Contributions for Sign Language translation: active signer detection and isolated sign recognition

Coline Petit-Jean, supervised by Hannah Bull and Michèle Gouiffès

June 2021

Introduction

This report presents the work I did during my M2 internship. This internship took place at the *Laboratoire Interdisciplinaire des Sciences du Numérique* (LISN) in Orsay, from march 2021 to july 2021. I was an intern the team *Architectures et Modèles pour l'Interaction* (AMI) of the lab, I was supervised by Michèle Gouiffès. I was also supervised by Hannah Bull from the team *Information Langues Écrites et Signées* (ILES). At a different scale, I was a part of the group *Modélisation & Traitement Automatique des Langues des Signes* (M&TALS). This group works on the following thematics: Sign Language Recognition & Computer Vision, Sign Language Modeling and Sign Language Avatars & Computer Graphics.

During the internship, I have worked on the thematic Sign Language Recognition & Computer Vision with Michèle, Hannah, and Ambroise Mopendza. The latter was an intern during the same period as I was. The first month of the internship was dedicated to my participation in the 2021 Looking at People Large Scale Signer Independent Isolated SLR CVPR Challenge [1] that was held along with the Computer Vision and Pattern Recognition conference. This was an occasion for me to get a first insight into both sign language and neural networks used for action recognition. After that, Hannah, Ambroise and I worked on videos provided by the bilingual french-french sign language media *Média'Pi!*. The goal was to publish it as a new dataset for French Sign Language (LSF) translation studies, along with some sign language processing tasks performed on it.

This report is organized as follows. The first section introduces elements of Translation Networks architecture and Sign Language Translation that are relevant to what comes next. The second section is about the data challenge I worked on. Eventually the last section presents the preprocessing and the tasks we performed on the videos of *Média'Pi!*.

1 Generalities about Translation Networks and Sign Language Processing

This section presents elements of description of translation networks and of sign language processing that are relevant to what comes next in the report. That is to say some background about Neural Networks used for translation and image processing and about corpuses used to work on Sign Language processing.

1.1 Corpuses in Sign Languages

In order to work on Sign Language Processing, as for any other language, one need to have some corpuses. However, creating a corpus for a sign language is much more a challenge than for a spoken language. Indeed, sign languages have no written form. This implies that video is the main way of captioning it. Moreover, death communities are smaller than speaking communities and are less visible, therefore there is much less data than there is for spoken languages. Eventually most of the content in sign languages available is not "natural" sign language but interpreting from spoken languages (this is for example the signers who interpret during the official speeches on TV). Therefore this content may lack some of the grammatical and lexical specificity of sign languages. It is also important to have several signers with diversity of gender, age, clothing, skin tone, body proportions, disability, fluency, background scenery, lighting conditions ... A more complete description of what corpus for Sign Language processing are is available in [2]

Two of the corpuses currently used for sign language translation are for example RTWH-Phoenix-2014 [3] and BSL-1k [4][5]. RTWH-Phoenix-2014 has a vocabulary of 1080 signs in continous German Sign Language, it features nine signers and totalizes more than ten hours of video. The recognition on this corpus is the standard benchmark task of the field. More recently, in 2020, the corpus BSL-1k was published. It features 40 signer using British Sign Language with a vocabulary of about 1000 signs. It totalizes 1000 hours of video. Both of these corpuses are made of interpretation from a spoken language, additionnally, they features signers shot in studio and in "real life conditions" of background and lightening.

The AUTSL corpus [1] that was used for the challenge contains 38,336 videos of 226 isolated signs performed by 43 different signers. Each video lasts around 2s. One particularity of this data is the diversity of the backgrounds and lighting conditions of the videos, which are close to real-life

(a) Signer #1 (b) Signer #11 (c) Signer #16

(d) Signer #18 (e) Signer #25 (f) Signer #35

Figure 1: The 6 signers from the validation set.

situations. The signers are filmed in indoor and outdoor settings at various distances to the camera. Moreover, some of the backgrounds of the signers are dynamic. Nevertheless, the signers are all facing the camera and are cropped to be in the centre of the video. The training set is relatively balanced, with approximately the same number of videos for each sign class.

The validation and test sets do not contain the same signers as the training set. This allows us to evaluate the robustness of models to unseen signers. However, models trained on this data are possibly not very robust to different pose angles (e.g. the signer not facing the camera). Figure 1 shows example frames from the validation set.

Eventually, the Media'Pi! corpus I worked on includes 368 videos. They last from some seconds for the shortest of them to several minutes for the longest. The corpus totalizes 27 hours of video. As mentioned before, Média'Pi! provides a journalistic content in French Sign Language. Therefore the videos show a wide variety of scenes. The simpliest of all to handle by computer would be a presenter, alone on the frame, facing the camera, in close or medium shot. The videos also show complex scenes, for example, several people outside, among who some are signing and some are not. The person signing may also move or not be totally facing the camera and may appear in long shots. This corpus is not published yet but videos showing the skeleton of the people appearing in the video were published in [6] and may be used for sign language processing purposes. They are available here

1.2 Method inspired from classic traduction tasks

Even though sign languages are very different from spoken languages, some of the methods used for spoken language translation apply for sign language translation. First, given embedding of the two languages one want to associate by the translation, the same networks can be used independently of the nature — spoken or signed — of the languages. Second, when one want to translate a sign language into a spoken language, they still need to embed the spoken language.

Transformers. The best state of the art model for translation are networks called transformers [7]. As many models for translation, they follow a decoder and encoder principle: an input sequence of

symbol representations $x = (x_1, \ldots, x_n)$, is mapped to a continuous representation $z = (z_1, \ldots, z_n)$ by the encoder. Then, given z, the decoder generates an output sequence of symbols $y = (y_1, \ldots, y_m)$.. The transformer combines this encoder-decoder architecture with attention mechanism. An attention function can be described as a learned map that associates a set of key-value pairs and a query to an output. The output is computes as a weighted sum of the values. For each value, the weight depends of the result of a compatibility function applied to the query and the corresponding key.

Words embedding. The embedding used for the spoken language in sign language translation are the same as for spoken language translation. Currently, the state of the art results are achieved by embedding such as Bidirectional Encoder Representation for Transformers [8] (BERT) which is an network that takes into account both the right and the left context of a word in a sentence to create the embedding.

1.3 Embedding of the video input

In the classical translation tasks, the inputs are either text or audio files. Conversely, in sign language translation, some of the inputs are video files. Then even though one can use the same kind of network as for spoken language translation tasks, the embedding of the input data needs to be different.

The embedding for sign language videos are largely inspired from the task Action Recognition. Indeed, given a video featuring an action, Action Recognition consists in identifying the action among a finite list of actions given with the dataset one is working on. For example, the actions for the dataset "something-something" are throwing something, catching something, ... From this point of view, sign language translation can be seen as an instance of Action Recognition where the actions are some lexical units of the sign language considered. Therefore the methods used for Action Recognition apply to Sign Language Translation. A complete study of the networks used for Action Recognition is available in [9].

Even tough there are a lot of different embedding, one can distinguish two main kinds among them: those who are made only of the features got by applying a convolutionnal neural network to the videos, and those that use CNN but also add to it features that are less convenient to compute. In both cases, the CNN used can be a two-dimensional spatial convolutional network applied to each frame separately followed by a Long Short Term Memory network [10] that aggregate information through time. However, the state of the art results are achieved by networks using directly three-dimensional convolutions over both time and space such as I3D [11]. In the second case, the additional features depend on the kind of action that needs to be identified. It can be for example more information on the position of the hands if the actions to identified implies fine movement of the hands. Conversely, an action that is more global, as for example identifying a sport practiced in a video, may need more information about background. In the case of Sign Language Translation, the additional features are typically a complement of information about face expressions, hands movement are body pose, because these elements are relevant to analyse the signed speech. They are computed thanks to body pose estimation algorithms such as OpenPose [12] or HRNet [13]. These algorithms take the video as input, and output a list of persons, with for each person a list of positions of its keypoints (hips, knees, shoulders ...)

2 Data Challenge

The first month of my internship was dedicated to my participation in the "2021 Looking at People Large Scale Signer Independent Isolated SLR CVPR Challenge". It allowed me to get more familiar with Sign Language Processing and with neural networks used for Action Recognition. In this section, I present the work I did on the challenge.

2.1 Model

In sign languages, different articulators (hands, eyes, mouth, body pose etc.) are used in parallel at different time scales: hands moving in a particular direction, eyes blinking rapidly or shoulders moving subtly. Therefore, models which take into account different temporal scales could perform well for sign language by learning longer and shorter movements. Moreover, such models pre-trained on action recognition datasets can provide a training boost for learning classes of signs.

	Top-1	Top-3	Top-5
I3D Slow ResNet50 [15]	10%	21%	28%
I3D ResNet50 [11]	12%	23%	30%
SlowFast 4x16 ResNet50 [15]	20%	37%	46%
TPN ResNet50 [16]	**23%**	**43%**	**54%**
Baseline	49%	69%	76%

Table 1: We extract the final fully-connected layer from various networks pre-trained on Kinetics-400 and then add an additional fully-connected layer with softmax activation to classify the 226 signs. We choose the best method based on performance on a subset of the training set (Signers #40, #41 and #42).

We justify the choice of pre-trained weights in Section 2.1.1, justify the choice of a model with different temporal scales in Section 2.1.2, describe in detail our chosen model in Section 2.1.3 and provide implementation details in Section 2.1.4.

2.1.1 Model initialisation

The Kinetics-400 dataset [14] contains 306,245 short video clips of 400 human action classes, such as shaking hands, brushing hair and salsa dancing. These actions are performed by people in various contexts, with different backgrounds and at different scales, similarly to the AUTSL dataset. Thus, in order to benefit from additional data, we use weights of the models initialised on Kinetics-400 and train these weights using the AUTSL data.

2.1.2 Choosing the model architecture

Table 1 compares the performance of out-of-the-box features of action recognition models trained on Kinetics-400 for this task. No fine-tuning of these networks is performed, but a linear layer is added on top of the extracted features in order to output scores for the 226 sign classes. Training such a small network takes about 10 minutes for each model.

The four networks from which we extract the Kinetics-400 features are I3D Slow ResNet50 [15], I3D ResNet50 [11], SlowFast ResNet50 [15] and TPN ResNet50 [16]. SlowFast ResNet50 and TPN ResNet50 both work on two temporal streams down-sampling the input frames at different rates. On the other hand, I3D Slow ResNet50 and I3D ResNet50 only have one stream, where the difference between these two networks is that in the former, the input frames are down-sampled at a slow rate.

The performance of the two networks using two streams are significantly better than the two other networks. This suggests that using different temporal scales is suited for the problem of sign recognition. Nevertheless, the fact that the results are quite low compared to the baseline highlights the fact that using only extracted features without any training on AUTSL is insufficient.

The best performing pre-trained features are when using the TPN ResNet 50 model described in [16]. We thus choose this model with initialisation on Kinetics-400 and train it using the AUTSL dataset.

2.1.3 TPN ResNet50

Our final model is the TPN ResNet50, presented in [16] and illustrated on Figure 2, which is composed of an Inflated ResNet50 SlowFast [15] as a 3D backbone. ResNet50 SlowFast is a convolutional network operating on two different framerates, and is therefore suited to detect long and short time dependencies in videos. This trait is amplified by TPN, which combines features with different depths in the network and therefore different temporal receptive fields.

More precisely, TPN collects a multi-depth pyramid of M hierarchical features $\{F_1 \dots F_M\}$ with increasing depth, where these features have sizes $\{C_1 \times T_1 \times W_1 \times H_1, \dots, C_M \times T_M \times W_M \times H_M\}$, where C_i is the number of channels, T_i is time, W_i is width and H_i is height. After collecting these features, TPN performs a Spatial Semantic Modulation consisting of convolutions with a specific stride for each feature. This aligns the spatial semantics of the features. TPN also performs a Temporal Rate Modulation by down-sampling each feature at a different rate. This temporal modulation allows a better control on the relative differences of temporal scales between features. Eventually, the features

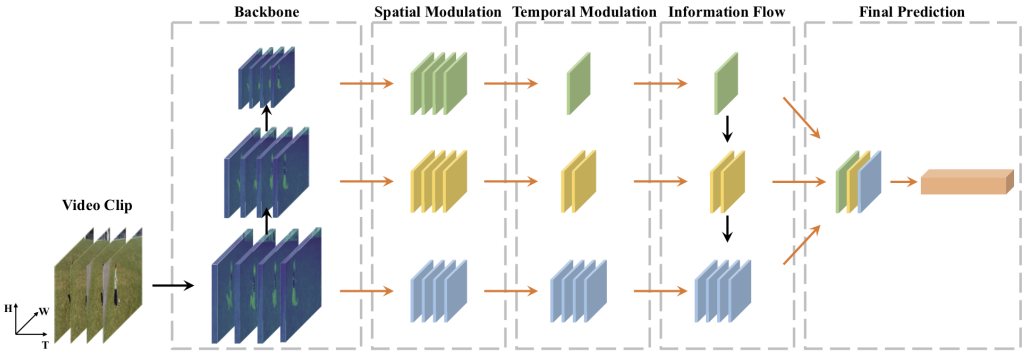

Figure 2: Temporal Pyramid Network

	Validation			Test
	Top-1	Top-3	Top-5	Top-1
Baseline	49.22	68.89	75.78	49.23
TPN ResNet50	**92.85**	**98.33**	**99.19**	**93.75**
Challenge winner	-	-	-	98.42

Table 2: Although our results are about 5% lower than the winning strategy on the test set, we significantly improve upon the baseline.

from which the final predictions are made are aggregated using parallel Top-down and Bottom-up flows on features of increasing depth.

2.1.4 Implementation

We train the model using SGD optimisation with a learning rate of 0.025, a weight decay of 0.0001 and a momentum of 0.9. Due to memory constraints, the batch size was initially set to 16. After 90 epochs, we change the batch size to 8 and the learning rate to 0.0001. As a data augmentation step, we implement random horizontal flips, so that the model learns to recognise signs performed by both left and right handed signers. We sample 32 frames with stride 2 for each video sequence, padding at the end with additional frames from the start of the video if necessary. We use 4 Nvidia RTX 2080 GPU with 8GB of memory each and train the model for around 3 days. We compute the Top-1, Top-3 and Top-5 accuracy metrics to evaluate the performance of our model.

2.2 Results

2.2.1 Key Results

Table 2 shows our results on the validation set and on the test set. On the validation set, we predict 92.85% signs correctly. The correct sign can be found amongst the top 3 predicted categories in 98.33% of cases and amongst the top 5 predicted categories in 99.19% of cases. On the test set, we obtain a score of 93.75%, in comparison to the winning entry, which achieved a score of 98.42%.

Figure 3 shows the performance of our model on the validation set at different epochs. We notice two jumps in performance: firstly, by adding our supplementary validation set back into the training data at epoch 70 and secondly, by reducing the learning rate and the batch size at epoch 90.

On the validation set, correct predictions are given a high score, with an average of 0.985 and standard deviation of 0.069 for the top predicted class. For incorrect predictions, the average score of the top predicted class is 0.800 with a standard deviation of 0.222. This shows that our model tends not to falsely predict incorrect classes with very high certitude.

Performance of our model is related to the size of the training data used. Table 3 shows the results of our model when trained on around 50% and 75% of the training data. Perhaps with additional training data, our model would continue to improve.

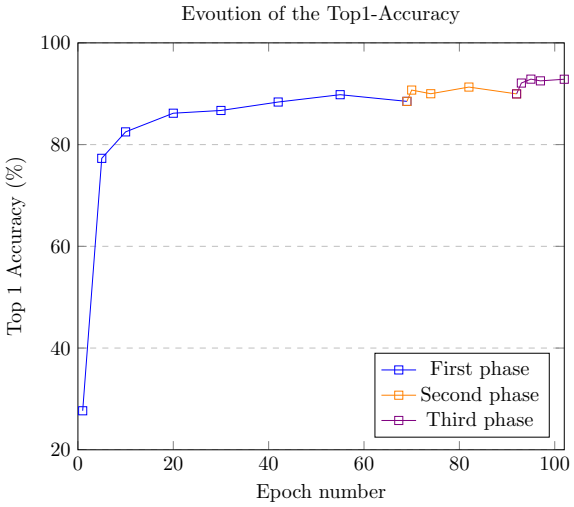

Figure 3: Top-1% accuracy on the validation data at selected epochs. During the first phase, we exclude part of the training data as a supplementary validation set (Signers #40, #41 and #42). During the second phase, we add this data back in to the training set. During the third phase, we reduce the learning rate from 0.025 to 0.0001 and the batch size from 16 to 8.

	Top-1	Top-3	Top-5
Baseline	49.22	68.89	75.78
14000 Samples	82.98	92.55	94.55
21000 Samples	88.59	96.20	97.74
28144 Samples	**92.85**	**98.33**	**99.19**

Table 3: Performance of the model on the validation set using different sizes of data.

Signer	Error Rate
#1	6.7%
#11	4.5%
#16	20.6%
#18	5.4%
#25	5.7%
#35	3.7%

Table 4: The errors for Signer #16 are much higher than for the other signers in the validation set (shown in Figure 1). This is possibly due to the fact that this signer is standing and further from the camera compared to the other signers in the validation set.

(a) Truth: 178, Prediction: 178, Signer #1 and Sample #318

(b) Truth: 178, Prediction: 175, Signer #16 and Sample #44

Figure 4: The same sign performed by two different people, one correctly and the other incorrectly predicted by our model.

2.2.2 Analysis

In order to understand the errors in our predictions, we look at some of the failure cases in the validation set. The signers in the validation set are shown in Figure 1. All of the signers in this validation set are sitting down, with the exception of Signer #16, who is standing up and is further away from the camera. It seems as though our model is not robust to this kind of variation, as in the validation set, the error rate for Signer #16 is much higher than for the other signers (Table 4). An example of where a sign is incorrectly predicted for Signer #16 but correctly predicted for another signer is provided in Figure 4.

Some of the sign categories are very similar. This is the case for the sign labelled 165 and the sign labeled 47, as shown in Figure 5. In the former sign category, shown in Figure 5a, the signer taps the wrist with only the index finger, whereas in the latter sign category, shown in Figure 5b, the signer taps the wrist with the index finger and the thumb. This is a very subtle difference not picked up by our model.

3 Mediapi Corpus: cleaning and active signer detection

During the remaining part of the internship, I worked with Hannah and Ambroise on videos coming from the bilingual LSF french media "Média'Pi!". This was an occasion to apply and adapt the methods used for the challenge to a more challenging problem: continous sign language processing in videos showing several potential signers. Indeed we could use the same reprentation model with networks such as I3D, but before that, a careful pre-processing was needed. We aimed at publishing some of their videos as a corpus in LSF for Sign Language Translation. This would be a great advance in french sign language processing because "Média'Pi!" offers a content that has both characteristics:

1. It is "natural sign language": the content is not interpreted from french or an other language.
2. It is wide: about one video per day is published. Additionnally, these videos vary from news to report and therefore offer different backgrounds and framing.

(a) Truth: 165, Prediction: 165, Signer #35 and Sample #64

(b) Truth: 47, Prediction: 165, Signer #35 and Sample #301

Figure 5: Two similar signs with two different labels, one correctly and the other incorrectly predicted by our model.

And there is no such corpus already available for french sign language except the work of Hannah Bull in [6] that have made available the keypoints of the people signing in "Média'Pi!" videos. In this section I present the preprocessing Ambroise and I applied to these videos and the Sign Language Translation task we tried on it: active signer detection.

3.1 Cleaning the data

The data we used comes from the Média'Pi! YouTube channel. It is composed for each video of an audiovisual file with the extension ".mp4" and a subtitle file with the extension ".fr.vtt".

Extraction of the people.

In order to carry out Sign Language Processing, one may want to have videos featuring only the signer framed with a medium shot. Therefore Ambroise used the network HRNet on our videos to extract the people present on each frame and each video.

Tracking. The output of HRNet is for each frame, a prediction of where there are people on it, and for each person detected, an estimation of the position of its keypoints. For each keypoint, we have (x, y, p) where x, y is an estimated position, and p the probability associated to this prediction. The tracking step aims at identifying the persons detected in one frame with those detected in the next frame. Doing so, we are able to build several temporal sequences, each representing one person with its keypoints across several consecutive frames.

Selection of the relevant potential signers The keypoint detection by HRNet was quite performant (introduire baseline plus haut) on our corpus. That is to say that it detected accurately most of the visible keypoints of the people appearing on the frame. Then the tracking allowed to detect a high number of temporal sequences (insérer chiffres ici encore). However it was not relevant to treat all of this sequences in order to perform active signer detection and to be more efficient. Therefore, we eliminated the following type of people:

- people having their back facing the camera
- people appearing too small with respect to the size of the frame
- people detected with a probability too low
- people detected on an interval of time too short

Extraction of the subvideos

Using the temporal sequences of people that we get with the previous step, Ambroise extracted subvideos cropping the people by estimating a bounding box of the person on each frame of the sequence, and smoothing the position of the bounding box over the sequence.

	percentage of incorrect frames	mean recall	mean precision	mean f1
validation set	0.10	0.65	0.74	0.69

3.2 Active Signer Detection

Once the set of cropped videos featuring all the people potentially signing in a video was constituted, I tried to perform Active Signer Detection on it. This section present this problem, the approach I used to perform it on our dataset and the results I get.

3.2.1 Problem formulation

Given a sequence of video frames of any kind, the goal of Active Signer Detection is to identify on the one hand if there are people signing in the video, and on the other hand, at which moment of the video the people are signing. Given the good performances achieved in the field of people detection in videos, the problem can be reduced to videos that are cropped to show only one person, the goal is then to predict the moment of the video when at which this person is signing. Therefore the problem is the following: given a sequence of frames $x = (x_1, \ldots, x_N)$ showing one person, predict a vector of corresponding labels $y = (y_1, \ldots, y_N) \in \{0, 1\}^N$ where 0 means that the person is not signing and one means that the person is signing. Note that we did not need to define explicitly what is a person signing because we used videos already annotated with subtitles.

3.2.2 Creation of the training set

We trained a supervised model. Therefore, a training set constituted of many videos associated each to a sequence of zeros and ones was needed. In order to create this set, given the cropped video corresponding to the time interval of one subtitle, the difficulty was to associate the subtitle to the person that was actually signing. The decision was done based on the following hand-crafted feature: the average height of the hands with respect to the shoulders, normalized by the size of the person. Indeed the person signing is the more likely to have their hands high during all the timelapse of the subtitle. Thanks to this association, I was able to label each video with a sequence of zeros and ones: for each frame the label corresponding was 1 if there was a subtitle at this moment of the video and if this subtitle was associated to the person featured on the video.

Note that even though the keypoints of the people are needed to create the training set, the training of the model on this set will allow to perform Active Signer Detection without pre-extracted keypoints. Therefore the keypoint extraction is not a part of the model.

3.2.3 Model

Following the approach adopted in [17] for sign language segmentation, we used the convolutional I3D architecture [11] and coupled it with the Multi-Stage Temporal Convolutional Network (MS-TCN) introduced in [18]. MS-TCN is composed of several stages named Single-Stage TCN (SS-TCN), that are run sequentially. it is illustrates on figure **??**. SS-TCN is a network that takes as input some features for each of the frames and that performs a classification by using a sequence of stacks of 1 dimensional convolutions with strides bigger and bigger: the convolutions of the first layer have stride 1, those of the second have stride 2, those of the third have stride 4 and so on, ending with the tenth layer whose convolutions have size 512. Running sequentially several SS-TCN means that the first SS-TCN has as input the sequence of features of each frame, and after that, the input features of each stage are the predictions of the precedent. As mentioned at the beginning of the paragraph, in our case the features used to feed MS-TCN are for each frame the outputs of I3D run on a stack of 16 frames around the frame. The ground truth used for training is a set of sequences of frames, each associated with a sequence of zeros and ones of the same size, identifying the moments when a person is signing. Its construction is detailed in the preceding section.

3.2.4 Results

The results of the application of MS-TCN to our *Média'Pi!* corpus are given in Table **??**

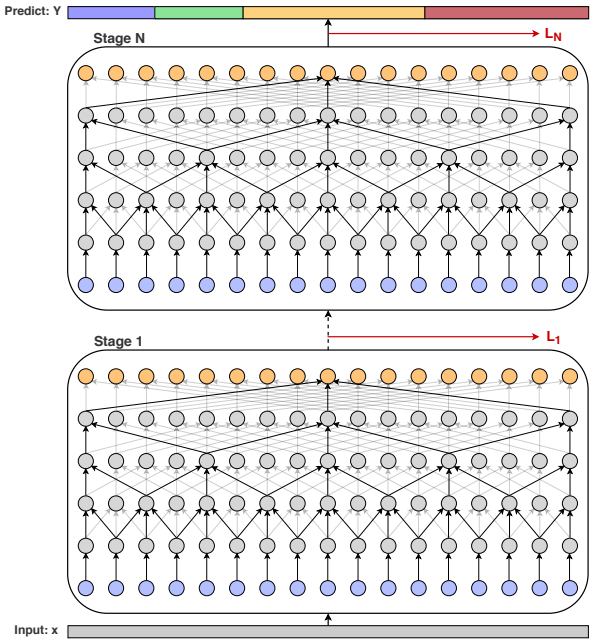

Figure 6: MS-TCN

Conclusion

To conclude, this internship allowed me to explore both machine learning and deaf culture. I participated in a data challenge and was able to produce a significant improvement with respect to the baseline using a network that had never been used to process sign language before. After that I worked with other members of the team on videos from the bilingual media *Média'Pi!*. We pre-processed the videos and began to apply classic translation tasks to it in order to publish it as a new corpus for sign language processing, that would be grammatically and syntactically rich and in "natural" sign language.

Acknowledgments

The context of my internship was really special this year for both external and personal issues. Due to the Covid-19 pandemic, the internship took place alternatively remotely and in presential. I have also been sick for two months at the end of the internship. For all of these reasons, it was quite challenging for me to hand my report in this year and I am proud I could do so. However this would not have been possible without the help of many persons. Therefore I would like to thank Michèle for her kindness, Ambroise for being such a joyful co-intern and Hannah for supervising me during the internship. I would like to thank all of my flatmates from *la cabane* and my friends for supporting me through this challenging time. Eventually I would like to thank Noémie Fanget and Sandrine Tonadre of the medical service of the *ENS de Lyon* and the team of the *UPC* of Édouard Herriot Hospital for supporting me medically and psychologically with a lot of humanity through the illness.

References

[1] 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.

[2] Danielle Bragg, Oscar Koller, Mary Bellard, Larwan Berke, Patrick Boudreault, Annelies Braffort, Naomi Caselli, Matt Huenerfauth, Hernisa Kacorri, Tessa Verhoef, et al. Sign language recog-

nition, generation, and translation: An interdisciplinary perspective. In *The 21st international ACM SIGACCESS conference on computers and accessibility*, pages 16–31, 2019.

[3] Jens Forster, Christoph Schmidt, Oscar Koller, Martin Bellgardt, and Hermann Ney. Extensions of the sign language recognition and translation corpus rwth-phoenix-weather. In *LREC*, pages 1911–1916, 2014.

[4] Samuel Albanie, Gül Varol, Liliane Momeni, Triantafyllos Afouras, Joon Son Chung, Neil Fox, and Andrew Zisserman. BSL-1K: Scaling up co-articulated sign language recognition using mouthing cues. In *ECCV*, 2020.

[5] Samuel Albanie, Gül Varol, Liliane Momeni, Triantafyllos Afouras, Joon Son Chung, Neil Fox, and Andrew Zisserman. Bsl-1k: Scaling up co-articulated sign language recognition using mouthing cues. In *European Conference on Computer Vision*, pages 35–53. Springer, 2020.

[6] Hannah Bull, Annelies Braffort, and Michèle Gouiffès. Mediapi-skel-a 2d-skeleton video database of french sign language with aligned french subtitles. In *Proceedings of the 12th Conference on Language Resources and Evaluation (LREC 2020)*, pages 6063–6068, 2020.

[7] 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.

[8] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[9] Yi Zhu, Xinyu Li, Chunhui Liu, Mohammadreza Zolfaghari, Yuanjun Xiong, Chongruo Wu, Zhi Zhang, Joseph Tighe, R Manmatha, and Mu Li. A comprehensive study of deep video action recognition. *arXiv preprint arXiv:2012.06567*, 2020.

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

[11] 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.

[12] Zhe Cao, Gines Hidalgo, Tomas Simon, Shih-En Wei, and Yaser Sheikh. Openpose: realtime multi-person 2d pose estimation using part affinity fields. *IEEE transactions on pattern analysis and machine intelligence*, 43(1):172–186, 2019.

[13] Bowen Cheng, Bin Xiao, Jingdong Wang, Honghui Shi, Thomas S Huang, and Lei Zhang. Higherhrnet: Scale-aware representation learning for bottom-up human pose estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5386–5395, 2020.

[14] Will Kay, Joao Carreira, Karen Simonyan, Brian Zhang, Chloe Hillier, Sudheendra Vijayanarasimhan, Fabio Viola, Tim Green, Trevor Back, Paul Natsev, et al. The kinetics human action video dataset. *arXiv preprint arXiv:1705.06950*, 2017.

[15] Christoph Feichtenhofer, Haoqi Fan, Jitendra Malik, and Kaiming He. Slowfast networks for video recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6202–6211, 2019.

[16] Ceyuan Yang, Yinghao Xu, Jianping Shi, Bo Dai, and Bolei Zhou. Temporal pyramid network for action recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 591–600, 2020.

[17] Katrin Renz, Nicolaj C Stache, Samuel Albanie, and Gül Varol. Sign language segmentation with temporal convolutional networks. *arXiv preprint arXiv:2011.12986*, 2020.

[18] Yazan Abu Farha and Jurgen Gall. Ms-tcn: Multi-stage temporal convolutional network for action segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3575–3584, 2019.