Continuous Sign-Language Recognition using Transformers and Augmented Pose Estimation

Reemt Hinrichs, Angelo Jovin Yamachui Sitcheu and Jörn Ostermann
Institut für Informationsverarbeitung/L3S Research Center, Leibniz University Hannover, Appelstr. 9a, Hannover, Germany

Keywords: Transformer, Continuous Sign-Language Recognition, Pose Estimation.

Abstract: Sign language is used by deaf to communicate with other humans. It consists of not only hand signs or gestures but encompasses also facial expressions and further body movements. To make machine-human interaction accessible for deaf, automatic sign language recognition has to be implemented which allows a machine to understand the signs and gestures of deaf. For this purpose, continous sign-language recognition, which is the mapping of a (visual) sequence of signs forming a (sign) sentence to a sequence of (text) words, has to be developed. In this work, continuous sign-language recognition using transformers is proposed. Using additional pose estimation, body markers are extracted and augmented through data imputation and velocity-like features, and then used together with a transformer network for continuous sign-language recognition. Using the proposed method, better than state-of-the-art results were obtained on the RWTH-PHOENIX-Weather 2014 dataset, achieving 19.2 %/19.5 % dev/test word error rate (WER) on the signer-independent subset and 16.9 %/17.4 % dev/test WER on the simpler multi-signer subset. The feature augmentation was found to improve the baseline word error rate by about 2.7 %/ 2.9 % dev/test.

1 INTRODUCTION

Current estimates suggest that about 20 % of the entire population of the world live with hearing loss, about 430 million of which suffering from disabling hearing loss (Chadha and Cieza, 2017). Around the world, a subset of these, the completely deaf, commonly use sign language to communicate with other people. Due to these factors, automatic recognition of sign language is a very important task to be mastered by machines (Wen et al., 2021). As sign language in practice consists of a sequence of combinations of gestures/hand signs, facial expressions and further body poses (together called a gloss), to properly recognize and understand sign language, sequences of glosses have to be mapped to sequences of words. In the instance, where no temporal boundaries between individual glosses are known, the problem is called continuous sign-language recognition. While the automatic recognition as described is already difficult enough, the fact that many different sign-languages (and dialects) are used throughout the world increases a complete solution of the problem even more. However, only one sign-language, the german standard sign-language, is considered in this work.

1.1 Related Work

Early work about vision based automatic sign-language recognition dates back at least to the 80s (Tamura and Kawasaki, 1988) with the field apparently getting track in the mid 90s (Starner and Pentland, 1995; Vogler and Metaxas, 1998), where hidden markov models were used to recognize hand gestures of a subset of the american sign-language. Large improvements in automatic sign-language recognition were achieved in the 2010s with the triumphant advance of machine learning or more specifically artifical neural networks, especially in the general field of computer vision. A good overview can be found in (Rastgoo et al., 2021). Pigou et al. (Pigou et al., 2015) achieved 91.4 % accuracy using convolutional neural networks for feature extraction and classification of 20 Italien gestures. Kumar et al. (Kumar et al., 2018) extracted face and hand features and used independent bayesian classifier combination for recognition of 51 dynamic sign word gestures. They achieved an accuracy of up to 96.04 % on this custom dataset. Mittal et al. (Mittal et al., 2019) used modified LSTM as well as a convolutional neural network to recognize sequences of hand gestures. They achieved an accuracy of 72.3 % for continuous sign language recognition on their own dataset.

Hinrichs, R., Sitcheu, A. and Ostermann, J.
Continuous Sign-Language Recognition using Transformers and Augmented Pose Estimation.
DOI: 10.5220/0011709100003411
In *Proceedings of the 12th International Conference on Pattern Recognition Applications and Methods (ICPRAM 2023)*, pages 672-678
ISBN: 978-989-758-626-2; ISSN: 2184-4313

Recently, on the RWTH-PHOENIX-Weather 2014 dataset (Koller et al., 2015), which this work focuses on and to be described in later sections, Cui et al. (Cui et al., 2017) combined convolutional neural networks with bidirectional long-short term memory (BiLSTM), one of the most popular approaches in the literature, and used staged optimization. They achieved a word error rate of 39.4 %/38.7 % dev/test.

Zhou et al. (Zhou et al., 2020) used multi-cue networks consisting of convolutional layers for pose and feature extraction and BiLSTM for mapping the extracted features to words. With their approach, they achieved a word error rate (WER) of 21.1 %/20.7 % dev/test. Papastratis et al. (Papastratis et al., 2021) achieved 23.7 %/23.4 % dev/test WER using a context-aware generative adversarial network. Most recently, Hu et al. (Hu et al., 2022), by combining so called temporal lift pooling with BiLSTMs, achieved a state-of-the-art WER of 19.7 %/20.8 % dev/test, for the first time obtaining below 20 % dev WER. Furthermore, Zuo et al. (Zuo and Mak, 2022) achieved a WER of 20.5 %/20.4 % dev/test, which is state-of-the-art for the test set, through combining spatial attention consistency with transformer networks.

While some publications make use of transformer models, and a few make use of dedicated pose/landmark estimation frameworks, no publication uses data imputation techniques to improve the landmark estimation before attempting to recognize the glosses. However, in (Bansal et al., 2021) the possible gains by superior landmark extraction became apparent. Therefore we believe, that a considerable improvement in word error rate can be achieved by combining state-of-the-art landmark estimation with data imputation techniques to decrease the error in the landmark estimation.

1.2 Contribution

In this work, we propose to first extract landmarks of the respective signers of the used dataset, to error correct these landmarks using a k nearest neighbor neural network and to augment these landmarks by several features, including velocity-like features as in (Bansal et al., 2021) as well as features allowing to recognize gloss borders more easily. The feature augmentation is described in detail in Section 2.3. Then, a transformer neural network is used to map this sign language representation to the german words. For landmark extraction, we use the recently published framework MediaPipe (Lugaresi et al., 2019), which we found to be rather reliable except in case of strong motion blur and some occlusions. Error correction targeted these problematic instances, where the landmark extraction failed, and attempted to interpolate missing markers. This work is structured as follows: First, the dataset, the feature augmentation and the utilized transformer network are described in Section 2. Then, the obtained word error rates as well as an ablation study and error analysis are presented in Section 3. These results are then discussed and compared to other authors in Section 4. The manuscript concludes in Section 5.

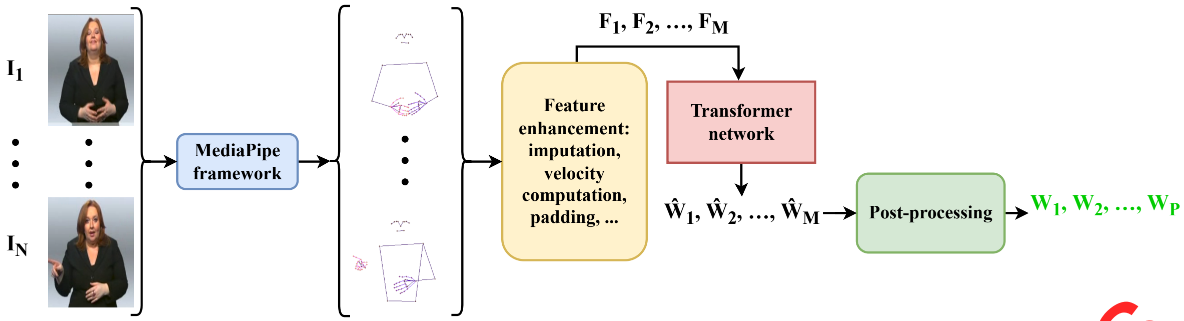
2 METHODS AND MATERIALS

2.1 Dataset

In this work the RWTH-PHOENIX-Weather 2014 continuous sign language recognition dataset was used. It consists of video recordings of the sign language transcript of the german weather forecast throughout the years 2011-2013 as shown on the german television channel Phoenix. The sampling rate f_s of all videos is 25 frames per second, each frame being of size 210 x 260 pixels. The videos show only the box of the signer. In total, nine different signers make up the entire dataset. The dataset is split into two subsets: a multi signer subset and a signer independent subset:

- Multi signer subset: This subset has a total number of 6841 videos, 77271 tokens/glosses and 837865 frames. It is divided into three data splits: a train set of 5672 samples, a development set with 540 samples and a test set of 629 samples. All nine signers appear in the three data splits. This subset has a vocabulary size of 1295 signs.
- Signer independent subset: This subset has a total number of 4667 videos, 53034 tokens/glosses and 655378 frames. It is divided into three data splits: a train set with 4376 samples, a development set with 111 samples and, a test set of 180 samples. With the exception of signer 05, all nine signers appear in the train set. The development set and the test set are signed only by the unseen signer 05. This subset has a vocabulary size of 1135 signs.

Due to the developement and test set of the signer independent subset being signed by the only signer not included in its train set, it poses a considerable greater challenge than the multi signer subset.



I_1
I_N
MediaPipe framework
Feature enhancement: imputation, velocity computation, padding, ...
$F_1, F_2, ..., F_M$
Transformer network
$\hat{W}_1, \hat{W}_2, ..., \hat{W}_M$
Post-processing
$W_1, W_2, ..., W_P$

Figure 1: Pipeline of the proposed algorithm. For each image of a sequence of glosses markers are extracted using the MediaPipe framework. In this example, MediaPipe failed to extract the right elbow landmark. Then, feature augmentation is performed including velocity features. These augmented features are fed to a transformer network which generates an output for each frame of the video, most being padding symbols included after the final recognized gloss. Finally, in the post-processing, the padding symbols are removed, yielding the final output of words $W_1,\dots,W_p$.

2.2 Feature Extraction

For feature extraction the framework MediaPipe, recently published by Google (Lugaresi et al., 2019), was used in its python implementation (MediaPipe, 2020). It was used to extract a total of 75 landmarks covering hands, arms, shoulders and parts of the face. The extracted landmarks can be seen in Fig. 1. Except for frames exhibiting heavy motion blur and some occlusions of one of the hands the marker extraction appeared to be very reliable. Each landmark $\mathbf{M}^{(i)}$ is represented by its three cartesian coordinates $x^{(i)}, y^{(i)}$ and $z^{(i)}$, i.e. we can identify according to $\mathbf{M^{(i)}} = (x^{(i)}, y^{(i)}, z^{(i)})$.

2.3 Feature Augmentation

The sequence of length N of L feature vectors $\mathbf{F_n} := (\mathbf{M_n^{(1)}},\dots,\mathbf{M_n^{(L)}})$ with landmarks $\mathbf{M_n^{(i)}} = (x_n^{(i)}, y_n^{(i)}, z_n^{(i)})$, $n = 1,2,\dots,N$ and $L = 75$, obtained by applying MediaPipe to a video recording consisting of N frames, was augmented in several ways: the sequence $\mathbf{F_n}$ was corrected by a k nearest neighbor algorithm as performed in (Yao and Ruzzo, 2006) implemented through sklearn. This k nearest neighbor regression was used only to correct any marker $\mathbf{M_n^{(i)}}$ which MediaPipe failed to extract, which are indicated by empty dictionary entries. For its interpolation it used the adjacent markers of future or previous time steps. Furthermore, the L2-norm of $\Delta M := (\mathbf{M_{n+1}^{(1)}} - \mathbf{M_n^{(1)}},\dots,\mathbf{M_{n+1}^{(L)}} - \mathbf{M_n^{(L)}})$ was used as an additional feature. The idea was, that minima of the velocity of the signer could help to identify the borders of the individual glosses, where a signer could come to a brief hold. Thirdly, the center of mass of each hand was included as a feature. Finally, the velocity of both hands and the individual fingers was used as an additional feature, i.e.

$$\mathbf{v_n^{(i)}} = \frac{\mathbf{M_{n+1}^{(i)}} - \mathbf{M_n^{(i)}}}{T} \qquad (1)$$

with $n = 1,\dots,N-1$ and i iterating through all markers of each hand including the center of mass and $T = \frac{1}{f_s}$. The substraction in Eq. 1 is to be understood componentwise in the natural sense. The idea was motivated by glosses like Regen (engl. rain) and Schnee (engl. snow) which use identical gestures except for the movement of the fingers, where for snow the fingers are moving and resting for rain.

2.4 Transformer Structure

The encoder and decoder of the transformer consisted of eight layers, each using eight heads. The embedding dimension d was set to $d = 512$. The expansion factor e was set to four, yielding an input/output size of the linear layers of $e \cdot d = 2048$. The total number of parameters was 108,859,665. Initital results suggested that reducing the number of parameters to 80 million decreases the performance to some degree. Larger numbers were not tested due to the computational complexity involved. Positional encoding used sine and cosine encoding, as described in (Vaswani et al., 2017). Swish served as activation function. The maximum sequence length of both, encoder and decoder, was set to M = 300, which was the maximum number of frames encountered in the entire dataset. Given a ground truth sentence of W words, during training the target output was padded at the end using M-W padding symbols.

2.5 Proposed Algorithm

The overall algorithm is depicted in Fig. 1. A video consisting of N frames is fed to MediaPipe

Table 1: Word error rate (WER) and share of insertion or deletion errors achieved by the Full learner on the development and test set of the multisigner subset. Only substitution errors occured.

Subset	#ins/#del (%) and WER (%)		
	del/ins	WER Dev	WER Test
Signer Independent	0.0/0.0	19.72	19.52
Multisigner	0.0/0.0	16.91	17.39

which extracts the landmarks. These are augmented as described in Sec. 2.3, yielding feature vectors $\mathbf{F_1}, \dots, \mathbf{F_M}$ with $\mathbf{F_k} = 0$ when $k > N$. These were then propagated to the transformer. The transformer maps this sequence of feature vectors to sequences of words $\hat{\mathbf{W}}_\mathbf{1}, \dots, \hat{\mathbf{W}}_\mathbf{M}$. This output still contains padding symbols at the end. In the final step, all padding symbols are removed, yielding the final output sequence $\mathbf{W_1}, \dots, \mathbf{W_P}$ with $\mathbf{W_k} = \hat{\mathbf{W}}_\mathbf{k}$ for $k = 1, \dots, P$ to obtain the final output of the algorithm. This output was then used to assess the performance, either through the cross-entropy loss or the word error rate.

2.6 Training

In all cases, the transformer was trained for 220 Epochs using the cross-entropy loss, a batch size of four and a learning rate of 0.09 using stochastic gradient descent. The learning rate was updated with a factor of 0.7 if the evaluation loss did not improve across two epochs. Dropout was set to 0.2 in all layers. A fixed random seed was used such that all transformers were initialized identically.

2.7 Evaluation

The main metric to assess the performance of the proposed algorithm is the word error rate (WER) on the respective dev and test sets. The WER is computed according to

$$WER = \left(\frac{\#insertions + \#deletions + \#substitutions}{\#words}\right) \cdot 100\,\% \qquad (2)$$

where #insertions, #deletions and #substitutions as well as #words are the respective sum across the entire respective dev or test set.

An ablation study was performed to evaluate the benefit of the feature augmentation. As baseline model, labeled MediaPipe learner, the immediate output of MediaPipe served as input of the transformer. Empty outputs of MediaPipe were replaced with zeros. Next, the MediaPipe learner was augmented by the L2-norm feature as described in Section 2.3 and this model was labeled L2 learner. The L2 Learner was augment through data imputation as described in Section 2.3 and this model was labeled Imputed learner. Finally, the Imputed learner was augment by all other feature augmentations described in Section 2.3, i.e. the center of gravity of each hand was added as well as the velocity of the handmarks. This model was labeled Full learner.

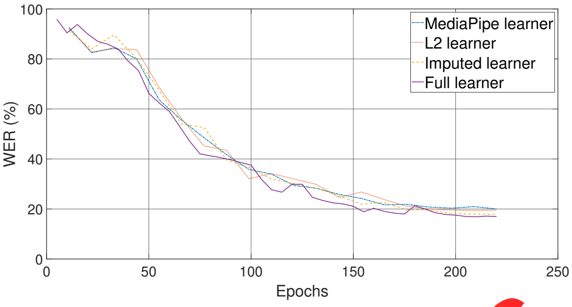


Figure 2: Evolution of the word error rate (WER) of the respective ablation models on the development set of the multisigner subset. The best word error rates are summarized in Table 2.

3 RESULTS

The word error rates achieved on the development and test set of the multisigner and signer independent subsets are reported in Table 1 together with the amount of deletion and insertion errors in percent. Actually, only substitution errors occured. Better than state-of-the-art word error rates of 16.9/17.4 % dev/test were achieved on the multisigner subset and state-of-the-art word error rates of 19.7/19.5 % dev/test were achieved on the signer independent dataset. A comparison to the previous state-of-the-art and a selection of further relevant results is given for the multi signer subset in Table 3 and for the signer independent subset in Table 4. To the best of our knowledge, we improved the state-of-the-art for the multisigner subset by 2.8 % on the developement set and 3 % on the test set. Furthermore, To the best of our knowledge, we improved the state-of-the-art for the signer independent subset by 25.4 % on the development set and by

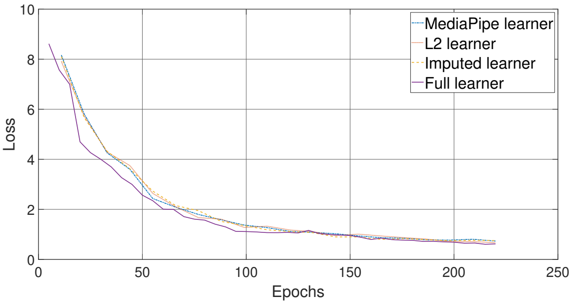


Figure 3: Evaluation loss across training epochs for all ablation models. It is apparent, that the training had not yet quite fully converged.

Table 2: Best word error rate and share of insertion or deletion errors achieved by each learner on the multisigner dev and test set.

Learner	WER (%) and #del/ins (%)		
	del/ins	Dev	Test
MediaPipe learner	0.0/0.0	19.66	20.20
L2 learner	0.0/0.0	19.54	20.09
Imputed learner	0.0/0.0	17.88	17.50
Full learner	0.0/0.0	16.91	17.39

24.6 % on the test set. Note that for the signer independent subset, which poses a considerably more difficult problem, only one result was found in the literature.

3.1 Ablation

Word error rates for all ablation models are summarized in Table 2. The evolution of the word error curve for the development set is depicted in Fig. 2. Around epoch 140 the differences between the ablation models become somewhat apparent. The word error rate of the vanilla MediaPipe learner was 19.7 %/20.2 % dev/test and for the Full learner 16.9 %/17.4 %. This results in an absolute improvement of 2.8 %/2.8 % dev/test. The loss curve for the development set of the multi signer subset for all ablation models is depicted in Fig. 3. As the slope of the evaluation curves had not yet approached zero, it is reasonable to assume that the word error rate could have been improved even further if the training had continued.

4 DISCUSSION

Going by the word error rate of the MediaPipe learner as given in Table 2, the transformer achieves out-of-the-box, aside from some initial investigations regarding, e.g., the learning rate, state-of-the-art results. This is impressive, seeing that the previous state-of-the-art approaches (Hu et al., 2022; Zuo and Mak, 2022) required a considerable greater engineering effort. However, seeing that continuous sign language recognition at its core means to learn a translation of sequences of feature vectors, which could be interpreted as a special representation of a language, to sequences of words, its performance is not that surprising anymore. Transformers continue to have great success in natural language processing (Chernyavskiy et al., 2021), and as such seem to be well suited for the issue of continuous sign language recognition once the three language streams, i.e. hand signs, facial expressions and other body movements, are adequately captured in the input data.

Table 3: Word error rate (WER) of the Full learner as defined in Section 2.7 on the development and test set of multisigner subset together with a selection of results from the literature, including the previous state-of-the-art.

Method	WER (%)	
	Dev	Test
(Koller et al., 2015) CSLR	55.0	53.0
(Cihan Camgoz et al., 2017) SubUNets	40.8	40.7
(Chen et al., 2021) RL transformer	38.0	38.3
(Koller et al., 2017) CNNs-BiLSTM	27.1	26.8
(Hao et al., 2021) SMKD	20.8	21.0
(Aditya et al., 2022) Spatio-Temporal CSLR	20.5	21.5
(Zuo and Mak, 2022) C2SLR	20.5	20.4
(Hu et al., 2022) Temporal lift pooling	19.7	20.8
Ours	**16.9**	**17.4**

Table 4: Word error rate (WER) of the Full learner as defined in Section 2.7 on the development and test set of signer independent subset together with a selection of results from the literature, including the previous state-of-the-art.

Method	WER (%)	
	Dev	Test
(Koller et al., 2017) Re-Sign	45.1	44.1
Ours	**19.7**	**19.5**

The benefit of the L2-norm was minor, which could be explained by only some body parts, e.g., the hands, being close to resting when a signer emphasizes a gloss. Perhaps introducing seperate L2-norms for individual body parts could help to identify gloss borders.

The greatest benefit was observed by data imputation. The explanation is obviously that important landmarks are sometimes not extracted by MediaPipe and that the transformer cannot restore them on its own. Noisy imputation appears to be better than missing data.

The introduction of the velocity of the landmarks and the center of gravity of each hand gave a noticeable boost on the development set, however, on the test set only a marginal improvement of about 0.1 % was observed. Such a small difference might be not systematic. Generally, the transformer should be able to compute velocity on its own. Introducing landmark velocity despite this fact was motivated by the fact that a neural network generally only attains local minima. Adding reasonable features might help to steer the training towards better local minima.

Due to the computational complexity, repeating the ablations a few times was impossible. Thus chance cannot be entirely ruled out as an explanation of the observed differences in the ablation study. However, because we fixed the random seed, the initialization was identical for all four ablations. This suggests the features as the cause of the improved word error rate.

The proposed approach to continuous sign language recognition has a few obvious advantages compared to other methods: Due to extracting key landmarks first with a separate, reliable framework, the feature extraction generalizes out of the box to novel data. This explains the very similiar performance of our approach on the more difficult signer independent subset. Additionally, it can be in principle guaranteed, that only relevant features are considered and the causes of recognition errors can be more easily tracked to either feature extraction or the mapping from feature space to output space.

A possible downside, shared with others, of the current approach is the necessity of setting some fixed maximum number of input and output symbols. While this could be set rather high, and thus should not be greatly disadvantagous in practice, it is a flaw in the design, as a dynamic output length would be desirable. For that, the proposed algorithm would have to output the respective gloss boundaries, such that the input video sequence can be cut at the beginning and the corresponding word could be flushed out of the output buffer.

One of the most glaring differences to other works is the absence of insertion and deletion errors of our model. Only substitution errors occured. In fact, the number of words at the output of the transformer after the post-processing stage always matched the ground truth number of words. This suggests, that the transformer is able to correctly estimate gloss borders resulting in the correct number of individual glosses. During the training, in earlier epochs, insertion and deletion errors still occur but vanish near the end of the training. We double checked our code and analyses to verify its correctness and were unable to find any mistake. No other work combined MediaPipe with transformers on the RWTH-Phoenix-Weather 2014 dataset and as such we cannot make detailed comparisons to other works in this specific point. All learners of the ablation making only substitution errors can be attributed to the fixed random seed used and the lack of cross-validation due to the involved computational complexity.

Despite the promising results of this work, the word error rate still appears to be way too high for practical applications. The word error rate translates to about one incorrect word out of six. A reasonable recognition system should achieve word error rates well below 10 %. Furthermore, in real situations, noise due to background humans should pose a considerable problem.

4.1 Comparison to Other Work

Perhaps the publication closest to our work is (Bansal et al., 2021), where MediaPipe alongside other pose estimation frameworks was also combined with a transformer network, and even velocity features were considered, although the authors did not report a ablation results to assess the feature importants. Their work however is concerned with american sign language. They also found the combination of MediaPipe and a transformer to perform very well, however, they found hidden markov models to perform better. More specifically, MediaPipe was found to perform the best in conjunction with the transformer out of three evaluated pose estimation frameworks and improved the performance by up to about 10 % with respect to the worst performing pose estimation framework. This underlines the importance quality of the landmark extraction.

In (Camgoz et al., 2020) a transformer was also used for sign language recognition and translation. The authors report a word error rate of 24.6 &/24.5 % dev/test for their best implementation on the very similiar PHOENIX14T dataset, which offers only a marginally smaller vocabulary. These word error rates are about 5 % worse than our corresponding MediaPipe learner. Their transformer network appears to be identical to the one used in this work. Specifically, the number of parameters was likely the same as ours. They did not, however, use MediaPipe for feature extraction but rather their own convolutional neural networks. These might extract suboptimal features. Considering the benefit of using MediaPipe in conjunction with a transformer according to (Bansal et al., 2021), this appears to be a rather plausible explanation and is in the range of improvement observed by Bansal et al. (Bansal et al., 2021) in a few instances.

5 CONCLUSIONS

This work investigated automatic continuous sign language recognition using transformers on the RWTH-Phoenix-Weather 2014 dataset. For feature extraction, Google's MediaPipe framework was applied. Through feature augmentation that included introducing velocity information, state-of-the-art word error rates of 16.9 %/17.4 % dev/test were achieved on the multisigner subset and state-of-the-art word error rates of 19.7 %/19.5 % dev/test were achieved on the more difficult signer independent subset. The feature augmentation was found to improve the baseline word error rate by about 2.7 %/ 2.9 % dev/test.

REFERENCES

Aditya, W., Shih, T. K., Thaipisutikul, T., Fitriajie, A. S., Gochoo, M., Utaminingrum, F., and Lin, C.-Y. (2022). Novel spatio-temporal continuous sign language recognition using an attentive multi-feature network. *Sensors*, 22(17):6452.

Bansal, D., Ravi, P., So, M., Agrawal, P., Chadha, I., Murugappan, G., and Duke, C. (2021). Copycat: Using sign language recognition to help deaf children acquire language skills. In *Extended Abstracts of the 2021 CHI Conference on Human Factors in Computing Systems*, CHI EA '21. Association for Computing Machinery.

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

Chadha, S. and Cieza, A. (2017). Promoting global action on hearing loss: World hearing day. *International Journal of Audiology*, 56(3):145–147. PMID: 28262049.

Chen, L., Lu, K., Rajeswaran, A., Lee, K., Grover, A., Laskin, M., Abbeel, P., Srinivas, A., and Mordatch, I. (2021). Decision transformer: Reinforcement learning via sequence modeling. *CoRR*, abs/2106.01345.

Chernyavskiy, A., Ilvovsky, D., and Nakov, P. (2021). Transformers: "the end of history" for natural language processing? In Oliver, N., Pérez-Cruz, F., Kramer, S., Read, J., and Lozano, J. A., editors, *Machine Learning and Knowledge Discovery in Databases. Research Track*, pages 677–693, Cham. Springer International Publishing.

Cihan Camgoz, N., Hadfield, S., Koller, O., and Bowden, R. (2017). Subunets: End-to-end hand shape and continuous sign language recognition. In *Proceedings of the IEEE international conference on computer vision*, pages 3056–3065.

Cui, R., Liu, H., and Zhang, C. (2017). Recurrent convolutional neural networks for continuous sign language recognition by staged optimization. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1610–1618.

Hao, A., Min, Y., and Chen, X. (2021). Self-mutual distillation learning for continuous sign language recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 11303–11312.

Hu, L., Gao, L., Liu, Z., and Feng, W. (2022). Temporal Lift Pooling for Continuous Sign Language Recognition. arXiv:2207.08734.

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

Koller, O., Zargaran, S., and Ney, H. (2017). Re-sign: Re-aligned end-to-end sequence modelling with deep recurrent cnn-hmms. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3416–3424.

Kumar, P., Roy, P. P., and Dogra, D. P. (2018). Independent bayesian classifier combination based sign language recognition using facial expression. *Information Sciences*, 428:30–48.

Lugaresi, C., Tang, J., Nash, H., McClanahan, C., Uboweja, E., Hays, M., Zhang, F., Chang, C.-L., Yong, M. G., Lee, J., Chang, W.-T., Hua, W., Georg, M., and Grundmann, M. (2019). Mediapipe: A framework for building perception pipelines. arXiv:1906.08172.

MediaPipe (2020). https://github.com/google/mediapipe, last accessed: 25.10.2022.

Mittal, A., Kumar, P., Roy, P. P., Balasubramanian, R., and Chaudhuri, B. B. (2019). A modified lstm model for continuous sign language recognition using leap motion. *IEEE Sensors Journal*, 19(16):7056–7063.

Papastratis, I., Dimitropoulos, K., and Daras, P. (2021). Continuous sign language recognition through a context-aware generative adversarial network. *Sensors*, 21(7).

Pigou, L., Dieleman, S., Kindermans, P.-J., and Schrauwen, B. (2015). Sign language recognition using convolutional neural networks. In Agapito, L., Bronstein, M. M., and Rother, C., editors, *Computer Vision - ECCV 2014 Workshops*, pages 572–578, Cham. Springer International Publishing.

Rastgoo, R., Kiani, K., and Escalera, S. (2021). Sign language recognition: A deep survey. *Expert Systems with Applications*, 164:113794.

Starner, T. and Pentland, A. (1995). Real-time american sign language recognition from video using hidden markov models. In *Proceedings of International Symposium on Computer Vision - ISCV*, pages 265–270.

Tamura, S. and Kawasaki, S. (1988). Recognition of sign language motion images. *Pattern Recognition*, 21(4):343–353.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., and Polosukhin, I. (2017). Attention is all you need.

Vogler, C. and Metaxas, D. (1998). Asl recognition based on a coupling between hmms and 3d motion analysis. In *Sixth International Conference on Computer Vision (IEEE Cat. No.98CH36271)*, pages 363–369.

Wen, F., Zhang, Z., He, T., and Lee, C. (2021). Ai enabled sign language recognition and vr space bidirectional communication using triboelectric smart glove. *Nature Communications*, 12:5378.

Yao, Z. and Ruzzo, W. (2006). A regression-based k nearest neighbor algorithm for gene function prediction from heterogeneous data. *BMC bioinformatics*, 7 Suppl 1:S11.

Zhou, H., gang Zhou, W., Zhou, Y., and Li, H. (2020). Spatial-temporal multi-cue network for continuous sign language recognition. In *AAAI*.

Zuo, R. and Mak, B. (2022). C2slr: Consistency-enhanced continuous sign language recognition. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5121–5130.