Sign Language Segmentation Using a Transformer-based Approach

Luis F. Pérez-Villegas,
Artificial Intelligence Dept.
Computer Science School, UNED.
Madrid, Spain
lperez1478@alumno.uned.es

Sonia M. Valladares Rodríguez,
Artificial Intelligence Dept.
Computer Science School, UNED.
Madrid, Spain
soniavr@dia.uned.es

Olga C. Santos
Artificial Intelligence Dept.
Computer Science School, UNED.
Madrid, Spain
ocsantos@dia.uned.es

Abstract

Continuous Sign Language Recognition (CSLR), predicting the meaning of the signs in sign language sentences, is one of the current challenges in translation between sign and spoken languages, that would benefit people with hearing impairment. An important limitation of this research field is the lack of annotated datasets, which could be minimized with Sign Segmentation approaches by automating the costly task of manually annotating the beginning and ending of each sign. The goal of this paper is to study the performance of an architecture which combines I3D CNN extracted features with a transformer-based model called ASFormer which was created specifically for Action Segmentation task. In our approach ASFormer, instead of separating actions in motions is separating signs in a signed speech. Several ablation studies are performed, and it is shown that ASFormer is suitable for segmenting the signs, with a performance near the ones of the state-of-the-art models, confirming the promising benefits of using attention-based approaches in this field.

1. Introduction

Approximately 430 million of people, up to 5 % of world population, possess some kind of deafness or auditive impairment [1]. It is of great importance to provide them with tools and resources for outcoming their communication difficulties. A significant part of the deaf community uses some type of sign languages, which are completely independent from the spoken and written language. Due to intelligibility between languages, even between sign languages, automatic translation, or more precisely, interpretation, is needed.

Most of the studies related to sign language focus on sign language recognition, where the lexical meaning of the signs is identified in units called glosses. However, most studies focus on the recognition of isolated signs (ISLR) [2] from little datasets extracted in controlled conditions, like videos with good illumination and background. In the past years, however, an increased interest has appeared [3] in the more complete problem of continuous sign language recognition (CSLR), where signs are recognized inside continuous sentences. Several problems arise in this kind of scenario, like the need of huge, annotated datasets of signs and their meanings for every sign language. This lack on annotated datasets is due to the need of highly prepared professionals, not only for capturing the videos in enough variety of situations, but also experts that can understand the meaning of the signs and could precisely annotate their boundaries, and because it is a high time-consuming task. In addition to that, there could be discrepancies between annotators that make automatic recognition task even harder.

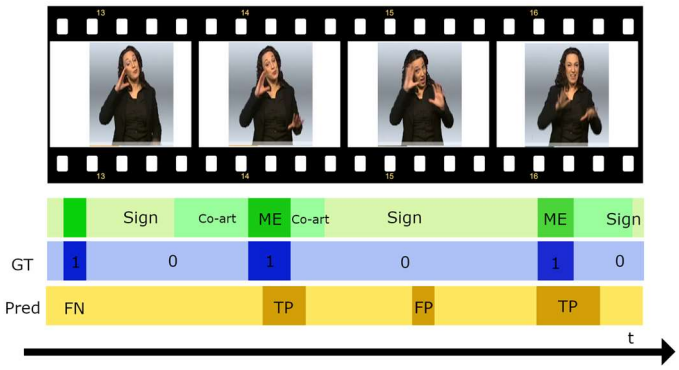


Figure 1: The purpose of temporal sign segmentation is to find the boundaries between signs. Most of the times these boundaries coincide with the movement epenthesis between signs. This problem could be modeled as a classification task where frames labeled with 1 are the boundaries, and the frames labeled with 0 are the signs. Predictions are considered correct when there is several consecutive

For these reasons, it is of big interest to automate the annotation of sign language datasets. This has made the arise of some subproblems like sign spotting, the automate search of identical signs without knowing the meaning or Sign Segmentation, where videos are split into the signs without knowing their meaning. This last subproblem have some issues associated, like the existence of co-articulation (where same signs adopt different shapes depending on the preceding and posterior signs) or differentiating signs from movement epenthesis (which is a type of co-articulation that consists in the link between signs without any

associated meaning). The detection of these two situations could not only be seen as a research challenge, but as a way for finding the boundaries of the signs.

1s near a boundary labeled in the ground truth.

Recently, a new architecture has been proposed [4] for sign segmentation which combines I3D extracted features [5] and a MS-TCN architecture [6], both modules already used in the more general Action Segmentation problem, where user actions, instead of signs, are recognized and split. This solution improved results from previous works in two of the most important sign language datasets. Inspired in this, the objective of this paper is trying to improve the results already reached in the literature and study the viability of using a new transformer-based architecture called ASFormer [7] which has proved to perform better than MS-TCN in the Action Segmentation task.

Thus, our **hypothesis** is that the ASFormer architecture can perform better segmentation of the signs and improve the results obtained in the MS-TCN architecture.

In consequence, the following research question has been posed in this paper: **Is the ASFormer a suitable solution for Sign Segmentation task improving previous state of the art results**?

The following objectives are explored: (1) It is demonstrated the effectiveness of using a transformer-based approach to produce sign boundary predictions, (2) the optimal configuration of such architecture is studied, and (3) results are compared with other state of the art models for two datasets, which are BSLCorpus [8], [9], [10] and RWTH-Phoenix-Weather14 [11][12]

The rest of sections of this article are a study of the literature on the subject (Section 2), a description of the Methodology employed (Section 3), Results of Experimentation (Section 4), Discussion (Section 5) and Conclusions (Section 6).

2. 2. Related work

This section is divided as followed. First, the systematic review of the state of the art that was performed this paper is described. Then the results are presented divided into the ones focusing on the detection of movement epenthesis and co-articulation and the ones directly focused on sign segmentation. After that, the review of some articles used in this work related to the more general Action Segmentation task is included. Finally, a review of annotated datasets prepared for Continuous Sign Language Recognition is done, followed by a summary with the most useful articles related with the final solution proposed in next section of this paper.

2.1. Systematic Review

A systematic review study has been performed for reviewing the literature related to the segmentation task and related termina. For that reason, we have consulted two databases: Scopus and Web of Science. A summary of the results given by each query could be seen in
Table 1.

From the 175 articles found, 46 were duplicated, 41 were excluded after reading the abstract and 49 were removed for several reasons like unavailability (2), being published in languages other than English or French (3), not being related to sign language (6), offering solutions not related to computer vision (6), etc. A full summary could be seen in the flowchart of decisions of Figure 2. At the end, 39 articles where selected and a more deep analysis was performed, based on the rules suggested in PRISMA [13][14] which will be presented as a separate systematic review paper. However, the most important conclusions are presented in the following sections.

Database	Query	Results	Repeated	Excluded	Selected
Scopus	ALL("sign language" AND "computer vision") AND TITLE(sign OR label**** OR "pseudo*label****" OR unsupervised OR epenthesis OR co-articulation OR coarticulation OR clustering) AND TITLE-ABS-KEY("pseudo label****" OR unsupervised OR epenthesis OR co-articulation OR coarticulation)	103		72	31
WoS	TI=(sign OR label**** OR "pseudo*label****" OR unsupervised OR epenthesis OR co-articulation OR coarticulation OR clustering) AND TS=("pseudo label****" OR unsupervised OR epenthesis OR co-articulation OR coarticulation) AND TS=(sign language) AND TS=(computer vision)	27	23	3	1
Scopus	TITLE-ABS-KEY ("sign language" AND ("sign segmentation" OR "sign language segmentation"))	26	5	14	7
WoS	TS=("sign language" AND ("sign segmentation" OR "sign language segmentation"))	19	18	1	0
Total		**175**	**46**	**90**	**39**

Table 1 Search queries performed in Scopus and Web of Science search engines. The first two queries are related with the detection of movement epenthesis, coarticulation and the use of pseudo labels. The final two queries are focused on the sign segmentation task.

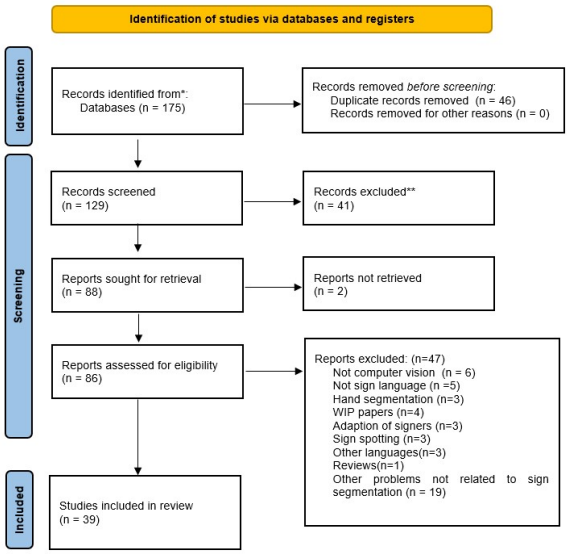


Figure 2: Research flow for including the datasets in the review of the state of the art.

2.2. Epenthesis and Co-articulation studies

Epenthesis and co-articulation are important concepts in sign language and very interesting for the sign segmentation perspective because they could be used as boundaries of the signs. In several works, however, epenthesis and co-articulation concepts are mixed. As defined in [15] coarticulation would be "the modification of signs when using them in utterances compared to when using them isolated" [15]. Meanwhile, in [16] it describes coarticulation as the process of joining two lexically different glosses in a sentence, and could be divided into **hold deletion**, in which the transition between signs is smooth, with minimum length of inter-sign pauses posing a challenge for finding the boundary between both signs, **metathesis**, in which "the point of articulation of a sign is affected by the next one in a sequence", **assimilation**, in which a sign takes the characteristics of the next one and **move epenthesis,** which are transitions between signs without any lexical information. Based on this, move epenthesis is a type of coarticulation formed when the hands move from the ending location of one sign to the starting location of the next one. Knowing this, in our literature research some studies were find that use coarticulation term to refer only to Movement Epenthesis [17].

Several studies tried to automatically detect ME or coarticulation between signs. Some of them are based on the study of the changes in velocity [18][19] , acceleration[20] or trajectories [17][21].

As per **speed** changes, in [18], following the hypothesis that transition movements were faster than sign movements, authors detected and removed the co-articulated strokes when their speed was about 3/2 above the average speed. Another study [19] concluded that transitions were delimited by two local minima in the norm of the velocity, and created a semi-automatic segmentation by detecting the speed minima of the hand motions. Due to a high number of false positives, this process was only used for improving manual annotations though. Another variant from this type of detection, but a slightly different type of feature is the one of [22] where epenthesis was detected by using the **motion vectors** partially decoded in H.264/AVC compressed videos. Frames with low-speed hand-motions are considered signs while frames with high speed are considered ME.

Another study [20] detected co-articulation of fingerspelling alphabets by finding peaks in **acceleration** between signs because stable movement was inferred as part of dynamic signs.

Several studies study the influx of the **trajectories** in detecting ME. For example, in [17] authors detect changes on trajectories, with a pair of gloves and magnetic trackers[1] to segment "phonemes". Also, in [21] differentiates movement epenthesis from trajectory signs (which were not part of any sign language though) using the trajectory phase as a feature. First, it segmented the sign using the height of the hand trajectory, and then used a feature set (two spatial feature, pairwise geometric histogram (PGH) and two temporal ones, height and orientation of the rectangle formed by the trajectory formed of p number of points at a time) for recognizing the segmented signs.

Some studies like Enhanced Dynamic Programming (EDP) [23] and 3-SU[24] are based in **subunits sign modelling** frameworks where signs are divided into components. While in [23] Parallel Hidden Markov Model (PaHMM) and 5 subunits are used, in [24], Bayesian parallel Hidden Markov Model (BPaHMM) is used an only 3 subunits (hand shape, velocity and position). These subunits are feed into the modelling phase. By using dynamic time warping DTW and minimum entropy clustering (MEC), ME are removed from the training

In [25] authors use an **optical flow based** approach to distinguish between signs and ME. For each frame, the optical flow is calculated with the Horn-Schunck method and the 2-norm of its magnitude parameter. Values above a threshold are considered movement epenthesis, and values below are considered signs.

In [26], [27] and [28] the concept of "**signemes**" is used, which are the part of the signs after removing the co-articulation. In [26] the edge pixels of the skin-colored

[1] It was decided to include this study even that it is not part of computer vision solution, because of the interesting approach based on trajectory detection that could be applied using computer vision as well.

patches are transformed as points in the Space of Relational Distribution (SoRD), then, the signemes are built by taking the parts of the sentences with the same sign that has highest resemblance score and taking the mean model of all them. In [27] the signemes are instead extracted by using a Bayesian framework. Possible starting points and width of signs are collected building conditional density functions, based on dynamic time warping (DTW) distances, and running several times Iterative Conditional Modes (ICM) to extract the signemes. HMM is used to correct DTW when the result of both models is very different. Also, in [28] **signemes** are extracted by aligning the videos with the speech cues and studying different techniques: HMM, coupled HMM and parallel HMM, concluding that the best one is the fusion of DTW and HMM)

2.3. Sign Segmentation studies

After looking at studies that only focused in detecting and or removing move epenthesis, we focus on studies that segment the signs, even when some of them use move epenthesis or coarticulation to find the boundaries. Sign Segmentation solutions could be classified [29] as mainly based on three approaches: (1) based on velocity- ,(2) template-matching-based features or (3) modelling of signs and ME.

Methods that are based on **velocity** are the most numerous. For example, in [29] a method is proposed for distinguish ME from signs by using adaptative thresholds on three features, shape (Zernike moments), velocity change (Lucas-Kanade optical flow method) and displacement of centroid (geometric moments). If two of these variables are above the thresholds it is considered as ME.

In [30] authors create a semi-automatic annotation system that analyses hand shapes, hand speed and face landmarks to annotate features and segment the signs. After that, the signs are classified into lexical (with a defined meaning), and iconic (illustrative) types based on a probabilistic model. In more detail, the temporal segmentation is done by using hand speed and taking into consideration that each sign begins with a maximum peak on speed and ends with a minimum. To avoid confusion with signs that perform repetitive movements, there is a scan that merges segmented signs with similar hand shapes. Authors argument that most of the time signs remain stable, even if there are some signs that change hand shape. Another contribution of this paper is the introduction of task source-free domain adaptation where the source data is available only in the initial training phase but not in the adaptation phase (for example, avoiding problems of data privacy)

In [31] hand shape trajectories are segmented and candidate boundaries are obtained based on minimal velocity and maximal change of direction. These points are filtered out by a set of rules. PCA of the final segments are extracted and clustered by k-means to derive “phonemes” or basic part of signs

In [32] segmentation of signs is achieved by computing the standard deviation extracted from the variation in waves of a RF signal that captures the movements of the hands.

In [33] motion features are extracted and the relative velocity between hands is compared to identify symmetric movements and static signs identifying possible Boundaries between signs. After that, hand shape is extracted to correct the limits that do not correspond to a hand shape changing. In [34] the segmentation proposed in previous article is used for suggesting annotations in a semi-automatic way, using Zebedee, a system for codifying and classify signs, where a sign is considered a set of dynamic geometric constraints applied to a skeleton, and a descending classification method.

From the second group, where **template matching features are used**, authors present a model [35] that uses Level building approach for segmenting and recognizing signs. They use DTW approach and enhancing it for considering Movement Epenthesis without modelling it explicitly, i.e., Frames are be labeled as ME, if not good sign matching is found. This same approach is followed by [36], [37] and[38] where DTW is replaced with a Hidden Markov Model (HMM)

As for the model based approaches, authors in [39] propose a system of system recognition using phonemes instead of signs and train the model using Hidden Markov Models HMM. In this approach, movement epenthesis is considered at the same level as the recognized lexical signs.

In [40] authors use GLATA, a Greedy cLustering Algorithm along the Time Axis, to segment signs based in k clustering. For each frame, the distance to only two clustering centers is computed. The algorithm optimizes iteratively the sum of the distances of all frames. A modification of this algorithm is used for training a recognition system and remove epenthesis as well. Then, the classification of sign and ME frames is performed by means of an FSM model.

In [41] Conditional random fields (CRF) is used for detecting the coarticulation points and use them to segment continuous sentences

Some solutions use subtitles [42] for finding the boundaries and align the sign with its meaning.

In [43] a random forest is proposed combined with geometric features computed from 3D body joint positions as input features.

In [4] it is proposed to use I3D for feature extraction and MS-TCN for the sign segmentation phase. Authors improve the model by adding Changepoint-Modulated Pseudo-Labelling (CMPL) [44]. This algorithm applies a pseudo-labelling technique an adds changepoint detection for avoiding bias towards under-segmentation. With pseudo-labelling a classifier is retrained on its own

predictions on unlabeled data for improving performance. The purpose of changepoint detection is to locate state changes, this increases sensitivity to abrupt changes in feature space and helps avoiding under segmentation.
Finally, there are several articles that while not being focused specifically on Sign Segmentation, but on the more general sign recognition problem, they introduce modifications to consider non signs.
For example, several authors use pseudo-labelling [45], [46], [47] and [48], for improving the results of a sign classifier by modifying the boundaries of the "labeled" signs. First, videos are divided into clips of 8 frames (4 of them overlapping adjacent clips) and features are extracted via a deep learning model. (18-layer 3D ResNet with dilated Convolution [45], ResNet-18 and ResNet-3D fused with ConvolutionPyramid (TCP) and Multi-Layer Perceptron (MLP) [46], 3D-ResNet [47] and I3D (3D-CNN inflated from Inception-v1 [48]) Pseudo-labels are generated then as a process of sequence learning using Connectionist Temporal Classification which introduces a "blank" label that represent no signs. These pseudo-labels ("pseudo" because they are generated by the model) are then feed iteratively into the feature extractor to improve the results.
There are also articles that doesn't segment signs, but parts for them, like in [49], where a framework called 2 S-U is introduced where Statistical subunits are built, which are primitives for building the signs. For building these statistical subunits, sentences are divided into dynamic and static training a 2-state ergodic HMM model to separate movements from non-movements. Dynamic subunits are modelled with the direction feature vector and clustering employing DTW. Static subunits, which are low velocity SUs, are clustered based on the position feature vector.

2.4. Action Segmentation studies

Sign segmentation is a problem that could be included inside the more general Action segmentation where the objective is to recognize different actions performed in a video by one or several actors. Review its state of the art give us good insights of the possible methods we could use in Sign Segmentation.
In [6] it uses Multi-Stage Temporal Convolutional Network (MS-TCN) which is an architecture built on several stages, each one performing a dilated temporal convolution. Results on the Action Segmentation benchmark were later improved by a solution based on transformers called ASFormer [7].
Special mention to the work published in the article Action Segment Refinement Framework (ASRF) [50] where limits between actions where considered as boundaries, calculated with a regression approach and used to refine the results obtained in the normal Action Segmentation part of the framework.

2.5. Datasets for Continuous Sign Language Recognition

In this section, a non-exhaustive review of the most important datasets used in Continuous Sign Language Recognition is presented. As the particularities of Sign Language Segmentation task aligns with the Recognition one, we could use them for this task.
Normally, most of datasets have been created for isolation sign language recognition (ISLR) but Continuous Sign Language Recognition (CSLR) needs bigger datasets with a sufficient variety of sign vocabulary, signers, and enough repetitions of each sign. In Table 2, the most important datasets for this task are shown. In addition, the existence of several different signs languages makes important to have at least one proper dataset for each language. This limitation, however, is not so important for Sign Language Segmentation, as the structure of sign languages is very similar, but it is reflected in the review for completeness.
The most used benchmark found in the literature[3] is the family of RWTH-PHOENIX-Weather in their versions of 2012 [51][52], 2014 [12] and 2014T [53] which is specially prepared for the problem of Sign Language Translation which not only recognizes signs but focus on the translation of sentences. This family of datasets was the

Dataset	Language	# Signers	# Signs	Vocabulary	Sentences	# Hours	Year	Public
[51] RWTH-PHOENIX-Weather (2012)	DGS	7	21822	911	1980	3.25	2012	Yes
[12] **RWTH-PHOENIX-Weather14 (2014)**	DGS	9	65277	1558	6861	10.73	2014	Yes
[53] RWTH-PHOENIX-Weather14T (2018)	DGS	9	76000	1066	7096	11	2018	Yes
[10] **BSLCorpus**	BSL	249	72K	5000	125	125	2013	Yes
[84] S-pot	FinSL	5	1211	6000	4k	9	2014	Yes
[54] KETI	KSL	14	15k	524	105	28	2019	-
[85] Greek SL Lemmas	GSL	7	41k	310	10k	10	2020	Yes
[55] BSL-1K	BSL	40	273K	1064	1M	1060	2020	No
[56] CSL-Daily	CSL	10	151k	2000	21k	23	2021	Yes
[57] BOBSL	BSL	39	452k	2281	1.2M	1467	2021	Yes
[58] How2Sign	ASL	11	-	16k	35000	79	2021	Yes

Table 2: Review of the most important datasets used for Continuous Sign Language Recognition. The datasets highlighted are the ones used in this paper.

first benchmark focused specially on CSLR and were directly responsible in the increase of articles in the subject. Even though, with a vocabulary of only 1558 signs, RWTH-PHOENIX-Weather14 was not fully suitable for complete Continuous Sign Language Recognition. This lack of vocabulary and examples were minimized by the fact that this dataset only focusses on a specific domain, the weather newscast videos from the German television. This approach is taken by other datasets like KETI [54], where the domain is composed only from emergency-related sentences in Korean Sign Language.

Other datasets focused in recording not only individual signing interpretation, but actual conversations between signers, like BSLCorpus [10]. This dataset, alongside with RWTH-PHOENIX-Weather, and BSL-1K[55] were used as benchmarks of the sign segmentation problem for the work of [4] and [44] and is the one used in this paper for comparing our results. BSL-1K dataset is not publicly available though so it is not used in this research.

Recently there have been three datasets that have increased significatively the corpus of signs: CSL-Daily[56] a benchmark for Chinese Sign Language, BOBSL [57], the biggest one, which is the public extension of BSL-1K [55] formed both of them in British Sign Language from the BBC newscast, and How2Sign[58], which contains sentences in American Sign Language and it is only annotated at sentence level, not specifying the specific glosses of each sign, what makes it not suitable for sign segmentation.

These datasets must be correctly annotated to be used in sign segmentation models. This is carried mostly manually, by using tools such as ANVIL[59] and ELAN [60]. Annotation on such huge datasets is very time consuming, and need the collaboration between several experts in sign language[2] so it is of great importance to follow annotation guidelines like the one defined by the team responsible of BSL Corpus[61].

Additionally, there are some works that have tried to automate the annotation task outside of the automatic sign segmentation effort. For example the authors of [62] demonstrate that a crowdfunding effort of non-experts in sign language could reach around 93% of accuracy when segmenting the signs. In another study [42], it was shown a way to recognize signs with subtitles. For annotating BOBSL [57], several techniques were applied like sign spotting using mouthings [55], the movements of the mouth that are part of several signs, dictionaries [63]and using weakly-aligned subtitles as well for localizing the signs[64].

These new datasets are not free of limitations, though, one of the most important is the lack of examples of partly-lexical[65] and non-lexical units, as sign languages have special signs that represent object position or are actually more like “theatrical plays” without lexical meaning or grammatical structure that are used to illustrate situations, stories, or emotions. Datasets with examples of these kind of signing are needed, as this would prove the next big challenge in Sign Language Translation Problem. Following this need, LSE-UVIGO[66], a dataset that takes into account some of this non-lexical signs for Spanish Sign Language, is being prepared.

2.6. Summary

As seen in this section, several solutions have been proposed for segmenting signs and / or detecting movement epenthesis and other co-articulation movements, which could be used as boundaries in sign segmentation. One of the latest proposed [4], uses I3D [5] for extracting the features from video frames, and MS-TCN[6] for learning to segment the signs. The same team improved their results in [44] with a pseudo-labelling method, establishing the actual state of the art results for Sign Segmentation task.

Also, a brief review of Action Segmentation techniques was shown, highlighting ASFormer [7], a transformer-based architecture that improved the results reached by using MS-TCN in that particular problem.

Finally, some of the most used datasets for continuous sign language recognition where reviewed along their limitations, RWTH-Phoenix-Weather14 [12] and BSLCorpus [10] being two of them. These two datasets have been used recently as a benchmark for sign language segmentation and are described in more detail in the next section.

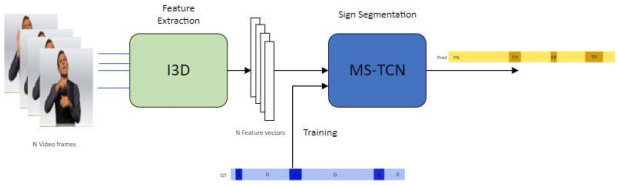


Figure 3: Proposed model from *[4]* which inspire this work, features are extracted from video frames as vectors using I3D. These features are used by the MS-TCN architecture for learning and predicting the segmentation of the signs.

[2] As explained in [79], the results of an annotating challenge [65] show that different teams come up with different annotations for the same signs [81]–[83].

3. 3.Methodology

In this section, the solution proposed, and the data used is described. It divides as follow. First, the sign segmentation problem is explained in section 3.1. Then, the datasets used and the split between training, validation and testing data is exposed in section 3.2. Finally, the method used for extracting the features, I3D is presented in section 3.3 and the ASFormer architecture in section 3.4.

3.1. Introduction

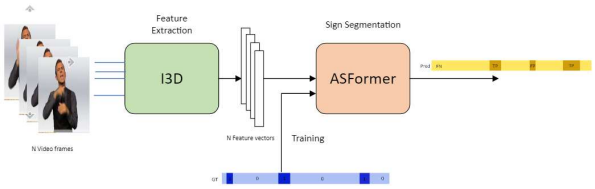


Figure 4: In our proposed model, features are extracted from video frames as vectors using I3D. These features are used by the ASFormer architecture for learning and predicting the segmentation of the signs.

While in sign recognition we are interested in the translation of signs into glosses or sentences, in sign segmentation we are only interested into finding the boundaries between signs, and detect the Movement Epenthesis, the parts that does not belong to any sign. We need to prepare a training dataset where frames are labeled as signs (1) and non-signs(0). Since there could be some labels between signs that are detected as non-signs we need to have a way to define a sign as a consecution of "sign" labels

Let be $V = \{V_1, \dots, V_n\}$ where n is the number of videos, $x = \{\{x_{11}, \dots, x_{1,m}\}, \dots \{x_{n1}, \dots, x_{nm}\}$ being x the frames of the video n with variable length m and the labels $y = \{y, \dots, y_{1,m}\}, \dots \{y_{n1}, \dots, y_{nm}\} \in \{0,1\}^n$ where 0 reflect a sign and 1 the boundary between signs respectively.

Model training: The proposed method takes a video sequence as input and generates a proposed sequence of labels. The model is divided into two steps , the feature extraction phase, which uses I3D and the segmentation phase which is based in ASFormer.

3.2. Datasets used

As already introduced in previous sections, it was decided to use the features already extracted from the dataset in the work of [4] since we are interested to measure the performance of ASFormer in comparison to previous works. These datasets where already presented in the state-of-the-art section and are RWTH-PHOENIX-Weather14[12], and BSLCorpus[10]. BSL-1k [55] which was used as well, was not publicly available, so it was not used in this study.

Dataset	Language		Training	Validation	Testing
BSL Corpus [10]	BSL	# videos	5413	763	703
		# signers	157	20	21
		Vocabulary	969	671	620
RWTH-PHOENIX-WEATHER 14 [12]	DGS	# videos	5672	540	629
		# signers	9	9	9
		Vocabulary	1081	467	500

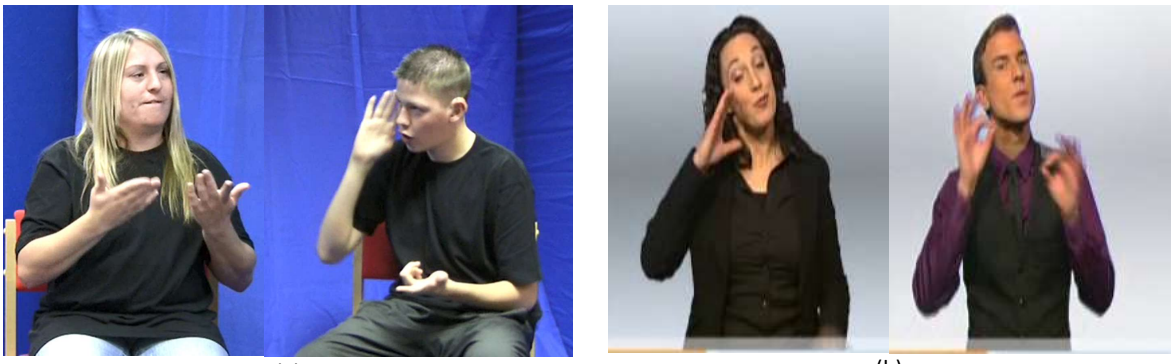
Table 3: Characteristics of the datasets used and division of the data intro training validation and testing sub sets.

Datasets are split into training and testing parts as shown in Table 3. However, we have used part of the training dataset of BSL corpus for hyperparameter optimization when performing k-cross validation with K=5. The dataset is divided into 5 and trained iteratively, each time taking 4 of the parts as training and one as the validation part. The final result is the mean of the 5 iterations.

In addition, the only dataset annotated in RWTH-Phoenix-Weather14 is the training part, following the work made by [67], for that reason, we split the training dataset into 4556 sequences of training and 1115 of testing. 3.3 Feature extraction

Feature extraction phase is the initial one, which takes the temporal convolutional I3D architecture [5], pretrained for action recognition on the Kinetics dataset [68] and BSLCorpus.

I3D is formed by two 3D Inflated ConvNet architectures trained separately: one based in RGB inputs and the other in optical flow algorithms. More in detail, Inflated 3D ConvNets are 2D ConvNets inflated into 3D by adding an additional temporal dimension to all the filters and pooling



(a) (b)

Figure 5: Examples of datasets from BSLCorpus (a) and RWTH-PHOENIX-Weather14 (b) datasets

kernels. This way, it avoids the more complexity of other 3D ConvNets architectures [69], that make models more difficult to train, but maintaining the same functionality.
The pretraining using Kinetics was adopted by previous work [4] arguing that it provide representation sensitive to fine-grained human motions. Also, in action segmentation task, its use improved significatively the results [5]. After that, I3D was also pretrained with BSLCorpus for making the model adapt better to sign language specific features.
As one of the objectives of our problem is to compare with the previous solutions found in the literature with using the new ASFormer approach , the features used are the ones already extracted in the works of [4] and [44] which are publicly available.

3.3. Segmentation process.

We replace the method MS-TCN used in previous works and instead use ASFormer[7] (for Action Segmentation transformer) which is transformers-based architecture adapted for the Action Segmentation problem, due to original transformer being not prepared for the specific requirements of this problem. Video input are normally too long and datasets too small for the original transformer model[70]. In consequence, there is a lack of inductive biases that difficult the training of the model and the *vanilla* transformer finds issues in forming an effective representation.
To avoid these limitations, ASFormer is designed with some modifications like removing positional encoding, adding dilated temporal convolution for bringing local inductive bias, or adding a hierarchical representation-pattern in the self-attention layer to improve cooperation between meaningful locations . These changes are described in more detail in Appendix 2.Transformers.
In addition, the loss used in both ASFormer and MS-TCN combines a classification loss and a smoothing loss for penalizing over-segmentation problem which is common in Action Segmentation problem.

$$L = L_{cls} + \lambda L_{smo} = -log(y_{t,\hat{c}}) + \lambda \frac{1}{TC} \sum_{t} \sum_{C} (y_{t-1,C} - y_{t,c})^2$$

4. Experimental results

Results obtained are presented in the following Sections. First, the metrics used are explained in section 4.1, then ablation studies for finding the best hyperparameters are performed in section 4.2. Finally, the final results are presented compared with previous works from the literature in section 4.3. An additional section 4.4 is added with considerations about the Computational Cost of ASFormer.

4.1. Evaluation metrics

The metrics used are the same ones proposed in [4], [44] based on the similarity of distance of the predicted boundaries to the ground truth and the width of the signs respectively. A boundary is considered a series of followings 1s.
mF1B. The boundary is considered correct if the distance of the boundary to the ground truth is lower than a threshold. mF1B is the mean of all F1 scores with thresholds of the interval [1,4]
mF1S. A sign segment is considered correct if the IoU between prediction and ground truth is higher than a threshold. mF1S is the mean of scores in the interval [0.4,0.75] with step size 0.05. This metric is not considered as important as MF1B as it is highly dependent on the annotator style[44].

4.2. Hyperparameters

There are several hyperparameters that are necessary to affine, starting with selection of the batch size and learning rate. The authors of ASFormer [7] recommend a batch size of 1, due to problems of lack of memory that could make the algorithm stopping unexpectedly so this value is the one used.[3] They also use a learning rate of 0.0005, a weight decay of 10^{-5} and a learning rate scheduler that reduces by half the learning rate when the loss doesn't improve for more than 3 epochs. These values are used in the following experiments.
Authors of ASFormer performed a study of the optimal number of decoders and blocks for the action segmentation problem concluding that the optimal solution is 3 decoders and 10 blocks [7]. A similar study is performed to study the same parameters in sign segmentation problem but doing a K-Cross Validation with K=5 to ensure results are statistically reliable. The training dataset was split into 5 parts and in each iteration one of these parts was used as validation set, while the others were using for training the model. Results could be seen in Table 4 for the comparison of number of decoders and Table 5 for the comparison of number of blocks

# Decoders	mF1B	mF1S
1	64.78 ± 0.88	55.1 ± 0.87
2	**65.42 ± 0.59**	**55.27 ± 0.98**
3	64.14 ± 2.28	54.02 ± 2.74
4	64.22 ± 2.6	54.31 ± 2.26

Table 4: Results of the BSLCorpus dataset for the variation of num of decoders with batch =1, lr= 0.0005, 10 blocks and epoch 50

[3] This was verified when performing preliminary experiments. Values higher than batch size of 8 give memory problems. However, it was decided to stick with the batch size of 1 as the original ASFormer recommended.

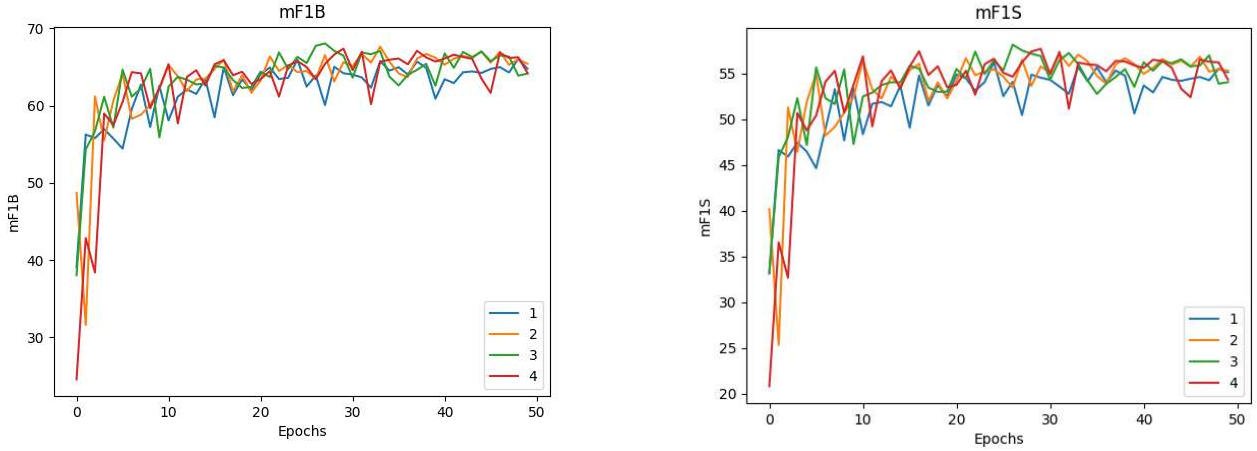
# Blocks	mF1B	mF1S
6	65.65 ± 1.8	55.43 ± 2.14
7	**66.64 ± 1.51**	**56.67 ± 1.43**
8	63.66 ± 5.29	52.92 ± 6.21
9	65.64 ± 2.62	55.48 ± 3.06
10	64.14 ± 2.28	54.02 ± 2.74
11	64.73 ± 0.8	54.96 ± 1.1

Table 5: Results of the BSLCorpus dataset for the variation of num of blocks inside each decoder with batch =1, lr= 0.0005 and 3 decoders. And epoch = 50

It could be seen that the best solution is that of 2 decoders and 7 blocks. However, looking more carefully into the evolution of the metrics over each iteration of the training, we could see a high variability in the results. For example, in **Error! Reference source not found.** we could see that results of 1 decoder are generally lower than the rest, but for the other results this is not so evident. Even more, maximum is reached for 3 decoders in epoch 28 with a value of mF1B =68.04.

Results given in the study of the relationship with the number of blocks in each decoder, show a similar behavior, as it can be seen in **Error! Not a valid bookmark self-reference.**. A maximum value of mF1B is reached in epoch 28 for 8 decoders.

A maximum value of mF1B is reached in epoch 28 for 8 decoders.



:Figure 6 Examples of datasets from BSLCorpus (a) and RWTH-PHOENIX-Weather14 (b) datasets

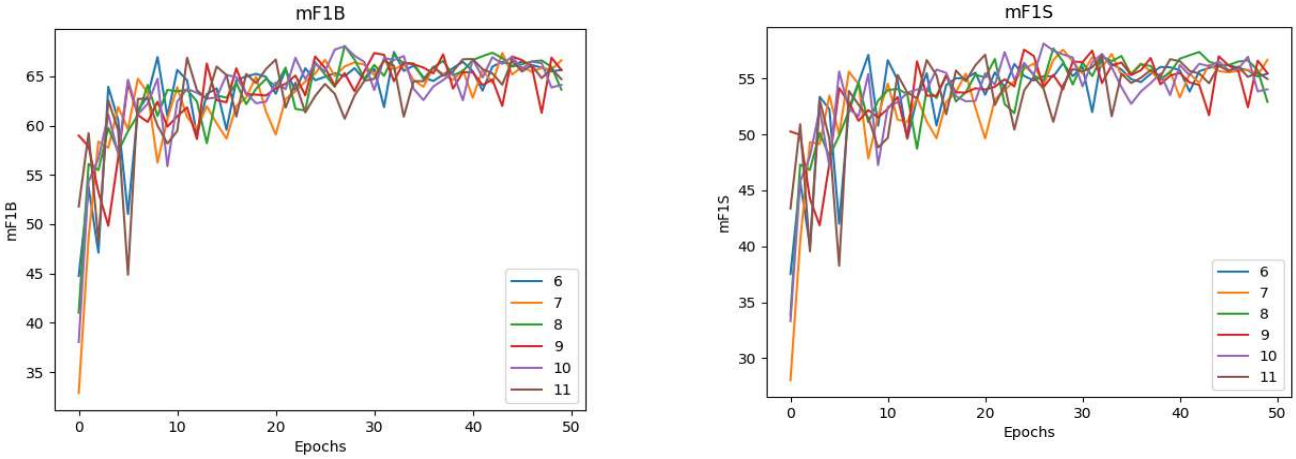


Figure 7: Comparison of the evolution of the performance per number of blocks and each iteration on the validation set. Graphs take the mean value of the K=5 iterations

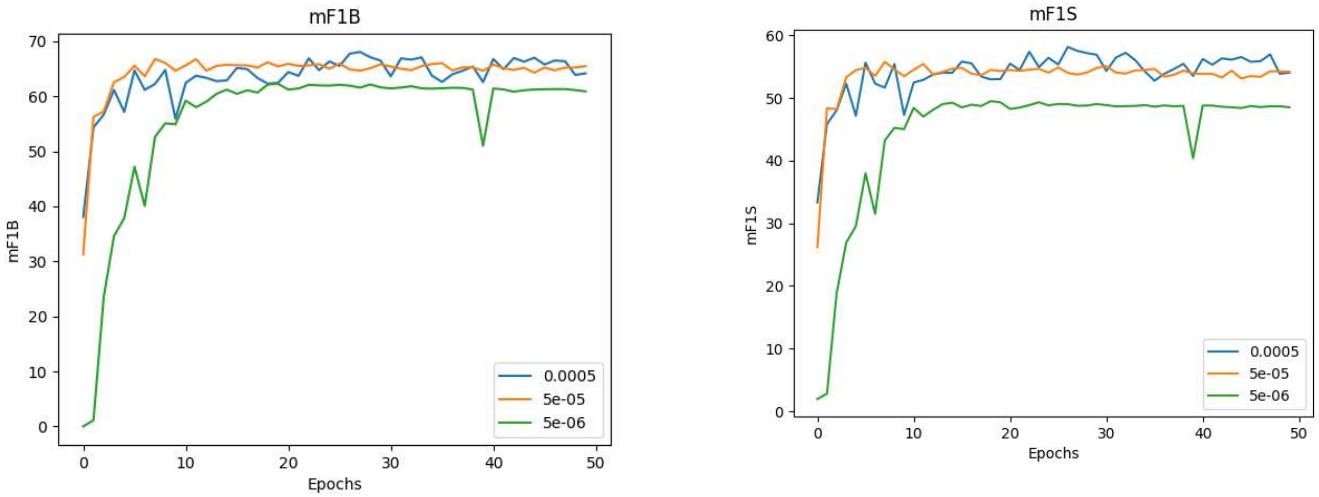


Figure 8: Comparison of the evolution of the performance per learning rate and each iteration on the validation set.

These results, make necessary the re-evaluation of the rest of the hyperparameters, starting with learning rate. Preliminary studies shown that using a value of 0.0005 gave slightly better results than the rest of values, this, and the fact that original ASFormer authors selected it in their experiments was the reason of using it in first place. However, after doing a K-cross validation it was evident that, even though a learning rate value of 0.0005 gives the best results on single iterations, the results of a learning rate of $50 \cdot 10^{-5}$ give better mean results and more stable over the epochs.

Learning rate	mF1B	mF1S
0.0005	64.14 ± 2.28	54.02 ± 2.74
$5\ 10^{-5}$	**65.48 ± 0.91**	**54.17 ± 0.79**
$5\ 10^{-6}$	60.89 ± 0.64	48.51 ± 0.42

Table 6: Results of the BSLCorpus dataset on epoch 50 for the variation of learning rate for a batch size of 1, 3 decoders and 10 blocks. Results above 0.005 were not reflected since they do not converge

Having seen that $5 \cdot 10^{-5}$ give more stable results, the study of the number of decoders and blocks inside them should be repeated for more insightful results.

4.3. Comparison in performance

Having studied the hyperparameters in previous section, we run 5 times our model with the best results given for a combination of batch size of 1 and a learning rate of 0.0005, which was 2 decoders and 7 blocks per decoder. Since results from other studies were studied for only 10 epochs and 3 different seeds, it was decided to repeat the training for 50 epochs and 5 different seeds.

Comparative results can be in the following tables por BSL Corpus Table 7 and Table 8 and for Phoenix-Weather14 in Table 9 and Table 10. For the results of Phoenix-Weather14, we are comparing the results when using I3D features pretrained with BSLCorpus dataset, not with Phoenix-Weather14. We could expect a better performance if the I3D is pretrained with the examples of its own dataset as it was proved in [44].

	mF1B	mF1S
Geometric features + RF [43]	50.49 ± 0.1	37.46 ± 0.1
I3D + MS-TCN [4]	**68.68 ± 0.6**	**47.71 ± 0.8**
I3D + MS-TCN (Current experiment)	**68.21± 0.61**	**46.24 ± 1.64**
I3D + ASFormer	66.26 ± 1.29	44.10 +- 2.11

Table 7: Results in the BSL-Corpus dataset with a training of 10 epochs. First two rows were extracted from [4] and were repeated 3 times. In our experiments it was decided to repeat them 5 times.

	mF1B	mF1S
I3D + MS-TCN (Current experiment)	65.17 ± 0.14	**48.65 ± 0.63**
I3D + ASFormer (Ours -50 epoch)	**65.89±1.35**	44.89± 1.93

Table 8: Results in the BSL-Corpus dataset with a training of 50 epochs and 5 repetitions.

In general, it could be seen how results achieved by MS-TCN are better than the ones of ASFormer in all examples. Results are better in ASFormer for 50 epochs, but MS-TCN performed better with only 10 epochs in all cases.

In Table 9, results from [44] paper are included as the actual state of the art of PHOENIX-Weather14 benchmark, but, since they apply an additional pseudo-labelling process, results are not fully comparable, as it is expected that the use of such technique in ASFormer would also improve the results obtained.

	mF1B	mF1S
I3D + MS-TCN [4]	65.06±0.5	44.42±2.0
CMPL inductive [44]	65.99±1.0	33.82 ± 0.0
CMPL inductive [44]	**67.01±2.2**	**49.96 ± 0.6**
I3D + MS-TCN (Current experiment)	64.71±0.74	42.85±2.73
I3D + ASFormer (Ours)	63.86±0.32	40.37 ± 1.43

Table 9: Results in the PHOENIX dataset with a training of 10 epochs. First three rows were extracted from [4]and were repeated 3 times. In our experiments it was decided to repeat them 5 times.

	mF1B	mF1S
I3D + MS-TCN (Current experiment)	60.36±0.31	**45.10±0.31**
I3D + ASFormer (Ours)	**65.88±0.77**	43.67 ± 1.03

Table 10: Results in the PHOENIX dataset with a training of 50 epochs

5. Discussion

Sign Language Segmentation task separates the signs in a Continuous Sign Language Sentence into the different signs without knowing their meaning. The purpose of this paper was, as explained before, study the use of a transformer-based architecture, confirming if it was suitable for this task, and comparing with the top state of the art solutions.

Analyzing the results from the previous section, it could be seen that replacing MS-TCN architecture used in previous work of [4] with ASFormer give a similar performance, but slightly lower. This confirms the viability of such solution for the sign segmentation task.

However, we have faced issues in finding the optimal hyperparameters due to a high variation in each iteration of the process for the learning rate selected in first place. The study of hyperparameters should continue since we expect an improvement in results for the final solution. On that matter, final ablation study on learning rate showed that the use of a learning rate of 10^{-5} for a batch size of 1 gives more stable results.

Aside from this, the particularities of ASFormer should be carefully analyzed to find the reason why it doesn't reach as promising results as in the Action Segmentation task. First, it should be noted that in action segmentation, tasks usually are formed by slower movements that in sign segmentation and sometimes involve objects (like preparing the food or eating). Sign segmentation, on the contrary, takes usually fast movements that involve only the hands, face, and body of the signer. Hands also took complex shapes and sometimes there are occlusions that complicate the processing of this information.

Knowing this, ASFormer architecture authors made some decisions for improving performance in Action Segmentation, like removing the positional encoding step of the encoder and decoder blocks of the transformer, due to the use of temporal convolutions made it redundant. The higher complexity of signs could make the use positional encoding more important in this case. Modifying these architectures and adapt them for the specific needs could be the answer for improving the performance already achieved with ASFormer.

In addition, we have seen a huge difference in the computational cost used for training ASFormer related to the one expended for training MS-TCN, which is much lower. We must consider if it is worthy to use ASFormer when results are not improving the ones of MS-TCN.

As coming back to the research question made at the beginning of this paper **Is the ASFormer a suitable solution for Sign Segmentation task improving previous state of the art results?** We should say that ASFormer could be used for sign segmentation, as it is near the best results achieved in the actual state of the art but being the training computationally more expensive and the results slightly worse, this is discouraged unless a more optimal set of hyperparameters is found. Also, we could foresee the promise of using transformer-based approaches for Sign Segmentation, where modifications in the ASFormer architecture adapted to this task could lead to even better performances.

As for the applications of this architecture, the most evident is that of helping in building new datasets and annotating the existing ones. Annotators could benefit from an initial segmentation of the signs and their work will transform into reviewing that the signs are correctly split and correcting only the errors. Also, the use of this architecture as a module of sign language translation could help in improving communication between deaf and speaking communities both helping in translating between sign languages, or between a sign language and a spoken one.

5.1. Limitations

Apart from the hyperparameter tuning issues and high computational cost discussed in previous section, there were two important limitations in this study. First, the need of even bigger datasets for training, but this is exactly what motivated this work on first place. The other limitation was the need a huge computation power for managing the computational cost of ASFormer and the k-cross validation studies made for hyperparameter tuning. Resources available, where limited which makes important to seek for more efficient alternatives.

Another possible limitation of this approach is that, while ASFormer is designed for Action Segmentation task, which is more general problem that Sign Segmentation, the differences between both tasks could explain why ASFormer didn't get even better results. Action segmentation benchmarks where ASFormer was tested on first place, are composed by long videos where actions are usually longer than the duration of signs in sign language, and there are also some objects involved. In contrast, sign

segmentation uses benchmarks with videos where the signs change very fast, and are composed only by the hands, face, and body of the signer. A deeper study on the characteristics of the normal datasets that sign segmentation faces proves necessary to modify the architecture

In addition, it would have been ideal to train our model in other languages, but the lack of annotated datasets and time constraints made it impossible. Even though, thanks to similarities between different sign languages, we could foresee similar performances.

5.2. Future work

There are several lines of work that could be interesting to follow, the most evident is to continue studying the influence of the hyperparameters of ASFormer in Action Segmentation to find the optimal set, adding weight decay as well and the learning rate scheduler hyperparameters. Also, it could be interesting to use a different loss more adapted to the specifical characteristics of the problem of Sign Segmentation as it was made in ASRF [50] for the Action Segmentation task or try to pretrain I3D with different strategies for capturing better the movements associated with sign language. For alleviating the high computation cost of studying this set of parameters alternative approaches can be considered such as DEHB [71], an algorithm that combines Hyperband with a Differential Evolution approach for exploring the hyperparameter results.

Another approach to explore is adding pseudo-label algorithm for improving the results, like it was presented in [44]. Taking into consideration that this method improved significatively the results from the original MS-TCN solution, it is expected that replicating it with ASFormer can give better results as well.

There are several newer solutions that improve results on the Action Segmentation task that worth to be explored in upcoming investigation. To name a few: Cross-Enhancement Transformer (CETNet) [72], Efficient U-Transformer(EUT) [73] and Unified Video Action Segmentation model via Transformers (UVAST), which views action segmentation as a seq2seq task instead of a frame-level prediction [74]. Additionally, another interesting approach to explore is the one inspired by UARL (Uncertainty-aware representation learning) method [75] which treats action segmentation boundaries not as abrupt changes, but as gradual transitions, something that could suit very well to the phenomenon of movement epenthesis.

Aside from the action segmentation task, it would be more important to focus in finding an optimal architecture for sign segmentation, taking into account its characteristics as explained in previous section. It would be interesting to search the influence of the removal of the positional encoding step in the transformer, as well as finding solutions for capturing the fast movements of sign language.

Another possible line of study could be applying this segmentation algorithm for the sign recognition problem as a middle section between the feature extraction and the learning sequence steps.

Finally, our intention is to create a Systematic Review paper with all the sign segmentation related articles found and analyzed in detail.

5.3. Ethical considerations

The most important ethical considerations of this work are in relationship with the data, its source, composition, and the use we give to it.

All the data used in this study, was publicly available for research purposes. Even more, thanks to the use of I3D features instead of real images, this method could be used for training models without the need of the images or videos of the signer, which is an important step for privacy.

Also, data used in this study ensures there is enough variety of signers in age, gender, and ethnicity as we analyze the datasets used. BSLCorpus [76] has examples recorded from 192 signers of different regions of United Kingdom from different ages, backgrounds and origins. Phoenix-Weather could pose more a problem in that sense, since the variety of signers is much lower (only 9).

Finally, as per the use of this model, it is expected to use it specially in the building of datasets. However, if it is needed in some real application, I3D and ASFormer models could be shared as there is no identification over the users that participated in the data used for training this model.

5.4. Lessons learnt

In this section, I give some considerations about lessons learnt in the research process.

Firstly, I want to highlight the importance of the systematic review performed as it was key for (1) identifying the lack of annotated datasets as one of the biggest problems in sign language recognition, (2) discovering automatic sign segmentation as possible solution for it, and (3) learnt about past solutions given to this problem for getting the inspiration for the final proposal. However, as a negative note, I should note that following PRISMA guidelines, which ensures the rigor of the approach, has a steep high curve that may not be convenient for a Master thesis. PRISMA structure follows closely the final review paper, focusing too early in defining the abstract, writing the introduction, etc. I have missed a more natural approach, where defining the research questions, identifying the search engines, or building up the correct research queries comes first

While reviewing the literature, I have noticed that several studies did not do enough repetitions, and use little datasets, so their results were not fully reliable or statistically significative. This analysis would be reflected on the upcoming systematic review paper. To avoid this, I employed k-cross validation for hyperparameter finetuning. Ideally, an optimal k value would have been 10, but it was decided to use 5 because of time constraints.

Related to this, however, I should point out the importance of reaching a compromise about using statistically reliable results and making the first surveys, specially when working with models that requires a high computational cost and the computational resources are limited. Because of this, bad hyperparameter decisions lead to a full week lost. On the other hand, when doing a single execution for selecting the learning rate, it appeared that the best solution was 0.0005, but .further executions made visible that the mean was not as it was not stable enough to appreciate differences between different solutions. To alleviate this, it would have been worthy investing time preparing TensorFlow for looking at the loss in the training and validation tests, since it would have given tips about models overfitting.

Related to time constraints, another important lesson acquired is the importance of investing time in parallelization of the tasks, specially on co-validation. Having used a batch size of 1, the use of a GPU was not so important, but the use of different threads reduced the time of the process from 11 days of execution to 7 which was already a huge improvement. However, parallelization is a double-edged sword, because too many threads could compete for the CPU cores and make the final time even longer. Other important investment would have to program an Early Stopping strategy, at least for preliminary results as I would have saved time with it.

Finally, I should mention the importance of selecting a good strategy for hyperparameter finetuning. I have invested a lot of time on making DEHB algorithm [71] working, which used Hyperband and Differential Evolution for exploring the space of all hyperparameters and select a good set of them. Again, due to time constraints it was not possible to use it at the end, as I decided to perform a classic ablation study, but it would have been a good contribution for this paper since I would have avoided some of the issues experienced with the manual selection of hyperparameters for my experiments.

6. Conclusion

In this paper, the combination of I3D features with the ASFormer architecture is explored as a viable solution to the sign segmentation problem. This solution gives results near the top state of the art that show the promise of using transformed-based approaches in this task. Further study should be performed to find the optimal set of hyperparameters.

7. Author Contributions:

Conceptualization, L.P.V., S.R.V., O.C.S.; Methodology, L.P.V., S.R.V., O.C.S.; Software, L.P.V.; Validation, L.P.V.; Formal Analysis, L.P.V.; Investigation, L.P.V.; Resources, L.P.V., S.R.V., O.C.S.; Data Curation, L.P.V.; Writing—Original Draft Preparation, L.P.V.; Writing—Review and Editing, L.P.V., S.R.V., O.C.S.; Visualization, L.P.V.; Project Administration, S.R.V., O.C.S.; Funding Acquisition, n/a.

8. Acknowledgement

I would like to thank José Ramón Álvarez Sánchez from UNED with his invaluable assistance helping with the use of Tesla machine for the experiments. I would also like to thank José Luis Alba Castro from the University of Vigo for his interest and counsel at the beginning of the project and his assistance with UVigo dataset.

Data:

Part of the data used in this article were collected for the British Sign Language Corpus Project (BSLCP) at University College London, funded by the Economic and Social Research Council UK (RES-620-28-6001), and supplied by the **CAVA repository**. The data are copyright. Phoenix dataset is available at https://www-i6.informatik.rwth-aachen.de/~koller/RWTH-PHOENIX/ I3D extracted features used in this study where trained by the work of Renz et al. [4] [44]and made publicly available in https://github.com/RenzKa/sign-segmentation

Code: It was built based on the code located on the following repositories:

- ASFormerhttps://github.com/ChinaYi/ASFormer
- I3D + MS-TCN Sign Segmentation [4] [44]: https://github.com/RenzKa/sign-segmentation repository, which was built, in turn, using github.com/yabufarha/ms-tcn and github.com/gulvarol/bsl1k. repositories.

9. References

[1] E. B. Braaten, “Deafness and Hearing Loss,” The SAGE Encyclopedia of Intellectual and Developmental Disorders, 2018. https://www.who.int/news-room/fact-sheets/detail/deafness-and-hearing-loss (accessed Mar. 06, 2021).

[2] N. Mohamed, M. B. Mustafa, and N. Jomhari, “A Review of the Hand Gesture Recognition System: Current Progress and Future Directions,” IEEE Access, vol. 9, pp. 157422–157436, 2021, doi: 10.1109/ACCESS.2021.3129650.

[3] O. Koller, “Quantitative Survey of the State of the Art in Sign Language Recognition,” arXiv, Aug. 2020, Accessed: Feb. 21, 2021. [Online]. Available: http://arxiv.org/abs/2008.09918.

[4] K. Renz, N. C. Stache, S. Albanie, and G. Varol, “Sign Language Segmentation with Temporal Convolutional Networks,” ICASSP, IEEE Int. Conf. Acoust. Speech Signal Process. - Proc., vol. 2021-June, pp. 2135–2139, 2021, doi: 10.1109/ICASSP39728.2021.9413817.

[5] J. Carreira and A. Zisserman, “Quo Vadis, action recognition? A new model and the kinetics dataset,” Proc. - 30th IEEE Conf. Comput. Vis. Pattern Recognition, CVPR 2017, vol. 2017-Janua, pp. 4724–4733, 2017, doi: 10.1109/CVPR.2017.502.

[6] Y. A. Farha and J. Gall, “MS-TCN: Multi-stage temporal convolutional network for action segmentation,” Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit., vol. 2019-June, pp. 3570–3579, 2019, doi: 10.1109/CVPR.2019.00369.

[7] F. Yi, H. Wen, and T. Jiang, “ASFormer: Transformer for Action Segmentation,” pp. 1–19, 2021, [Online]. Available: http://arxiv.org/abs/2110.08568.

[8] A. Schembri, J. Fenlon, R. Rentelis, and K. Cormier, British Sign Language Corpus Project: A corpus of digital video data of British Sign Language 2008-2014 (Second Edition), Second Edi. 2014.

[9] A. Schembri, J. Fenlon, R. Rentelis, and K. Cormier, British Sign Language Corpus Project: A corpus of digital video data of British Sign Language 2008-2017 (Third Edition), Third. 2017.

[10] A. Schembri, J. Fenlon, R. Rentelis, S. Reynolds, and K. Cormier, “Building the British Sign Language Corpus,” Lang. Doc. Conserv., vol. 7, pp. 136–154, 2013.

[11] O. Koller, J. Forster, and H. Ney, “Continuous sign language recognition: Towards large vocabulary statistical recognition systems handling multiple signers,” Comput. Vis. Image Underst., vol. 141, pp. 108–125, 2015, doi: 10.1016/j.cviu.2015.09.013.

[12] J. Forster, C. Schmidt, O. Koller, M. Bellgardt, and H. Ney, “Extensions of the sign language recognition and translation corpus RWTH-PHOENIX-Weather,” Proc. 9th Int. Conf. Lang. Resour. Eval. Lr. 2014, pp. 1911–1916, 2014.

[13] M. J. Page et al., “The PRISMA 2020 statement: An updated guideline for reporting systematic reviews,” BMJ, vol. 372, 2021, doi: 10.1136/bmj.n71.

[14] M. J. Page et al., “PRISMA 2020 explanation and elaboration: Updated guidance and exemplars for reporting systematic reviews,” BMJ, vol. 372, 2021, doi: 10.1136/bmj.n160.

[15] J. Segouat and A. Braffort, “Toward modeling sign language coarticulation,” 8th International Gesture Workshop: Gesture in Embodied Communication and Human-Computer Interaction, GW 2009, vol. 5934. ["LIMSI-CNRS, BP 133, 91400 Orsay, France", “WebSourd, Bât A, 99 Route d’Espagne, 31100 Toulouse, France”] Bielefeld, pp. 325–336, 2009, [Online]. Available: https://www.scopus.com/inward/record.uri?eid=2-s2.0-78650359085&doi=10.1007%2F978-3-642-12553-9_29&partnerID=40&md5=b564a8c8f798adaa3ea5325ee0512996.

[16] [S. Khan, “Segmentation of Continuous Sign Language,” pp. 1–156, 2014.

[17] [W. W. Kong and S. Ranganath, “Towards subject independent continuous sign language recognition: A segment and merge approach,” Pattern Recognit., vol. 47, no. 3, pp. 1294–1308, 2014, [Online]. Available: https://www.scopus.com/inward/record.uri?eid=2-s2.0-84888325954&doi=10.1016%2Fj.patcog.2013.09.014&partnerID=40&md5=eea55741fcfcb43c6beeb95a53f5cde6.

[18] J. Singha and R. H. Laskar, “Self co-articulation detection and trajectory guided recognition for dynamic hand gestures,” IET Comput. Vis., vol. 10, no. 2, pp. 143–152, 2016, [Online]. Available: https://www.scopus.com/inward/record.uri?eid=2-s2.0-84959051410&doi=10.1049%2Fiet-cvi.2014.0432&partnerID=40&md5=131781ab3be5e2aa8a323ab988d67689.

[19] L. Naert, C. Larboulette, and S. Gibet, “Coarticulation Analysis for Sign Language Synthesis,” UNIVERSAL ACCESS IN HUMAN-COMPUTER INTERACTION: DESIGNING NOVEL INTERACTIONS, PT II, vol. 10278. pp. 55–75, 2017.

[20] P. K. Athira, C. J. Sruthi, and A. Lijiya, “A Signer Independent Sign Language Recognition with Co-articulation Elimination from Live Videos: An Indian Scenario,” J. King Saud Univ. - Comput. Inf. Sci., vol. 34, no. 3, pp. 771–781, 2022, [Online]. Available: https://www.scopus.com/inward/record.uri?eid=2-s2.0-85065538601&doi=10.1016%2Fj.jksuci.2019.05.002&partnerID=40&md5=bd2c49b38aa12e2dfb8af292872c9c8c.

[21] A. Choudhury, A. K. Talukdar, M. K. Bhuyan, and K. K. Sarma, “Movement Epenthesis Detection for Continuous Sign Language Recognition,” J. Intell. Syst., vol. 26, no. 3, pp. 471–481, 2017, [Online]. Available: https://www.scopus.com/inward/record.uri?eid=2-s2.0-85023176634&doi=10.1515%2Fjisys-2016-0009&partnerID=40&md5=2edb1e38a757aa6459fbc3df4f1c7583.

[22] A. K. Talukdar and M. K. Bhuyan, “Movement epenthesis detection in continuous fingerspelling from a coarsely sampled motion vector field in H.264/AVC video,” in 2018 IEEE Recent Advances in Intelligent Computational Systems, RAICS 2018, 2019, pp. 26–30, [Online]. Available: https://www.scopus.com/inward/record.uri?eid=2-s2.0-85062853629&doi=10.1109%2FRAICS.2018.8634902&partnerID=40&md5=006084a0d5be11131fe78f8661d9a0e4.

[23] R. Elakkiya and K. Selvamani, “Enhanced dynamic programming approach for subunit modelling to handle segmentation and recognition ambiguities in sign language,” J. Parallel Distrib. Comput., vol. 117, pp. 246–255, 2018, [Online]. Available:

https://www.scopus.com/inward/record.uri?eid=2-s2.0-85026765786&doi=10.1016%2Fj.jpdc.2017.07.001&partnerID=40&md5=6e0cccc24739cb1ce24c857a8c785e0a.

[24] R. Elakkiya and K. Selvamani, "Subunit sign modeling framework for continuous sign language recognition," Comput. Electr. Eng., vol. 74, pp. 379–390, 2019, [Online]. Available: https://www.scopus.com/inward/record.uri?eid=2-s2.0-85061728474&doi=10.1016%2Fj.compeleceng.2019.02.012&partnerID=40&md5=a430271e8467908d252ef755c978579c.

[25] N. Nayan, D. Ghosh, and P. M. Pradhan, "An optical flow based approach to detect movement epenthesis in continuous fingerspelling of sign language," 2021, [Online]. Available: https://www.scopus.com/inward/record.uri?eid=2-s2.0-85116109605&doi=10.1109%2FNCC52529.2021.9530076&partnerID=40&md5=307051e98271960cb1a4e8b9da94e4cc.

[26] S. Nayak, S. Sarkar, and B. Loeding, "Unsupervised modeling of signs embedded in continuous sentences," IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. Work., vol. 2005-Septe, 2005, doi: 10.1109/CVPR.2005.547.

[27] S. Nayak, S. Sarkar, and B. Loeding, "Automated extraction of signs from continuous sign language sentences using iterated conditional modes," 2009 IEEE Conf. Comput. Vis. Pattern Recognition, CVPR 2009, pp. 2583–2590, 2009, doi: 10.1109/CVPRW.2009.5206599.

[28] P. Santemiz, O. Aran, M. Saraclar, and L. Akarun, "Automatic sign segmentation from continuous signing via multiple sequence alignment," 2009 IEEE 12th Int. Conf. Comput. Vis. Work. ICCV Work. 2009, pp. 2001–2008, 2009, doi: 10.1109/ICCVW.2009.5457527.

[29] [29] A. Choudhury, A. K. Talukdar, K. K. Sarma, and M. K. Bhuyan, "An Adaptive Thresholding-Based Movement Epenthesis Detection Technique Using Hybrid Feature Set for Continuous Fingerspelling Recognition," SN Comput. Sci., vol. 2, no. 2, pp. 1–21, 2021, doi: 10.1007/s42979-021-00544-5.

[30] H. Chaaban, M. Gouiffès, and A. Braffort, "Automatic annotation and segmentation of sign language videos: Base-level features and lexical signs classification," VISIGRAPP 2021 - Proc. 16th Int. Jt. Conf. Comput. Vision, Imaging Comput. Graph. Theory Appl., vol. 5, pp. 484–491, 2021, doi: 10.5220/0010247104840491.

[31] W. W. Kong and S. Ranganath, "Sign language phoneme transcription with rule-based hand trajectory segmentation," J. Signal Process. Syst., vol. 59, no. 2, pp. 211–222, 2010, [Online]. Available: https://www.scopus.com/inward/record.uri?eid=2-s2.0-77951258217&doi=10.1007%2Fs11265-008-0292-5&partnerID=40&md5=f2fce5511d442ca505849fa8312f7631.

[32] X. Meng et al., "Sentence-Level Sign Language Recognition Using RF signals," BESC 2019 - 6th Int. Conf. Behav. Econ. Socio-Cultural Comput. Proc., pp. 1–6, 2019, doi: 10.1109/BESC48373.2019.8963177.

[33] M. Gonzalez and C. Collet, "Sign segmentation using dynamics and hand configuration for semi-automatic annotation of sign language corpora," Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics), vol. 7206 LNAI, pp. 204–215, 2012, doi: 10.1007/978-3-642-34182-3_19.

[34] M. Gonzalez, M. Filhol, and C. Collet, "Semi-automatic sign language corpora annotation using lexical representations of signs," in 8th International Conference on Language Resources and Evaluation, LREC 2012, 2012, pp. 2430–2434, [Online]. Available: https://www.scopus.com/inward/record.uri?eid=2-s2.0-85037362447&partnerID=40&md5=058f7e6daa6c3f3f56f43d946041fbf9.

[35] R. Yang, S. Sarkar, and B. Loeding, "Handling movement epenthesis and hand segmentation ambiguities in continuous sign language recognition using nested dynamic programming," IEEE Trans. Pattern Anal. Mach. Intell., vol. 32, no. 3, pp. 462–477, 2010, [Online]. Available: https://www.scopus.com/inward/record.uri?eid=2-s2.0-73549100193&doi=10.1109%2FTPAMI.2009.26&partnerID=40&md5=05ac77812559a2fb0db4e9bf1d25725c.

[36] R. Elakkiya, K. Selvamani, and S. Kanimozhi, "Hand gesture recognition framework for recognizing sign gestures and handling movement epenthesis using Level Building nested dynamic programming approach," 2014, [Online]. Available: https://www.scopus.com/inward/record.uri?eid=2-s2.0-84908431009&doi=10.1109%2FCCECE.2014.6901162&partnerID=40&md5=17b2fcb4c67b74fb2cdbb322fdff0897.

[37] R. Elakkiya and K. Selvamani, "An Active Learning Framework for Human Hand Sign Gestures and Handling Movement Epenthesis Using Enhanced Level Building Approach," INTERNATIONAL CONFERENCE ON COMPUTER, COMMUNICATION AND CONVERGENCE (ICCC 2015), vol. 48. pp. 606–611, 2015.

[38] W. Yang, J. Tao, and Z. Ye, "Continuous sign language recognition using level building based on fast hidden Markov model," Pattern Recognit. Lett., vol. 78, pp. 28–35, 2016, doi: 10.1016/j.patrec.2016.03.030.

[39] C. Vogler and D. Metaxas, "Toward Scalability in ASL Recognition : Breaking Down Signs into Phonemes 1 Introduction 2 Related Work," pp. 1–12, 1999.

[40] J. Wu, W. Gao, J. Liang, and X. Wu, "A greedy clustering algorithm along the time axis for Chinese language recognition," in Proceedings of the IEEE International Conference on Systems, Man and Cybernetics, 2001, vol. 4, pp. 2439–2444, doi: 10.1109/icsmc.2001.972923.

[41] R. Yang and S. Sarkar, "Detecting coarticulation in sign language using conditional random fields," Proc. - Int. Conf. Pattern Recognit., vol. 2, pp. 108–112, 2006, doi: 10.1109/ICPR.2006.431.

[42] H. Cooper and R. Bowden, "Learning signs from subtitles: A weakly supervised approach to sign language recognition," 2009 IEEE Conf. Comput. Vis. Pattern Recognition, CVPR 2009, pp. 2568–2574, 2009, doi: 10.1109/CVPRW.2009.5206647.

[43] I. Farag and H. Brock, "Learning Motion Disfluencies for Automatic Sign Language Segmentation," ICASSP, IEEE Int. Conf. Acoust. Speech Signal Process. - Proc., vol. 2019-May, pp. 7360–7364, 2019, doi: 10.1109/ICASSP.2019.8683523.

[44] K. Renz, N. C. Stache, N. Fox, G. Varol, and S. Albanie, "Sign segmentation with changepoint-modulated pseudo-

labelling," 2021 IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Work. CVPRW 2021, pp. 3398–3407, 2021, [Online]. Available: https://www.scopus.com/inward/record.uri?eid=2-s2.0-85116010244&doi=10.1109%2FCVPRW53098.2021.00379&partnerID=40&md5=aa8af2467b4e4115dac80c76aa5ccba4.

[45] J. Pu, W. Zhou, H. Li, and L. J., "Dilated convolutional network with iterative optimization for continuous sign language recognition," in 27th International Joint Conference on Artificial Intelligence, IJCAI 2018, 2018, vol. 2018, pp. 885–891, [Online]. Available: https://www.scopus.com/inward/record.uri?eid=2-s2.0-85055682784&doi=10.24963%2Fijcai.2018%2F123&partnerID=40&md5=20594e6eb9bdad99f9799f11edd28fb6.

[46] D. Guo, S. Tang, M. Wang, K. S., and B. Xiao-i, "Connectionist temporal modeling of video and language: A joint model for translation and sign labeling," in 28th International Joint Conference on Artificial Intelligence, IJCAI 2019, 2019, vol. 2019, pp. 751–757, [Online]. Available: https://www.scopus.com/inward/record.uri?eid=2-s2.0-85074915392&doi=10.24963%2Fijcai.2019%2F106&partnerID=40&md5=67a35be34db913f68d42bcd570fe7578.

[47] X. Pei, D. Guo, Y. Zhao, and A. C. M. SIGMM, "Continuous sign language recognition based on pseudo-supervised learning," 2nd Work. Multimed. Access. Hum. Comput. Interfaces, MAHCI 2019, conjunction with ACM Multimed. 2019, pp. 33–39, 2019, [Online]. Available: https://www.scopus.com/inward/record.uri?eid=2-s2.0-85075152978&doi=10.1145%2F3347319.3356837&partnerID=40&md5=c4ba2e3df65574e06eee5b392207ec87.

[48] H. Zhou, W. Zhou, H. Li, and et al.; iQIYI; JD.Com; Kuaishou; MEGVII; Microsoft, "Dynamic pseudo label decoding for continuous sign language recognition," Proc. - IEEE Int. Conf. Multimed. Expo, vol. 2019-July, pp. 1282–1287, 2019, doi: 10.1109/ICME.2019.00223.

[49] S. Theodorakis, V. Pitsikalis, and P. Maragos, "Dynamic-static unsupervised sequentiality, statistical subunits and lexicon for sign language recognition," Image Vis. Comput., vol. 32, no. 8, pp. 533–549, 2014, doi: 10.1016/j.imavis.2014.04.012.

[50] Y. Ishikawa, S. Kasai, Y. Aoki, and H. Kataoka, "Alleviating over-segmentation errors by detecting action boundaries," Proc. - 2021 IEEE Winter Conf. Appl. Comput. Vision, WACV 2021, pp. 2321–2330, 2021, doi: 10.1109/WACV48630.2021.00237.

[51] J. Forster et al., "RWTH-PHOENIX-weather: A large vocabulary sign language recognition and translation corpus," Proc. 8th Int. Conf. Lang. Resour. Eval. Lr. 2012, pp. 3785–3789, 2012.

[52] O. Koller, J. Forster, and H. Ney, "Continuous sign language recognition: Towards large vocabulary statistical recognition systems handling multiple signers," 2015. doi: 10.1016/j.cviu.2015.09.013.

[53] N. C. Camgoz, S. Hadfield, O. Koller, H. Ney, and R. Bowden, "Neural Sign Language Translation," Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit., pp. 7784–7793, 2018, doi: 10.1109/CVPR.2018.00812.

[54] S. K. Ko, C. J. Kim, H. Jung, and C. Cho, "Neural sign language translation based on human keypoint estimation," Appl. Sci., vol. 9, no. 13, pp. 1–19, 2019, doi: 10.3390/app9132683.

[55] S. Albanie et al., "BSL-1K: Scaling Up Co-articulated Sign Language Recognition Using Mouthing Cues," Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics), vol. 12356 LNCS, pp. 35–53, 2020, doi: 10.1007/978-3-030-58621-8_3.

[56] H. Zhou, W. Zhou, W. Qi, J. Pu, and H. Li, "Improving Sign Language Translation with Monolingual Data by Sign Back-Translation," Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit., pp. 1316–1325, 2021, doi: 10.1109/CVPR46437.2021.00137.

[57] S. Albanie et al., "BBC-Oxford British Sign Language Dataset," pp. 1–15, 2021, [Online]. Available: http://arxiv.org/abs/2111.03635.

[58] A. Duarte et al., "How2Sign: A Large-scale Multimodal Dataset for Continuous American Sign Language," Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit., pp. 2734–2743, 2021, doi: 10.1109/CVPR46437.2021.00276.

[59] M. Kipp, "ANVIL A generic annotation tool for multimodal dialogue," EUROSPEECH 2001 - Scand. - 7th Eur. Conf. Speech Commun. Technol., pp. 1367–1370, 2001, doi: 10.21437/eurospeech.2001-354.

[60] P. Wittenburg, H. Brugman, A. Russel, A. Klassmann, and H. Sloetjes, "ELAN: A professional framework for multimodality research," Proc. 5th Int. Conf. Lang. Resour. Eval. Lr. 2006, pp. 1556–1559, 2006.

[61] K. Cormier and J. Fenlon, "BSL Corpus Annotation Guidelines," no. October, pp. 1–14, 2014.

[62] M. N. De Amorim, C. A. S. Santos, and O. De L. Tavares, "A Crowdsourcing Method for Sign Segmentation in Brazilian Sign Language Videos," ACM Int. Conf. Proceeding Ser., pp. 105–112, 2020, doi: 10.1145/3428658.3431083.

[63] L. Momeni, G. Varol, S. Albanie, T. Afouras, and A. Zisserman, "Watch, Read and Lookup: Learning to Spot Signs from Multiple Supervisors," Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics), vol. 12627 LNCS, pp. 291–308, 2021, doi: 10.1007/978-3-030-69544-6_18.

[64] G. Varol, L. Momeni, S. Albanie, T. Afouras, and A. Zisserman, "Read and attend: Temporal localisation in sign language videos," Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit., pp. 16852–16861, 2021, doi: 10.1109/CVPR46437.2021.01658.

[65] M. De Sisto, D. Shterionov, I. Murtagh, M. Vermeerbergen, and L. Leeson, "Defining meaningful units. Challenges in sign segmentation and segment-meaning mapping," Proc. 1st Int. Work. Autom. Transl. Signed Spok. Lang. AT4SSL 2021, pp. 98–103, 2021.

[66] L. Doc\'\io-Fernández et al., "{LSE}_{UVIGO}: A Multi-source Database for {S}panish {S}ign {L}anguage Recognition," 2020. Accessed: Mar. 03, 2021. [Online]. Available: https://www.aclweb.org/anthology/2020.signlang-1.8.

[67] O. Koller, S. Zargaran, and H. Ney, "Re-sign: Re-aligned end-to-end sequence modelling with deep recurrent CNN-HMMs," Proc. - 30th IEEE Conf. Comput. Vis. Pattern Recognition, CVPR 2017, vol. 2017-Janua, pp. 3416–3424, 2017, doi: 10.1109/CVPR.2017.364.

[68] W. Kay et al., “The Kinetics Human Action Video Dataset,” 2017, [Online]. Available: http://arxiv.org/abs/1705.06950.
[69] S. Ji, W. Xu, M. Yang, and K. Yu, “3D Convolutional neural networks for human action recognition,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 35, no. 1, pp. 221–231, 2013, doi: 10.1109/TPAMI.2012.59.
[70] A. Vaswani et al., “Attention is all you need,” Adv. Neural Inf. Process. Syst., vol. 2017-Decem, no. Nips, pp. 5999–6009, 2017.
[71] N. Awad, N. Mallik, and F. Hutter, “DEHB: Evolutionary Hyberband for Scalable, Robust and Efficient Hyperparameter Optimization,” IJCAI Int. Jt. Conf. Artif. Intell., no. 3, pp. 2147–2153, 2021, doi: 10.24963/ijcai.2021/296.
[72] J. Wang, Z. Wang, S. Zhuang, and H. Wang, “Cross-Enhancement Transformer for Action Segmentation,” 2022, [Online]. Available: http://arxiv.org/abs/2205.09445.
[73] D. Du, B. Su, Y. Li, Z. Qi, L. Si, and Y. Shan, “Efficient U-Transformer with Boundary-Aware Loss for Action Segmentation,” 2022, [Online]. Available: http://arxiv.org/abs/2205.13425.
[74] N. Behrmann, S. Alireza Golestaneh, Z. Kolter, J. Gall, and M. Noroozi, “Unified Fully and Timestamp Supervised Temporal Action Segmentation via Sequence to Sequence Translation,” Eccv, pp. 1–24, 2022.
[75] L. Chen, M. Li, Y. Duan, J. Zhou, and J. Lu, “Uncertainty-Aware Representation Learning for Action Segmentation,” pp. 820–826, 2022, doi: 10.24963/ijcai.2022/115.
[76] “British Sign Language Corpus Project: A corpus of digital video data and annotations of British Sign Language.” https://bslcorpusproject.org/ (accessed Sep. 02, 2022).
[77] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, “Learning Internal Representations by Error Propagation,” Readings Cogn. Sci. A Perspect. from Psychol. Artif. Intell., no. V, pp. 399–421, 2013, doi: 10.1016/B978-1-4832-1446-7.50035-2.
[78] S. Hochreiter and J. Schmidhuber, “Long Short-Term Memory,” Neural Comput., vol. 9, no. 8, pp. 1735–1780, 1997, doi: 10.1162/neco.1997.9.8.1735.
[79] M. Gonzalez, P. Joly, and C. Collet, “Recognition in the Context of Sign Language Annotation,” 2012.
[80] A. Braffort, L. Boutora, A. Braffort, and L. Boutora, “Défi d ’ annotation DEGELS2012 : la segmentation To cite this version : HAL Id : hal-01803808,” pp. 1–8, 2018.
[81] M. Gonzalez, “Un système de segmentation automatique de gestes appliqué à la Langue des Signes,” JEP-TALN-RECITAL 2012, Atelier DEGELS 2012 Défi GEste Lang. des Signes, pp. 93–98, 2012, [Online]. Available: http://degels2012.limsi.fr/actes/pdf/degels2012_7.pdf.
[82] F. Lefebvre-Albaret and J. Segouat, “Influence de la segmentation temporelle sur la caractérisation de signes,” JEP-TALN-RECITAL 2012, Atelier DEGELS 2012 Défi GEste Lang. des Signes, pp. 73–83, 2012, [Online]. Available: http://degels2012.limsi.fr/actes/pdf/degels2012_5.pdf.
[83] A. Millet and I. Estève, “Segmenter et annoter le discours d’un locuteur de LSF : permanence formelle et variablité fonctionnelle des unités,” JEP-TALN-RECITAL 2012, Atelier DEGELS 2012 Défi GEste Lang. des Signes, pp. 57–72, 2012, [Online]. Available: http://degels2012.limsi.fr/actes/pdf/degels2012_4.pdf.
[84] V. Viitaniemi, T. Jantunen, L. Savolainen, M. Karppa, and J. Laaksonen, “S-pot - A benchmark in spotting signs within continuous signing,” Proc. 9th Int. Conf. Lang. Resour. Eval. Lr. 2014, vol. 30, no. 2, pp. 1892–1897, 2014.
[85] N. Adaloglou et al., “A Comprehensive Study on Deep Learning-Based Methods for Sign Language Recognition,” IEEE Trans. Multimed., vol. 24, pp. 1750–1762, 2022, doi: 10.1109/TMM.2021.3070438.

Appendix 1.Table Results

Decoders	K Iter	mF1B	mF1S
1	1	64.69	55.46
1	2	66.47	56.16
1	3	63.92	54.97
1	4	64.45	55.39
1	5	64.38	53.54
2	1	65.95	56.09
2	2	65.98	53.62
2	3	65.74	56.35
2	4	64.85	54.82
2	5	64.57	55.49
3	1	61.99	51
3	2	66.36	57.26
3	3	66.87	56.88
3	4	61.15	50.92
3	5	64.32	54.06
4	1	61.23	51.25
4	2	65.13	54.97
4	3	**68.38**	**58.05**
4	4	64.7	54.3
4	5	61.65	52.96

Table 11: Results depending on # decoders when # blocks = 10 bz=1 and lr = 0.005 in epoch = 50

Learning rate	K Iter	mF1B	mF1S
0.0005	1	65.28	48.97
0.0005	2	61.99	51
0.0005	3	66.36	57.26
0.0005	4	66.87	56.88
0.0005	5	61.15	50.92
5·10-5	1	64.32	54.06
5·10-5	2	64.05	53.18
5·10-5	3	66.01	54.28
5·10-5	4	64.83	53.74
5·10-5	5	65.92	54.07
5·10-6	1	66.59	55.56
5·10-6	2	61.65	49.03
5·10-6	3	60.84	48.6
5·10-6	4	61.2	47.92
5·10-6	5	61.02	48.86

Table 12: Results depending on # of blocks when # decoders = 10, bz=1 and lr = 0.005 in epoch = 50

Blocks	K Iter	mF1B	mF1S
6	1	61.99	51
6	2	**68.48**	57.87
6	3	66.87	56.88
6	4	63.26	52.62
6	5	64.32	54.06
7	1	**68.58**	57.66
7	2	66.46	57
7	3	65.28	55.85
7	4	68.13	**58.47**
7	5	64.75	54.37
8	1	65.96	56.65
8	2	66.62	55.39
8	3	65.87	55.06
8	4	53.11	40.57
8	5	66.73	56.92
9	1	65.5	55.8
9	2	**68.55**	57.73
9	3	66.66	57.32
9	4	60.78	49.5
9	5	66.73	57.06
10	1	61.99	51
10	2	66.36	57.26
10	3	66.87	56.88
10	4	61.15	50.92
10	5	64.32	54.06
11	1	63.97	56.3
11	2	65.15	55.35
11	3	63.62	53.5
11	4	65.78	55.8
11	5	65.11	53.83

Table 13: Results depending on # of blocks when # decoders = 10, bz=1 and lr = 0.005 in epoch = 50

Appendix 2.Transformers.

In this appendix, the foundations of transformers and particularly ASFormer would be explained. First, the need of transformer would be explained. Then transformer architecture will be shown. Finally, ASFormer modification would be explained

9.1. Transformers:

While in previous methods from literature like RNN[77] or LSTM [78], the trained model considered the memory context, this was limited, and an improvement was needed. In this context, Transformers [70] were introduced as a new paradigm with two important improvements. First, it is based on the concept of attention, which has an infinite reference window, considering all the previous data. Second, their architecture allowed an easy parallelization of processes as we would see later.

Transformer's architecture is based in an Encoder-Decoder structure, as we could see in Figure 9. The encoder is responsible of mapping the input sequence into an abstract representation, while the decoder takes it and generates the output sequentially, feeding it continuously with the generated results.

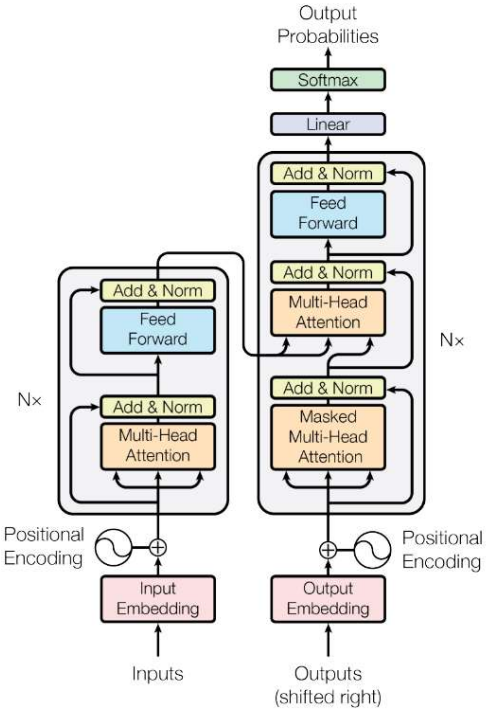


Figure 9: Transformer architecture taken from [70]

Before entering the proper encoder, data must pass two steps, the input embedding and positional encoding. In the **Input embedding** the data is transformed into an "embedding space" transforming each input into a vector that could be digested by the encoder, with the particularity that similar data would be located in nearby places as representing "meaning". For example, "synonyms" would share the same vector. However, as this is not enough to represent the location, an additional step is added, the **Positional encoding**, which adds context based on the position of the data by using the following formulas.

$$\text{Odd positions: } \mathrm{PE}(\mathrm{pos}, 2\mathrm{i}+1) = \cos\left(\frac{\mathrm{pos}}{1000^{\frac{2\mathrm{i}}{\mathrm{dmodel}}}}\right)$$

$$\text{Even positions: } \mathrm{PE}(\mathrm{pos}, 2\mathrm{i}) = \sin\left(\frac{\mathrm{pos}}{1000^{\frac{2\mathrm{i}}{\mathrm{dmodel}}}}\right)$$

Once the data has been mapped to the embedding space considering both meaning and position, then it enters the **encoder** formed by N layers of a Multi-Head Attention sublayer and a Feed-Forward encoder. The concept is that each of the N layers learns a different representation to boost the performance.

In the attention layer, the data is divided into three: The **query**, the **key** and the **value** and recombined as in the following formula. The dot product between query and the key generates a matrix that determines the relationship between different words of the sequence. This matrix is then divided by the square root of the dimension and then a SoftMax function is applied for depressing the lower scores and increasing the highest ones for enhancing the difference. After this, it is multiplied by the value

$$Attention(Q,K,V) = softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$

The multi-head attention takes this basic model and repeats it several times (8) for taking the weighted average.

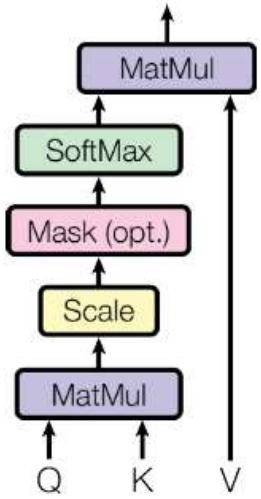


Figure 10: Scaled Dot-Product Attention

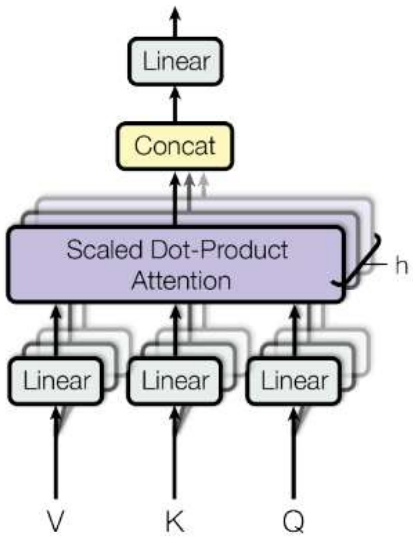


Figure 11: Multi-Head attention.

After passing through the attention layer, the data enters a **Feed Forward network** which transforms the attention vector preparing it from entering the following layer. In this step we could benefit from the use of parallelization. Finally, there are residual connections between layers, and they are also followed by a normalization layer that helps stabilize the network.

This info is passed into the decoder which is formed by the same modules of the Encoder: it uses the embedding and positional encoding to transform the output into vectors and then passes to a multi-header attention layer with a variation. As we do not know which would be the future values of the output, it is masked, and the upper part of the matrix should be 0.

Then, the result of this module, united with the result of the decoder is feed into a new multi-head attention layer. The result is feed into a feed forward layer as it was in the encoder and then passes to a SoftMax layer that calculates the probability distribution. The final output would be the most probable "word".

9.2. ASFormer

ASFormer [7] is a transformer modified for the problem of Action Segmentation. As pointed in their paper, there are three main limitations of the original transformer in the Action Segmentation Task:

1. Datasets are small, and the size of the training set is small. In consequence there is a **lack of inductive biases** that makes it difficult for the model to be learnt.
2. Videos are usually very long which makes the transformer **hard to form an effective representation.**
3. Finally, original encoder-decoder architecture **does not meet the refinement demand of action segmentation task** where temporal relationships between actions are important (e.g., the action after taking a bottle of water and pouring it inside a glass usually is drinking the water). In that sense, some works before ASFormer apply additional TCNs or GCNs over the initial prediction to enhance it.

To avoid these problems, the following modifications to the original transformer are made.

- Local inductive bias is strengthened by adding temporal convolutions in each layer. In the encoder and decoders, the feed forward layer is not a pointwise fully connected layer as in the original, but a dilated temporal convolution
- Since this temporal convolution has already the ability to model the positional relationships between actions, position encoding is redundant, and it is shown that removing it improves the performance.
- In the encoder, a single-head self-attention layer is used instead of the multiple-head one.
- As the model has difficulties to learn meaningful locations, a pre-defined hierarchical representation pattern is set in the self-attention

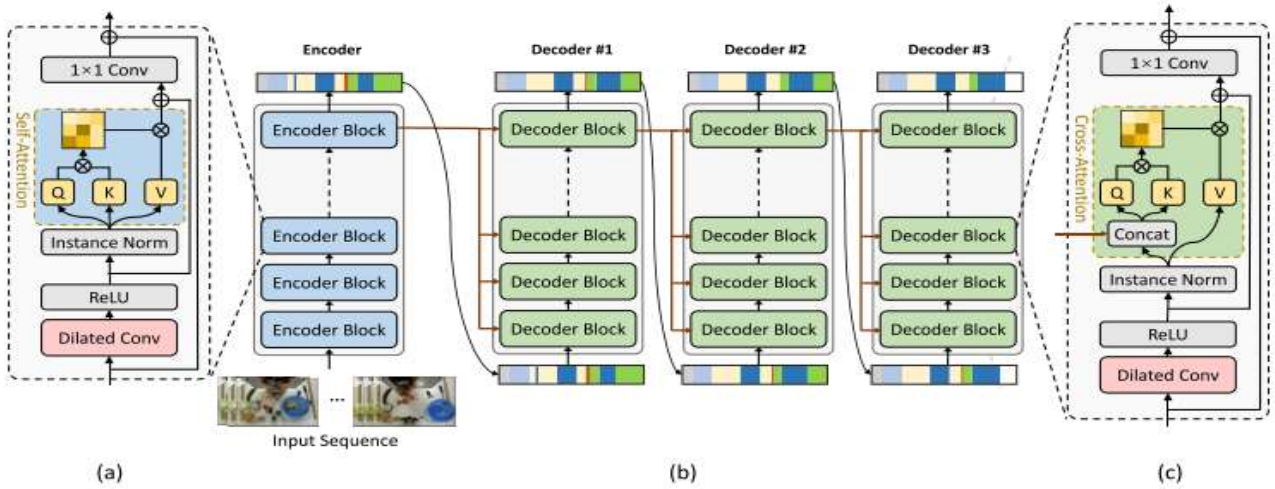


Figure 12: ASFormer architecture taken from [7]

layer. This, way, the low-level self-attention layers focus on the local relations while the high-level ones gradually capture longer dependencies. An improvement of this change is a reduction of memory from original transformer from $J \cdot T \cdot T$ to $((2 - \varepsilon) \cdot 2^J \cdot T)$ where J is the number of blocks, T is the length of the video and ε is a small number

- Finally, to improve the refinement demand of the action segmentation task, there is a cross-attention mechanism added in the decoder: every position in the encoder attend all positions in the refinement process for avoiding the disturbance of the encoder to the learned feature space in the refinement phase

ASFormer performed better in the Action Segmentation problem than other architectures like MS-TCN [6], or ASRF [50]