d$, and $\boldsymbol{v} \in \mathcal{R}^d$ are the learnable parameters of linear layers. d and $\dim(\cdot)$ denote the dimensions of the intermediate feature and hidden vectors, respectively. Furthermore, the energy value $e_{s,t}$ is normalized using the softmax function.

$$\alpha_{s,t} = \frac{\exp(e_{s,t})}{\sum_{i=1}^{T}\exp(e_{s,i})} \tag{4}$$

Finally, the context vector $\boldsymbol{c}_s$ is computed as a weighted average of an attention weight $\alpha_{s,t}$ and a hidden vector $\boldsymbol{h}^e_t$.

$$\boldsymbol{c}_s = \sum_{t=1}^{T} \alpha_{s,t}\boldsymbol{h}^e_t \tag{5}$$

As described in Equation (5), the computation of the context vector $\boldsymbol{c}_s$ requires the encoder’s hidden vector sequence of the entire frame $\boldsymbol{H}^e$. This hinders the application of the model to online recognition.

3.2.2 Hard Monotonic Attention

To address the aforementioned issue of the soft attention, Raffel et al. proposed the HMA (Raffel et al., 2017). Similar to the soft attention, the energy value $e_{s,t}$ is first calculated as

$$\begin{aligned} e_{s,t} &= \mathrm{MonotonicEnergy}(\boldsymbol{h}^d_{s-1}, \boldsymbol{h}^e_t), \\ &= g\frac{\boldsymbol{v}^{\mathcal{T}}}{||\boldsymbol{v}||}\sigma(\boldsymbol{W}^d\boldsymbol{h}^d_{s-1} + \boldsymbol{W}^e\boldsymbol{h}^e_t + \boldsymbol{b}) + \gamma, \end{aligned} \tag{6}$$

where g and γ are the learnable parameters that stabilize the computation of the energy value $e_{s,t}$. In this study, the ReLU function was used as the activation function for the HMA computation. Next, the energy value $e_{s,t}$ is transformed to a probability $p_{s,t}$ using the sigmoid function.

$$p_{s,t} = \mathrm{Sigmoid}(e_{s,t}). \tag{7}$$

Finally, the probability $p_{s,t}$ is transformed into a binary hard attention weight $z_{s,t}$ by thresholding. 0.04 was employed as the threshold in this study.

$$z_{s,t} = \mathbb{1}_{p>0.04}(p_{s,t}). \tag{8}$$

At this time, the hidden vector of the frame where $z_{s,t} = 1$ is assigned as the context vector.

$$\begin{aligned} \boldsymbol{c}_s &= \boldsymbol{h}^e_{t_s}, \\ t_s &= \min\{t; t_{s-1} \leq t \leq T, z_{s,t} = 1.\} \end{aligned} \tag{9}$$

The inference process of the HMA does not require the entire frame of the encoder’s hidden vector sequence. This characteristic is familiar with online recognition.

3.2.3 Soft Attention over Chunks

Although the HMA is available in online situations, its context vector relies on a single hidden vector of the encoder. Therefore, the recognition performance tends to be degraded from the soft attention. MoChA (Chiu and Raffel, 2018) and MILK (Arivazhagan et al., 2019) can remedy this issue by applying the soft attention over a limited range of frames.

In the inference process, MoChA and MILK calculate the softmax attention from the past frames of the frame t_s sampled using the HMA. The inference processes of MoChA and MILK are as follows: After the HMA samples the frame t_s, MoChA and MILK compute the energy value as

$$e_{s,i} = \mathrm{ChunkEnergy}(\boldsymbol{h}^d_{s-1}, \boldsymbol{h}^e_i); i \in \{t_s - w + 1, \ldots, t_s\}. \tag{10}$$

$\mathrm{ChunkEnergy}(\cdot)$ is the same as $\mathrm{Energy}(\cdot)$, except that the computation is performed in a limited range of frames. The ReLU function was employed as the activation function in the computation of MoChA and MILK in this study. w is the window size. MoChA applies a fixed window size, and $w = 4$ was employed in this study. MILK uses a variable-sized window $w = t_s$. The context vector is computed in the same manner as the standard soft attention, except that the computation is performed over frames in the window.

$$\begin{aligned} \boldsymbol{c}_s &= \sum_{i=t_s-w+1}^{t_s} \beta_{s,i} \boldsymbol{h}^e_i, \\ \beta_{s,i} &= \frac{\exp(e_{s,i})}{\sum_{j=t_s-w+1}^{t_s} \exp(e_{s,j})}. \end{aligned} \tag{11}$$

$\beta_{s,i}$ indicates an attention weight in the window.

4 EVALUATION

4.1 Dataset

A conventional JSL video dataset (Takayama et al., 2021b) was used to evaluate the proposed method. This dataset included 275 types of isolated word videos and 113 types of continuous word videos performed by 37 signers. All signers were adults who have experience in JSL. The vocabulary was related to the conversation at the city office. The videos were recorded with a smartphone camera. All the video frames were recorded at 30 frames per second with 640×360 pixels.

Figure 3 shows an example of a continuous word video. As shown in Figure 3, the single signer sat on a chair and performed each word and sentence in front of the camera. The signers were posed in the static posture at the beginning and end of the sign. The frames between these static postures were defined as an action instance. The continuous word consists of "NYUUSEKI" and "KIBOU", which mean "registration of marriage" and "hope" in JSL. The combination of these words represents "I'd like to register our marriage." "short pause (SP)," "arm up (AU)," "transition (TR)," and "arm down (AD)" are marginal motions that do not have lexical meanings. The marginal motions were not included in the recognition targets in this study. The tracking points extracted by OpenPose were used as the inputs, and the raw video frames were discarded because this study focuses on skeleton-based sign language recognition.

Table 1: Summy of the dataset.

Subset types	Training		Test	
# of signers	35		2	
# of isolated words	22640	(275)	3862	(210)
# of sentences	7466	(113)	1372	(107)

The statistics of the dataset are summarized in Table 1. The number within the parentheses indicates the number of action types. Horizontally flipped tracking sequences were included to avoid the effect of the dominant hand.

4.2 Recognition Performance

The word error rate (WER) was used as the performance metric in this study.

$$WER = \frac{\mathrm{dist}(L_{ref}, L_{pred})}{|L_{ref}|} * 100. \tag{12}$$

$\mathrm{dist}(\cdot,\cdot)$ is Levenshtein distance. L_{ref} and L_{pred} are the ground truth and estimated word sequence, respectively. $|L|$ denotes the number of words in a sequence.

The training settings are described as follows: The input feature sequences and word sequences were padded to have the maximum lengths of $T = 578$ and $S = 13$ during the training, respectively. The batch size was 32 throughout the training of the recognition model. The learning rate was set to 0.0003, and the adaptive moment estimation method (Kingma and Ba, 2015) was used to update the parameters. The categorical cross-entropy was applied as a loss function. 150 training epochs were used for all the training procedures.

The situations where the input feature sequences are streamed in chunk units were considered for this study. The recognition performance was evaluated under the following three conditions.

- Offline: Chunk $= \{\boldsymbol{X}_{1:T}\}$,
- Online 1: Chunk $= \{\boldsymbol{X}_{1:u}, \ldots, \boldsymbol{X}_{iu+1:(i+1)u}, \ldots, \boldsymbol{X}_{nu+1:T}\}$,
- Online 2: Chunk $= \{\boldsymbol{X}_{1:u}, \ldots, \boldsymbol{X}_{1:(i+1)u}, \ldots, \boldsymbol{X}_{1:T}\}$.

A chunk size of $u = 30$ was employed for the online recognition. The model uses the input chunks unchanged in "Online 1". In contrast, the model buffers the past chunks and utilizes concatenated chunks for recognition in "Online 2."

The best word error rate during the training loop is described in Table 2. As shown in Table 2, the soft attention achieved the best WER 9.51 in the offline situation. However, the performance of the soft attention was significantly degraded in the online sit-

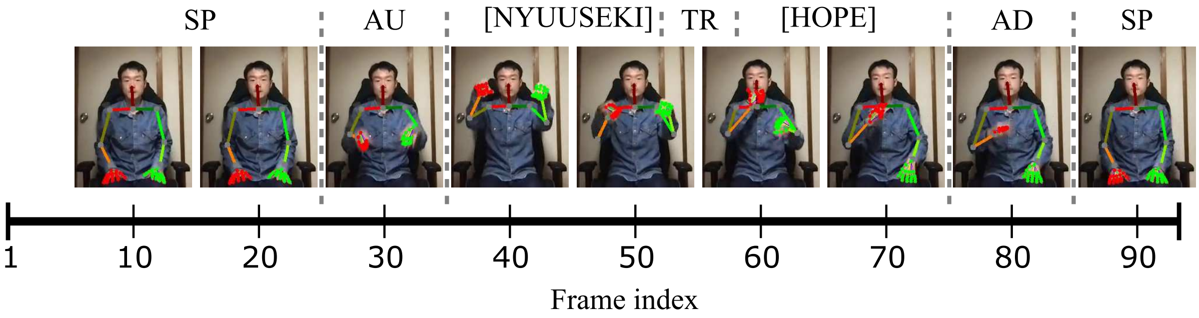
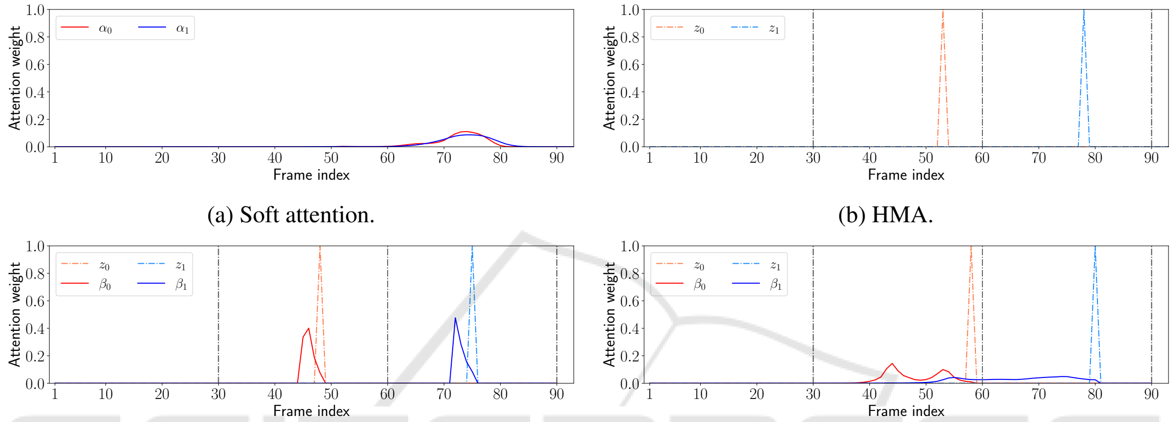


Figure 3: Example of a JSL video. The vertical dashed lines indicate the borders of motion units.



(a) Soft attention.

(b) HMA.

(c) MoChA.

(d) MILK.

Figure 4: Comparison of the attention weight's behaviors. The vertical dashed lines in (b)-(d) indicate the borders of the input chunks. The solid and dashed lines indicate soft attention and hard attention weights, respectively.

uations because these situations violate the soft attention's prerequisites that entire frames are available. The monotonic attention methods retained similar performance in all situations. As shown in the results of the online situations, HMA and MILK had the best WERs 19.47 and 11.08 for “Online 1” and “Online 2”, respectively. The performance of MILK was degraded in “Online 1” because this situation violates MILK's prerequisites that all past frames are available. MoChA had the balanced performances in all cases, and it achieved the second-best WERs 20.03 and 12.88 for “Online 1” and “Online 2”, respectively.

Table 2: Recognition performance [%].

Model	Offline	Online 1	Online 2
Soft	**9.51**	75.27	72.22
HMA	12.16	**19.47**	13.65
MoChA	11.12	20.03	12.88
MILK	10.89	31.19	**11.08**

Finally, the behaviors of attention weights of each attention layer are shown in Figure 4. Figure 4 (a), (b), (c), and (d) show the behaviors of attention weights generated by the standard soft attention, HMA, MoChA, and MILK, respectively. “Offline,” “Online1,” and “Online2” were applied to the standard soft attention, HMA and MoChA, and MILK, respectively, to generate the attention weights in Figure 4.

As shown in α_0 and α_1 in Figure 4 (a), the standard soft attention gave weights to the entire input sequence. The standard soft attention preferred the latter part of the input sequence in this examination. It is expected that the features of this part were sufficient to infer words because there was only one type of two-word sentence in the dataset. The standard soft attention does not always attend to the part of the sequence that matches the word, and it requires the entire sequence for inference. Therefore, the standard soft attention is difficult to use in online situations. In contrast, as shown in z_0 and z_1 in Figure 4 (b), (c), and (d), the monotonic attention has succeeded in inferring the first and second words for the second and third chunks, respectively, in the online situations. Moreover, as shown in β_0 and β_1 in Figure 4 (c) and (d), MoChA and MILK gave appropriate weights to the sequence parts representing each word.

5 CONCLUSION

In this study, skeleton-based online sign language recognition using monotonic attention was investigated. A total of three monotonic attention techniques were applied to continuous sign language word recognition based on the STGCN-RNNA model. The effectiveness of the monotonic attention for online continuous sign language word recognition was demonstrated through the results of the evaluation using the JSL video dataset.

Seq2Seq-based online recognition has been well researched within the speech recognition and natural language processing domains. Recently, Transformer-based online recognition methods have also been proposed (Tsunoo et al., 2019; Inaguma et al., 2020; Miao et al., 2020; Li et al., 2021) in the field. Future studies will include investigations on the applicability of these methods to online sign language recognition.

Furthermore, the authors have considered online sign language translation as an interesting research topic for future studies. The techniques for simultaneous translation in natural language processing (Gu et al., 2017; Dalvi et al., 2018; Ma et al., 2019) can be expected to contribute in this direction.

ACKNOWLEDGEMENTS

This research is supported by SoftBank Corp.

REFERENCES

Arivazhagan, N., Cherry, C., Macherey, W., Chiu, C.-C., Yavuz, S., Pang, R., Li, W., and Raffel, C. (2019). Monotonic infinite lookback attention for simultaneous machine translation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1313–1323.

Bahdanau, D., Cho, K., and Bengio, Y. (2015). Neural machine translation by jointly learning to align and translate. In *Proceedings of the Third International Conference on Learning Representations*, pages 1–15.

Camgoz, N. C., 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.

Camgoz, N. C., Koller, O., Hadfield, S., and Bowden, R. (2020). Sign language transformers: Joint end-to-end sign language recognition and translation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 10023–10033.

Cao, Z., Hidalgo, G., Simon, T., Wei, S.-E., and Sheikh, Y. (2021). Openpose: Realtime multi-person 2d pose estimation using part affinity fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 43(1):172–186.

Chiu, C.-C. and Raffel, C. (2018). Monotonic chunkwise attention. In *Proceedings of the Sixth International Conference on Learning Representations*, pages 1–16.

Cho, K., van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., and Bengio, Y. (2014). Learning phrase representations using RNN encoder–decoder for statistical machine translation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 1724–1734.

Cooper, H., Pugeault, N., and Bowden, R. (2011). Reading the signs: A video based sign dictionary. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pages 914–919.

Cui, R., Liu, H., and Zhang, C. (2019). A deep neural framework for continuous sign language recognition by iterative training. *IEEE Transactions on Multimedia*, 21(7):1880–1891.

Dalvi, F., Durrani, N., Sajjad, H., and Vogel, S. (2018). Incremental decoding and training methods for simultaneous translation in neural machine translation. In *Proceedings of the Annual Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 493–499.

De Coster, M., Van Herreweghe, M., and Dambre, J. (2020). Sign language recognition with transformer networks. In *Proceedings of the Twelveth Language Resources and Evaluation Conference*, pages 6018–6024.

Forster, J., Koller, O., Oberdörfer, C., Gweth, Y., and Ney, H. (2013). Improving continuous sign language recognition: Speech recognition techniques and system design. In *Proceedings of the Fourth Workshop on Speech and Language Processing for Assistive Technologies*, pages 41–46.

Gu, J., Neubig, G., Cho, K., and Li, V. O. (2017). Learning to translate in real-time with neural machine translation. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics*, pages 1053–1062.

Guo, D., Zhou, W., Li, A., Li, H., and Wang, M. (2020). Hierarchical recurrent deep fusion using adaptive clip summarization for sign language translation. *IEEE Transactions on Image Processing*, 29:1575–1590.

Huang, J., Zhou, W., Zhang, Q., Li, H., and Li, W. (2018). Video-based sign language recognition without temporal segmentation. In *Proceedings of the 32nd AAAI Conference on Artificial Intelligence*, pages 2257–2264.

Inaguma, H., Mimura, M., and Kawahara, T. (2020). Enhancing monotonic multihead attention for streaming asr. In *Proceedings of the 21st INTERSPEECH*, pages 2137–2141.

Jiang, S., Wang, L., Bai, Y., Li, K., and Fu, Y. (2021). Skeleton aware multi-modal sign language recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*.

Kingma, D. P. and Ba, J. (2015). Adam: A method for stochastic optimization. In *Proceedings of the third International Conference on Learning Representations*.

Koller, O., Camgoz, N. C., Ney, H., and Bowden, R. (2020). Weakly supervised learning with multi-stream cnn-lstm-hmms to discover sequential parallelism in sign language videos. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 42(9):2306–2320.

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 3416–3424.

Kumar, D. A., Sastry, A. S. C., Kishore, P. V. V., Kumar, E. K., and Kumar, M. T. K. (2019). S3drgf: Spatial 3-d relational geometric features for 3-d sign language representation and recognition. *IEEE Signal Processing Letters*, 26(1):169–173.

Li, D., Rodriguez, C., Yu, X., and Li, H. (2020). 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*, pages 1459–1469.

Li, M., Zorilă, C., and Doddipatla, R. (2021). Transformer-based online speech recognition with decoder-end adaptive computation steps. In *Proceedings of the IEEE Spoken Language Technology Workshop*, pages 1–7.

Liu, X. and Di, X. (2020). Tanhexp: A smooth activation function with high convergence speed for lightweight neural networks. arXiv preprint arXiv:2003.09855.

Ma, M., Huang, L., Xiong, H., Zheng, R., Liu, K., Zheng, B., Zhang, C., He, Z., Liu, H., Li, X., Wu, H., and Wang, H. (2019). Stacl: Simultaneous translation with implicit anticipation and controllable latency using prefix-to-prefix framework. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3025–3036.

Miao, H., Cheng, G., Gao, C., Zhang, P., and Yan, Y. (2020). Transformer-based online ctc/attention end-to-end speech recognition architecture. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 6084–6088.

Moryossef, A., Tsochantaridis, I., Aharoni, R., Ebling, S., and Narayanan, S. (2020). Real-time sign language detection using human pose estimation. In *Proceedings of the European Conference on Computer Vision Workshops, LNCS 12536*, pages 237–248.

Papastratis, I., Dimitropoulos, K., Konstantinidis, D., and Daras, P. (2020). Continuous sign language recognition through cross-modal alignment of video and text embeddings in a joint-latent space. *IEEE Access*, 8:91170–91180.

Pu, J., Zhou, W., and Li, H. (2019). Iterative alignment network for continuous sign language recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4160–4169.

Raffel, C., Luong, M.-T., Liu, P. J., Weiss, R. J., and Eck, D. (2017). Online and linear-time attention by enforcing monotonic alignments. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70, pages 2837–2846.

Takayama, N., Benitez-Garcia, G., and Takahashi, H. (2021a). Masked batch normalization to improve tracking-based sign language recognition using graph convolutional networks. In *Proceedings of the IEEE International Conference on Automatic Face and Gesture Recognition*.

Takayama, N., Benitez-Garcia, G., and Takahashi, H. (2021b). Sign language recognition based on spatial-temporal graph convolution-transformer. *Journal of Japan Society for Precision Engineering*, 87(12):1028–1035.

Tsunoo, E., Kashiwagi, Y., Kumakura, T., and Watanabe, S. (2019). Towards online end-to-end transformer automatic speech recognition. arXiv preprint arXiv:1910.11871.

Yan, S., Xiong, Y., and Lin, D. (2018). Spatial temporal graph convolutional networks for skeleton-based action recognition. In *Proceedings of the 32nd AAAI Conference on Artificial Intelligence*, pages 7444–7452.

Zhou, H., Zhou, W., Qi, W., Pu, J., and Li, H. (2021). Improving sign language translation with monolingual data by sign back-translation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1316–1325.

Zhou, M., Ng, M., Cai, Z., and Ka, C. C. (2020). Self-attention-based fully-inception networks for continuous sign language recognition. In *Proceedings of the 24th European Conference on Artificial Intelligence*, volume 325 of *Frontiers in Artificial Intelligence and Applications*, pages 2832–2839.