Sign Pose-based Transformer for Word-level Sign Language Recognition

Matyáš Boháček Marek Hrúz

University of West Bohemia, Faculty of Applied Sciences,
Department of Cybernetics and New Technologies for the Information Society
Technická 8, 301 00 Plzeň, Czech Republic
matyas.bohacek@matsworld.io mhruz@ntis.zcu.cz

Abstract

In this paper we present a system for word-level sign language recognition based on the Transformer model. We aim at a solution with low computational cost, since we see great potential in the usage of such recognition system on hand-held devices. We base the recognition on the estimation of the pose of the human body in the form of 2D landmark locations. We introduce a robust pose normalization scheme which takes the signing space in consideration and processes the hand poses in a separate local coordinate system, independent on the body pose. We show experimentally the significant impact of this normalization on the accuracy of our proposed system. We introduce several augmentations of the body pose that further improve the accuracy, including a novel sequential joint rotation augmentation. With all the systems in place, we achieve state of the art top-1 results on the WLASL and LSA64 datasets. For WLASL, we are able to successfully recognize 63.18 % of sign recordings in the 100-gloss subset, which is a relative improvement of 5 % from the prior state of the art. For the 300-gloss subset, we achieve recognition rate of 43.78 % which is a relative improvement of 3.8 %. With the LSA64 dataset, we report test recognition accuracy of 100 %.

1. Introduction

Sign Languages (SLs) are the main form of communication in Deaf communities. They are composed of manual and non-manual components through which complex semantics can be conveyed. The manual component is represented by the movement of the arms and hands, the non-manual component represents (micro-)motions such as facial expression or posture. In this work we are concerned with the automatic SL recognition (SLR) from RGB videos. There are two levels of SLR - isolated SLR (sometimes referred in literature as word-level) which classifies recordings of individual signs into glosses and continuous SLR which recognizes whole utterances. Fingerspelling is a special case of isolated SLR, when gestures are classified into individual characters of an alphabet.

Furthermore, the methods can be divided according to the form of input data. One form is a sequence of RGB or RGB-D images. Methods using this form usually perform better in terms of accuracy, but are computationally more demanding. The other form is a sequence of body poses represented by locations of skeletal joints and facial landmarks. Methods based on this representation achieve lower accuracy, but the classification models are lightweight and suitable for *e.g.* mobile devices. Making SLR able to run on such devices dramatically increases their potential in everyday use.

There are several datasets referenced in literature for the purpose of model training and evaluation. They differ in the SL that has been recorded, the size of the data, and the sensors used to capture them. The most prominent datasets include: LSA64 - A Dataset for Argentinian Sign Language [30], which uses colored gloves for trivial hand segmentation. DGS Kinect 40 [26] dataset of German SL was recorded using a depth sensor. The GSL [1] dataset of Greek SL provides both RGB and depth recordings. AUTSL [35] is a recent dataset of Turkish SL used in the ChaLearn competition [34]. MS-ASL - A Large-Scale Data Set and Benchmark for Understanding American Sign Language [37] is a collection of publicly available recordings of American SL, similar to WLASL [21] dataset. We provide an overview of the datsets in Table 1.

In this work we focus on isolated SLR based on the body pose representation. We analyze the capabilities of the Transformer model [38]. These models are relatively computationally cheap and have outstanding performance in sequence processing tasks. This makes them a perfect choice for computationally lightweight solution capable of running on modern mobile devices.

Dataset	Language	Sensor	Classes	Inst.
LSA64	Argent.	RGB	64	3,200
DGS	German	Depth	40	3,000
GSL	Greek	RGB+D	310	40,785
AUTSL	Turkish	RGB+D	226	38,336
MS-ASL	Americ.	RGB	1.000	25,000
WLASL	Americ.	RGB	2.000	21,083

Table 1. Overview of existing datasets. *D* means that a depth sensor was used, *Inst.* represents the number of recordings in the dataset.

The main contributions of this paper include:

- Constituting state of the art on the WLASL-100, WLASL-300, and LSA64 datasets when considering pose-based SLR.
- Novel normalization scheme.
- Sequential joint rotation augmentation of the body pose.
- Analysis of the pose-based vs appearance-based approaches.

2. Related Work

In the following section, we review existing approaches to isolated SLR and relevant overlaps from the general task of action recognition.

The task of isolated SLR has been approached in various manners of appearance and motion representations from the videos. Such techniques can be either expertly designed using handcrafted features, so that findings from SL linguistics can be employed. On the other hand deep learning can be used so that the features and their classification is learned by the model itself. Early works have used handcrafted features [2, 4, 8, 25], which caused the models to lack greater generalization ability. The onset of deep learning methods resulted in a boost in the overall performance and generalization of such systems. In the following text we will mainly focus on such methods.

There are two primary methods of both handcrafted and deep feature representations for SLR commonly used in recent works: either using raw RGB/RGB+Depth data or leveraging skeletal representation of the signer's pose. Some methods also combine both of these approaches, since they can complement each other [5, 16].

2.1. Visual data based methods

Initial works within the approach of utilizing raw RGB data as input representations have employed Convolutional Neural Networks (CNNs) to create holistic representations of all video frames [6, 9, 10, 17, 29, 32], which can be used for recognition. Then, recurrent neural networks such as Long short-term memory (LSTM) network [9, 17], Bidirectional LSTM network [10] or Transformers [5, 32] have been used for temporal information encoding.

3D CNNs have also been employed for this task, as they can, in addition to learning the representations of all video frames, learn spatio-temporal features as well. Tran *et al*. [15] proposed the C3D model, which was the first 3D CNN for action recognition. This was soon followed by many adaptations of 3D CNN action recognition architectures for SLR, such as the I3D [7] architecture, which was used for SLR in [21, 37].

The usage of depth cameras has also been studied in regards to this task, as the depth stream can help the models learn more complex gestures within the signing area, as well as ignore the background of the videos. Early works used ensemble models, such as conditional random fields [40] or multi-layered random forests [20], for recognition on top of such depth representations. Recently, Park *et al*. [27] proposed the SUGO model for SLR based on 3D CNN, which utilizes the LIDAR scanner and enables inference directly on modern mobile devices with suitable computational predispositions.

2.2. Pose data based methods

Extraction of the human pose from images or video recordings has proven itself expedient for the general task of action recognition. Yan *et al*. [39] were the first to propose a spatio-temporal graph convolutional network for action recognition, which was able to learn the temporal skeleton dynamics and classify them. Many architectures, such as MS-G3D [24] or AS-GCN [23], followed this approach. Nie *et al*. [25] later proposed a framework for joint pose estimation and action recognition built on top of a graphical model.

Architectures based on skeletal data have recently been applied to the specific task of SLR as well. This approach assumes that all the necessary information for recognition of a SL can be retrieved from the pose of the signer's body, hands, and optionally face. Works [13, 21] both employ the extraction of body pose sequence from individual video frames for SLR followed by graph convolutional neural network [21] or complemented by a 3D CNN [13]. Both of these works have shown that this approach can deliver comparable results to the appearance-based representation methods.

3. Dataset

We evaluate our model on two datasets: the Word Level American Sign Language (WLASL) dataset [21] and the LSA64 dataset [30]. The sign instances contained in the WLASL dataset are always performed by native American SL signers or interpreters. The data was collected from multiple public resources intended primarily for the teaching of

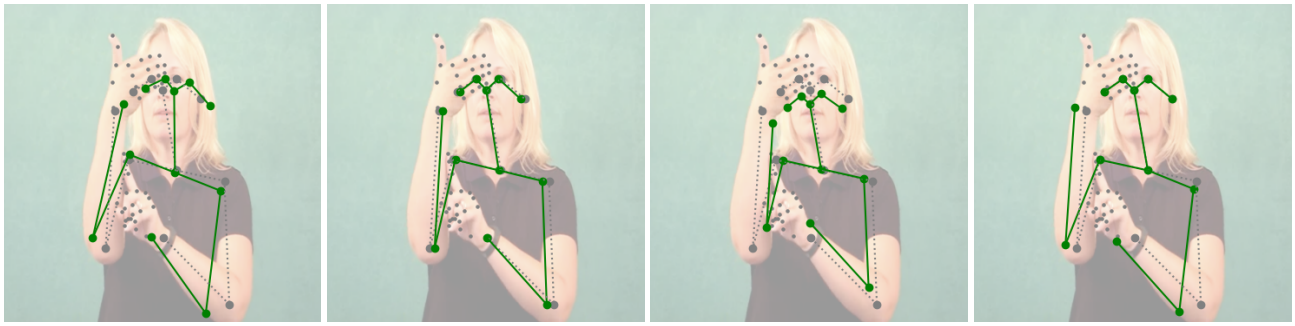

Figure 1. Depiction of individual augmentations applied on single frames. From left to right, there is in-plane rotation, squeeze, perspective transformation, and sequential joint rotation augmentation.

Subset	Gloss	Videos	Mean	Signers
WLASL100	100	2,038	20.4	97
WLASL300	300	5,117	17.1	109
WLASL1000	1,000	13,168	13.2	116
WLASL2000	2,000	21,083	10.5	119

Table 2. Overview of the WLASL dataset's subsets. Column "Mean" refers to the average number of video instances per gloss (class).

SL, thus unrestricted varieties of signing styles or dialects, as well as video backgrounds are present. The authors of the dataset created four subsets of data, named WLASL100, WLASL300, WLASL1000, and WLASL2000, each containing the respective number of glosses (classes). Detailed statistics of the subsets can be found in Table 2. We follow the public dataset split released by its authors.[1]

The LSA64 dataset holds 3200 videos of 64 different glosses from the Argentinian SL, which were selected among the most commonly used ones in the LSA lexicon and include both verbs and nouns. The instances are performed by 10 non-expert subjects.

4. Methodology

In this section, we describe the details and individual components of our data pipeline and our proposed model architecture. We call the method SPOTER - Sign POse-based TransformER, which reflects the facts that we handle the body pose according to the detected signing space, and we use a Transformer to classify the pose sequence into a sign gloss.

4.1. Preprocessing

We obtained pose (head, body, and hand landmarks) estimates from each video frame using the standard pose estimation algorithm from Vision API[2]. This could be, however, substituted by any other pose estimation framework. We generally assume that these estimations are correct and we are able provide our estimates for the purpose of reproducibility or for comparing the classification models. We extract 54 body landmarks including 5 head landmarks and 21 landmarks per hand. Excluding the head landmarks, they represent body joints. The head landmarks span the eyes, ears, and nose. The hand joints follow the standard protocol from hand pose estimation task - four joints per finger (including the fingertips) and one joint for the wrist. All the landmarks are two dimensional, hence we obtain 108 dimensional pose vector per frame.

If there was no person located in the frame or any individual landmark could not have been identified, zeros have been filled for the respective coordinate values. We leave the classifier to cope with this representation of such absence on its own.

The landmark coordinate values are relative to the frame, where the bottom left corner is represented as $[0; 0]$ and the top right as $[1; 1]$.

4.2. Augmentations

To prevent overfitting and boost the generalization capability of the model, we apply the following spatial augmentations on the skeletal data during training. The parameters of the augmentations are always randomly selected from uniform distribution, but kept consistent for all the frames within a sign instance.

In-plane rotation All the joint coordinates in each frame are rotated by a random angle θ up to 13 degrees in the following manner:

$$f_{\text{rotate}}(x, y) = ((x - 0.5)\cos\theta - (y - 0.5)\sin\theta + 0.5,$$
$$(y - 0.5)\cos\theta + (x - 0.5)\sin\theta + 0.5),$$

[1] We requested and obtained all videos, which were no longer publicly available, directly from the authors according to the provided instructions.

[2] https://developer.apple.com/documentation/vision

with the center of rotation lying in the center of the frame, which is equal to $[0.5; 0.5]$.

Squeeze All the frames are squeezed from both horizontal sides. Two different random proportions up to 15% of the original frame's width w_1 (for left side) and w_2 (for right side) are cut. The x values of the joint coordinates are then re-calculated with respect to the new plane as follows:

$$f_{\text{squeeze}}(x) = \frac{x - w_1}{W - (w_1 + w_2)},$$

where x is the original landmark's x value and W denotes the width of the frame. The y values are kept the same.

Perspective transformation The joint coordinates are projected onto a new plane with a spatially defined center of projection, which simulates recording the sign video with a slight tilt. Each time, the right or left side, as well as the proportion by which both the width and height will be reduced, are chosen randomly. This proportion is selected from a uniform distribution on the $[0, 1)$ interval. Subsequently, the new plane is delineated by reducing the width at the desired side and the respective vertical edge (height) at both of its adjacent corners.

Sequential joint rotation The joint coordinates of both arms are passed successively, and the impending landmark is slightly rotated with respect to the current one. The chance of each joint to be rotated is 3:10 and the angle of alternation is a uniform random angle up to ± 4 degrees. This simulates slight, negligible variances in each execution of a sign, which do not change its semantic meaning.

The augmentations were tested in various combinations, as described in Section 5.3. Their sample visualizations can be seen in Figure 1.

4.3. Normalization

Since the distances from the camera, tilting, and other positional properties of the signers in the recordings largely differ and the input landmark coordinates come in values relative to the frame, without normalization, the model would be learning many spatial features irrelevant to the performed sign itself. Furthermore, we presume that it would also require longer training and would not reach as generalizable results, as those potentially unlocked by normalization omitting most of the noisy properties such as the signer's body proportions, distance from the camera, or location within the frame.

Hence, our normalization method utilizes findings from the SL linguistics [3] concerning the use and delimitation of space in order to project the body landmarks onto the signing space. Its application on a single frame can be seen in Figure 2. The signing area is a three-dimensional space in front of the signer and their immediate surroundings within the reach of hands for performing SL. While it does not have a detailed, universally accepted definition, in most of the literature, it is outlined as the area from the waist to the area slightly above the signer's head, spanning transversely from elbow to elbow when both arms are kept loosely bent.

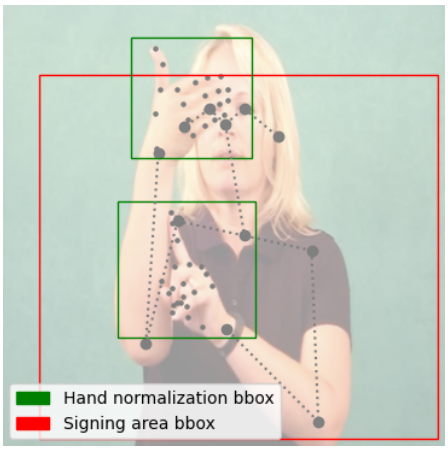

Figure 2. Visualized normalization applied to each frame of the sign instance. The bounding boxes represent individual planes whereby the corresponding landmarks are normalized, i. e. their coordinates are transferred relatively to the bounding box.

We employ the known proportions of the human body parts with respect to the height of the head [11], which are indicative and applicable regardless of the individual body structure and distance of the signer from the camera, and define the signing area on the basis of the head metric. We define one head unit as the height of the head, which we estimated by halving the distance between the signer's shoulders. We delimit the signing space as being 6 head units wide with the nose as its horizontal center, and 7 head units high. The vertical position of the bounding box is determined by the left eye, where the top edge of the bounding box is 0.5 head units upright from it and the bottom edge is 6 head units below.

The hand pose landmarks are normalized according to their own individual bounding boxes, which are found as the smallest possible upright squares encompassing all the joints of the corresponding hand while having its corners at least one-tenth of both the bounding box's width and height distant from the protruding landmarks. This enables the model to truly focus on the shape of the hand, rather than combining its spatial anchoring within the whole frame. The information of the hand location with respect to the signing space is also crucial, however, we argue that the model should be able to learn this anyway, since the body landmarks contain both wrists.

Finally, the normalized coordinates are shifted by $[-0.5; -0.5]$, so that the mean lies in 0 and their spread is 1. We have experimented with various spreads and mean values and found this one to perform the best.

Encoder Lay.	Decoder Lay.	heads	hidden dim.	feed-forward dim.	input dim.
6	6	9	108	2048	108

Table 3. Summary of the parameters of the Transformer model.

4.4. Proposed architecture

The overall architecture of our model is a lightly modified Transformer as was proposed by Vaswani *et al*. [38]. The model is depicted in Figure 3. The input of our system is a sequence of normalized body poses as described in Section 4.3. The pose is composed of 54 joint locations, yielding a 108 dimensional pose vector for each image. Next, the positional encoding is added to the individual vectors. We use a learned encoding with dimension of 108 and add it element-wise to the pose vector. This gives us the input sequence that is presented to the encoder layers of the transformer. There, the sequence flows through the self-attention module and feed forward network composed of two layers, same as in the original transformer. There are 6 encoder layers in total and 9 heads in the self-attention module.
The decoder of the transformer has one query at the input. This query is decoded into the class representing the sign and hence we call it the Class Query. The class query passes through a Multi-Head Projection module. This module is a special case of the Multi-Head Attention module, when there is only one element in the processed sequence. In this case, the softmax in the attention module always results in 1 and thus the attention has no influence on the value vector. Hence, only the projection of the input vector into the value space has any meaning and we do not learn the key and query spaces in this module. We keep several parallel projection heads as in the original multi-headed attention module. These projections are then concatenated and processed by the final linear layer of the internal hidden dimension. Next, the output of the encoder is combined with the projected class query in another Multi-Head Attention module. Again, we use 6 decoder layers and 9 heads. The decoded class query is inputted into a linear layer with number of neurons equal to the number of classes and the softmax activation is used to predict the confidences of each class. The parameters of the model are summarized in Table 3. We also experimented with the architecture of a Vision Transformer [12], but we were not able to achieve the accuracy of the proposed architecture. However, more experimenting with the size of the Vision Transformer would be required to come to conclusive results.

5. Experiments and analysis

In this section, we describe the experimental setup and provide quantitative results of our architecture with comparison to current benchmarks.

5.1. Implementation details

The proposed SPOTER architecture has been implemented in PyTorch [28]. We have edited the standard PyTorch implementation of the Transformer so as to prevent the redundant computation of the query and key pair and their successive scaled dot-product attentions in the decoder multi-head attention module, which would otherwise take place due to the passage of the Class Query through the standard multi-head attention modules. We open-source our code along with the preprocessed pose data of the WLASL and LSA64 datasets extracted with Vision API[3] for reproducibility.

We train the model for 350 epochs using an SGD optimizer with an initial learning rate of 10^{-3}. No scheduler is employed. We use the standard cross-entropy loss. The weights are initialized randomly from a uniform distribution on the interval $[0, 1)$. The momentum and weight decay were both set to 0.

The pipeline is as follows: we obtain the image from the recording, detect all relevant body landmarks, perform the augmentation, and normalization. We process every frame of the recording in this way and input it into the transformer model.

5.2. Results

We report the top-1 macro accuracy on the WLASL100 and WLASL300 dataset subsplits, as well as the LSA64 dataset. The results with comparison to existing models for the WLASL subsets are shown in Table 4. The table is divided into two main sections, where the first contains approaches based on the appearance representation, whereas the second includes only pose-based representations. The results of appearance-based approaches are included for convenience and potential comparison of development in both main data representation streams for SLR. However, we compare our model solely with the relevant category of pose-based methods. By achieving 63.18% accuracy, our proposed method surpasses the previous state of the art pose-based approach by more than 3 percentage points on the WLASL100. We establish state of the art result of 43.78% accuracy on the WLASL300 subsplit as well, surpassing the previous one by 1.6%.

Although the approaches leveraging appearance data still outperform our model, we argue that these results come at a distinctively larger computational cost originating mainly from the additional dimensionality, which is reduced in

[3]The code and data are publicly available at `https://github.com/matyasbohacek/spoter`.

Model	App.	Pose	Backbone	WLASL100	WLASL300
I3D (baseline) [21]	✓	✗	✓	65.89	56.14
TK-3D ConvNet [22]	✓	✗	✓	**77.55**	**68.75**
Fusion-3 [13]	✓	✗	✓	75.67	68.30
GCN-BERT [36]	✗	✓	✗	60.15	42.18
Pose-TGCN [21]	✗	✓	✗	55.43	38.32
Pose-GRU [21]	✗	✓	✗	46.51	33.68
SPOTER (Ours)	✗	✓	✗	**63.18**	**43.78**

Table 4. Top-1 macro average recognition accuracy achieved by each model (by row) on the WLASL100 and WLASL300 subsets. *App.* denotes the usage of appearance representation (i.e. direct frame images) as the input to the model, *Pose* column then marks the usage of skeletal data as inputs. All the entries used data augmentation techniques.

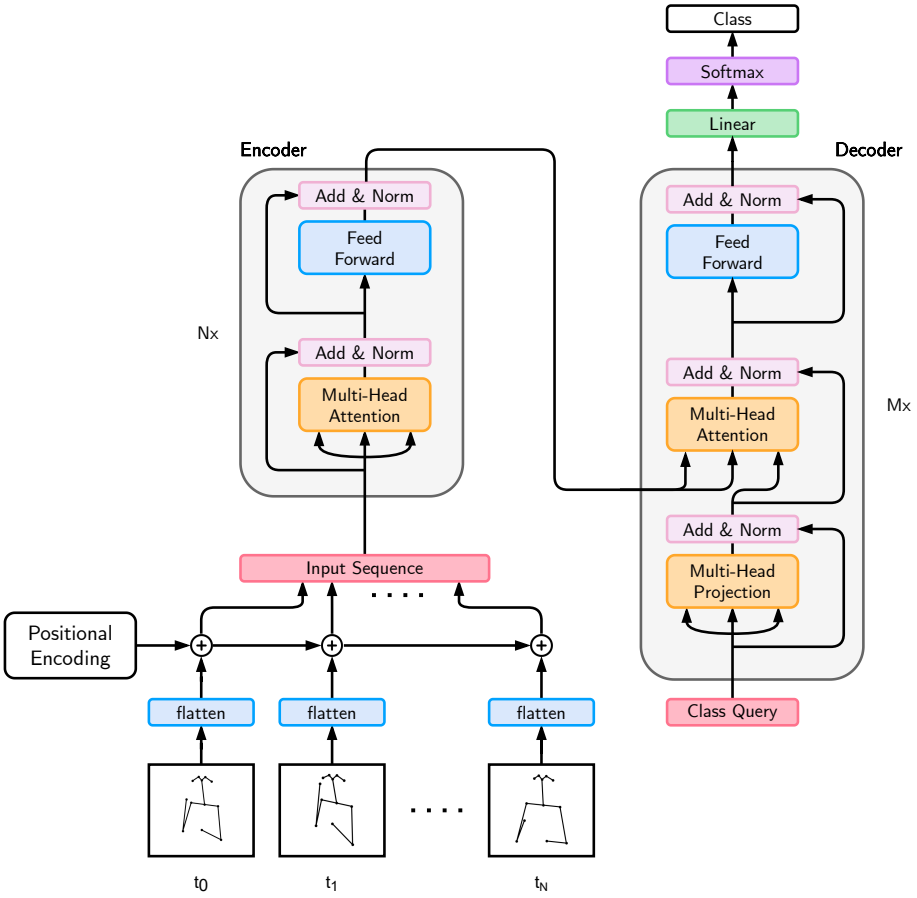

Figure 3. The proposed architecture. A standard Transformer is used with an Encoder and a Decoder part. There is however only one query - Class Query - to be decoded, which renders the multi-head self-attention in the decoder useless. That is why we call it the Mutli-Head Projection module instead, which can be implemented in a more computationally efficient way.

our system, even together with the pose estimation framework, as studied in Section 5.4. Moreover, many of the appearance-based methods require starting on the basis of a pre-trained backbone. Even with the reduced dimensionality and the backbone being absent, our model is closing on the baseline I3D model with a relatively small difference of absolute 3% in test accuracy.

The results on the LSA64 dataset are presented in Table 5. All of the reported approaches utilized the appearance representations, which were in some cases also supplemented with the pose data. We establish state of the art result by achieving 100% test accuracy.

5.3. Ablation study

The results of an ablation study investigating the effects of our normalization and data augmentation techniques are

Model	App.	Pose	Accuracy
LSTM + LDS [19]	✓	✗	98.09 ± 0.59 *
LSTM + DSC [18]	✓	✓	99.84 ± 0.19 *
DeepSign CNN [33]	✓	✗	96.00
MEMP [41]	✓	✗	99.06
ELM + MN CNN [14]	✓	✓	97.81
I3D	✓	✗	98.91
SPOTER (Ours)	✗	✓	**100.00 ± 0 ***

Table 5. Top-1 macro average recognition accuracy achieved by several models on the LSA64 dataset, as reported in respective papers. *App.* denotes the usage of appearance representation (i.e. direct frame images) as the input to the model, *Pose* column then marks the usage of skeletal data as inputs. The metric was obtained by using a single random split of 80% of the data for training and the rest 20% for testing, stratified uniformly to preserve class distributions. Entries marked with an * were obtained using cross-validation over 5 repetitions.

shown in Table 6. In order to evaluate the importance of each superstructural component, we first trained a baseline model (A) without normalization nor any augmentations applied to the data, which achieved 44.96% test accuracy on the WLASL100 subsplit.

Normalization We evaluate the importance of normalization by training the model (B) with the custom normalization approach described in Section 4.3 applied to all data. A significant increase of over 14% in accuracy can be observed, which confirms its crucialness for achieving good results. We hence fix the normalization for all subsequent configurations.

Augmentations For the following model variants (C, D, E, F), we always incorporate a single augmentation technique to assess its contribution to learning. The augmentations are executed randomly with the chance of 0.5 on the fly. Surprisingly, we find that the standard augmentations (In-plane rotation, Squeeze) provide approximately identical benefits as the augmentations specifically designed to address the SL characteristics (Perspective transformation, Arm joint rotation). Regardless, all of the augmentations boost the overall model performance by approximately $1 - 2\%$, which vindicates their benefits for the prevention of overfitting and enhanced generalization. When all of the proposed augmentations are combined, an even better result of 62.79% test accuracy is achieved, as has been demonstrated with the model (G). Once we have established the clear benefits of the outlined augmentations, we have trained the final model configuration (H) with the addition of Gaussian noise[4] applied on the training set. This has produced the best result of 63.17% accuracy.

[4]The used Gaussian noise transformation has the mean of 0 and the standard deviation of 10^{-3}.

Model	Norm.	Aug.	Accuracy
A	✗	✗	44.96
B	✓	✗	58.97
C	✓	Rotate	61.24
D	✓	Squeeze	60.85
E	✓	Perspective t.	60.47
F	✓	Arm joint rotate	61.24
G	✓	All	62.79
H	✓	All + Gaussian noise	63.18

Table 6. Ablation study results on the WLASL100 dataset split. Perspective t. denotes the perspective transformation augmentation technique described in Section 4.2.

5.4. Performance study

In order to assess the performance and efficiency properties of the pose- and appearance-based models, we perform a comparative analysis of SPOTER and the baseline I3D architecture. We started with the counts of model parameters: SPOTER has 5.92 million parameters, whereas the I3D has 12.35 million parameters, more than twice as much.

We then assessed the computational difficulty of both architectures by measuring their inference efficiencies and times. For our approach we include the Vision Pose Estimation into this measurement. We used the Deepspeed [31] library's Flops profiler to gauge the floating point operations (FLOPs) performed during inference. As the length of the input video affects the number of necessary computations for inference on the I3D model, we have randomly chosen 100 videos from the LSA64 dataset and averaged the required FLOPs, as well as the inference times for both models. The evaluation of SPOTER and I3D has been performed on a single NVIDIA Tesla T4 GPU machine, whereas the Vision API's pose estimation has been examined on a machine with the Apple M1 chip, as the library relies on macOS. Our approach greatly outperforms the I3D model. The combined inference of the Vision Pose Estimation and SPOTER required by average 1.42 GFLOPs and took 0.05 second, while the I3D required 5.22 GFLOPs and took 0.55 second. All of these performance attributes are depicted relatively in Figure 4.

We also tested the ability of both models to learn and generalize from only smaller training sets by sampling the sizes of the training dataset and comparing the resulting models against a constant test set. We chose the smaller LSA64 for this experiment.

We fixed the seeds and split a 20% set for testing. Next, we trained 10 SPOTER and I3D models, each time with a different proportion of the training set, starting with a $10\ \%$ subset all the way to the full set, adding $10\ \%$ each time. All the sets were stratified uniformly to preserve class distributions. Each model has then been evaluated on the original testing set. We followed the baseline I3D implemen-

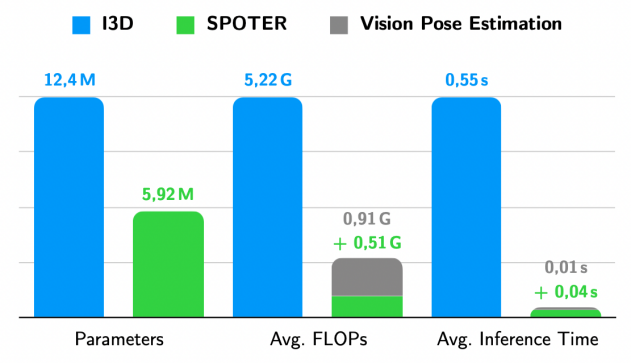

Figure 4. Relative comparison of the SPOTER and I3D model attributes: model parameters, average inference FLOPs, and average inference time. All of the metrics were measured and later averaged on 100 random videos from the LSA64 dataset.

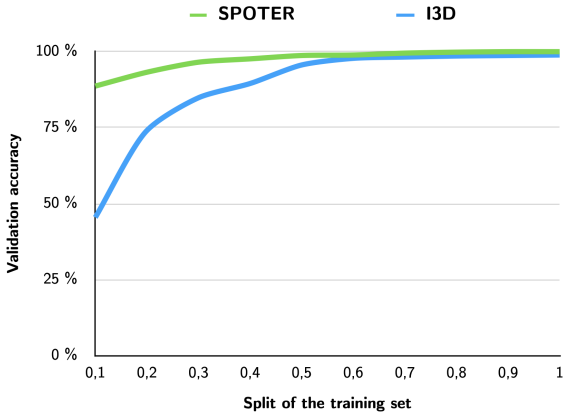

Figure 5. Top-1 macro accuracies of the SPOTER and I3D models trained on 10 gradually enlarging portions of the training set. A fixed 20% split of the LSA64 dataset is used for evaluation, while the rest is used for training. All the seeds were fixed beforehand.

tation from [21] and their accompanying repository, where we only doubled the original frame cut-off threshold to 128 to accommodate the characteristics of the LSA64 dataset.

The results are shown in Figure 5. Even the smallest SPOTER model trained on just 10% split was able to achieve an accuracy of 88.68%, while the I3D model lagged behind with just 45.47%. SPOTER continued to improve until it eventually reached the accuracy of 100.00% at the 90% train set split. The I3D improved as the training split grew, however, it did not manage to catch up with SPOTER at any split. It came the nearest at the final split, where it achieved the testing accuracy of 98.91%.

We attribute this behavior to the necessity for the I3D model to first learn general concepts required for SL semantic decoding (such as human body mechanics), which is harder on a small training set. The SPOTER architecture, on the contrary, does not need to acquire such understanding, as the handcrafted input feature representation of body and hand pose already contains sufficient information for such decoding, and thus requires a substantially smaller training set to gain ample results.

Both of the previous experiments demonstrate the SPOTER's dominance on small instance datasets. Combined with the model size, inference computational demands, and speed, SPOTER proves much more suitable for applications in the wild as opposed to appearance-based approaches, such as I3D.

6. Conclusion

In this paper, we explore the application of Transformer in the task of isolated SLR. Previous works tackling this problem have frequently used a computationally heavy approach to obtain sensible results or relied on pre-trained backbones, which we are preventing by using handcrafted pose feature representations and hence reducing the dimensionality. Furthermore, previous systems could not fully utilize any normalization or augmentations beyond the standard ones applied to visual data. We propose a novel approach of utilizing Transformer for this task. Since our model operates on top of body pose sequence representations, we apply knowledge from SL linguistics to create a robust normalization technique as well as new data augmentation techniques specific for the SL.

We validated our approach on two datasets for isolated SLR. We achieved overall state of the art results for the LSA64 and established state of the art results in the pose-based model category for the WLASL. We have also performed a performance study comparing our model to the I3D baseline, which proved that the newly proposed architecture is substantially less demanding and generalizes well even on very small training sets.

Apart from the obvious reason of computational efficiency, the SPOTER model could be utilized in a Human in the Loop fashion. A relatively small portion of human labeled data could be used to train the model to predict relatively precise labels to be corrected by the annotators. Then these data could be used to train a appearance-based model if needed.

Acknowledgement

The work described herein has been supported by the Ministry of Education, Youth and Sports of the Czech Republic, Project No. LM2018101 LINDAT/CLARIAH-CZ. Computational resources were supplied by the project "e-Infrastruktura CZ" (e-INFRA CZ LM2018140) supported by the Ministry of Education, Youth and Sports of the Czech Republic.

References

[1] Nikolas Adaloglou, Theocharis Chatzis, Ilias Papastratis, Andreas Stergioulas, Georgios Th Papadopoulos, Vassia Zacharopoulou, George J Xydopoulos, Klimnis Atzakas, Dimitris Papazachariou, and Petros Daras. A comprehensive study on sign language recognition methods. *arXiv preprint arXiv:2007.12530*, 2020.

[2] Purva C. Badhe and Vaishali Kulkarni. Indian sign language translator using gesture recognition algorithm. In *2015 IEEE International Conference on Computer Graphics, Vision and Information Security (CGVIS)*, pages 195–200, 2015.

[3] Anastasia Bauer. *The Use of Signing Space in a Shared Sign Language of Australia*. De Gruyter, 1 edition, 2014.

[4] Patrick Buehler, Andrew Zisserman, and Mark Everingham. Learning sign language by watching tv (using weakly aligned subtitles). In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 2961–2968, 2009.

[5] Necati Cihan Camgoz, Oscar Koller, Simon Hadfield, and Richard Bowden. Multi-channel transformers for multi-articulatory sign language translation. In *European Conference on Computer Vision*, pages 301–319. Springer, 2020.

[6] Necati Cihan Camgoz, Oscar Koller, Simon Hadfield, and Richard Bowden. Sign language transformers: Joint end-to-end sign language recognition and translation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10023–10033, 2020.

[7] Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6299–6308, 2017.

[8] Helen Cooper, Eng-Jon Ong, Nicolas Pugeault, and Richard Bowden. Sign language recognition using sub-units. *J. Mach. Learn. Res.*, 13(1):2205–2231, July 2012.

[9] Runpeng Cui, Hu Liu, and Changshui Zhang. 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, 2017.

[10] Runpeng Cui, Hu Liu, and Changshui Zhang. A deep neural framework for continuous sign language recognition by iterative training. *IEEE Transactions on Multimedia*, 21(7):1880–1891, 2019.

[11] Liyanage De Silva. Audiovisual sensing of human movements for home-care and security in a smart environment. *International Journal On Smart Sensing and Intelligent Systems*, 1, 01 2008.

[12] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2020.

[13] Al Amin Hosain, Panneer Selvam Santhalingam, Parth Pathak, Huzefa Rangwala, and Jana Kosecka. Hand pose guided 3d pooling for word-level sign language recognition. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 3429–3439, 2021.

[14] Javed Imran and Balasubramanian Raman. Deep motion templates and extreme learning machine for sign language recognition. *The Visual Computer*, 36(6):1233–1246, 2020.

[15] Y Jia, E Shelhamer, J Donahue, S Karayev, J Long, R Girshick, S Guadarrama, T Darrell, UCB Eecs, A Karpathy, et al. C3d: Generic features for video analysis. In *2014 IEEE Conf. Comput. Vis. Pattern Recognit*, pages 675–678, 2014.

[16] Songyao Jiang, Bin Sun, Lichen Wang, Yue Bai, Kunpeng Li, and Yun Fu. Skeleton aware multi-modal sign language recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pages 3413–3423, June 2021.

[17] Oscar Koller, Necati Cihan Camgoz, Hermann Ney, and Richard Bowden. 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, 2020.

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

[19] Dimitrios Konstantinidis, Kosmas Dimitropoulos, and Petros Daras. Sign language recognition based on hand and body skeletal data. In *2018-3DTV-Conference: The True Vision-Capture, Transmission and Display of 3D Video (3DTV-CON)*, pages 1–4. IEEE, 2018.

[20] Alina Kuznetsova, Laura Leal-Taixé, and Bodo Rosenhahn. Real-time sign language recognition using a consumer depth camera. In *2013 IEEE International Conference on Computer Vision Workshops*, pages 83–90, 2013.

[21] Dongxu Li, Cristian Rodriguez, Xin Yu, and Hongdong Li. Word-level deep sign language recognition from video: A new large-scale dataset and methods comparison. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pages 1459–1469, 2020.

[22] Dongxu Li, Xin Yu, Chenchen Xu, Lars Petersson, and Hongdong Li. Transferring Cross-Domain Knowledge for Video Sign Language Recognition. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 6204–6213, 2020.

[23] Maosen Li, Siheng Chen, Xu Chen, Ya Zhang, Yanfeng Wang, and Qi Tian. Actional-structural graph convolutional networks for skeleton-based action recognition. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3595–3603, 2019.

[24] Ziyu Liu, Hongwen Zhang, Zhenghao Chen, Zhiyong Wang, and Wanli Ouyang. Disentangling and unifying graph convolutions for skeleton-based action recognition. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 143–152, 2020.

[25] Bruce Xiaohan Nie, Caiming Xiong, and Song-Chun Zhu. Joint action recognition and pose estimation from video. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1293–1301, 2015.

[26] Eng-Jon Ong, Helen Cooper, Nicolas Pugeault, and Richard Bowden. Sign language recognition using sequential pattern

trees. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Providence, Rhode Island, USA, June 16 – 21 2012.

[27] HyeonJung Park, Youngki Lee, and JeongGil Ko. Enabling real-time sign language translation on mobile platforms with on-board depth cameras. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.*, 5(2), June 2021.

[28] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019.

[29] Lionel Pigou, M. V. Herreweghe, and J. Dambre. Sign classification in sign language corpora with deep neural networks. In *LREC 2016*, 2016.

[30] Facundo Quiroga, Franco Ronchetti, César Armando Estrebou, Laura Cristina Lanzarini, and Alejandro Rosete. Lsa64: An argentinian sign language dataset. In *XXII Congreso Argentino de Ciencias de la Computación (CACIC 2016)*, pages 794–803.

[31] Jeff Rasley, Samyam Rajbhandari, Olatunji Ruwase, and Yuxiong He. Deepspeed: System optimizations enable training deep learning models with over 100 billion parameters. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 3505–3506, 2020.

[32] Ben Saunders, Necati Cihan Camgoz, and Richard Bowden. Continuous 3d multi-channel sign language production via progressive transformers and mixture density networks. *International Journal of Computer Vision*, pages 1–23, 2021.

[33] Jai Amrish Shah et al. *Deepsign: A deep-learning architecture for sign language*. PhD thesis, 2018.

[34] Ozge Mercanoglu Sincan, Julio Junior, CS Jacques, Sergio Escalera, and Hacer Yalim Keles. Chalearn lap large scale signer independent isolated sign language recognition challenge: Design, results and future research. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3472–3481, 2021.

[35] Ozge Mercanoglu Sincan and Hacer Yalim Keles. Autsl: A large scale multi-modal turkish sign language dataset and baseline methods. *IEEE Access*, 8:181340–181355, 2020.

[36] Anirudh Tunga, Sai Vidyaranya Nuthalapati, and Juan Wachs. Pose-based Sign Language Recognition using GCN and BERT. *Proceedings - 2021 IEEE Winter Conference on Applications of Computer Vision Workshops, WACVW 2021*, pages 31–40, 2021.

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

[38] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017.

[39] Sijie Yan, Yuanjun Xiong, and Dahua Lin. Spatial temporal graph convolutional networks for skeleton-based action recognition. In *Thirty-second AAAI conference on artificial intelligence*, 2018.

[40] Hee-Deok Yang. Sign language recognition with the kinect sensor based on conditional random fields. *Sensors*, 15(1):135–147, 2015.

[41] Xinyu Zhang and Xiaoqiang Li. Dynamic gesture recognition based on memp network. *Future Internet*, 11(4):91, 2019.