UNIVERSITÉ DU QUÉBEC À MONTRÉAL

SIGN LANGUAGE RECOGNITION AND TRANSLATION

DISSERTATION

PRESENTED

AS PARTIAL REQUIREMENT

TO THE MASTERS IN COMPUTER SCIENCE

BY

FARES BEN SLIMANE

SEPTEMBER 2020

UNIVERSITÉ DU QUÉBEC À MONTRÉAL

RECONNAISSANCE ET TRADUCTION DE LA LANGUE DES SIGNES

MÉMOIRE

PRÉSENTÉ

COMME EXIGENCE PARTIELLE

DE LA MAÎTRISE EN INFORMATIQUE

PAR

FARES BEN SLIMANE

SEPTEMBRE 2020

ACKNOWLEDGMENTS

I would then like to thank my thesis advisor, Mr. Mohamed Bouguessa, for his support and advice. He consistently steered and guided me in the right direction throughout my years of study and through the process of researching and writing this thesis.

I would of course like to express my most sincere gratitude to my parents for their unconditional love, support and continuous encouragement throughout the years. They always believed in me and took the time to encourage and motivate me even in the most difficult situations. Mom and Dad; Thank You.

I would also like to thank my lab colleagues and friends (Nairouz, Yassine, Amine, Fawzi, Ghaieth and Antoine, etc.), who provided me with moral and intellectual support throughout this master's degree. Many thanks to Khalid Askia for his advice and support.

CONTENTS

LIST OF TABLES v
LIST OF FIGURES vi
CHAPTER I INTRODUCTION 1
1.1 Context 1
1.2 Motivations 4
1.3 Contributions 7
1.4 Thesis plan 8
CHAPTER II RELATED WORK 9
2.1 Sign Language Recognition 9
2.2 Dynamic Sign Language Recognition 11
2.2.1 Continuous Sign Language Recognition 12
2.2.2 Sign Language Translation 21
2.3 Attention for Recognition 24
2.4 Conclusion 25
CHAPTER III PROPOSED APPROACH: SIGN TRANSFORMER NETWORK 27
3.1 Background Information on the Transformer Network 28
3.1.1 Word Embedding 30
3.1.2 Attention Mechanism 31
3.1.3 Multi-Head Attention 32
3.1.4 Positional Encoding 33
3.1.5 Encoder Module 33
3.1.6 Decoder Module 34
3.2 The Proposed Sign Transformer Network 34

3.3 Context-Hand Attention Layer 37
3.4 Relative Local Context Masking 39
3.5 Encoder-Decoder Self-Attention 40
3.6 Hybrid Training 41
3.7 Implementation Details 42
3.8 Training Details 43
CHAPTER IV EXPERIMENTAL RESULTS 45
4.1 CSLR Experiments 45
4.1.1 Quantitative Results 46
4.1.2 Qualitative Analysis 49
4.2 Hyperparameters Validation 50
4.3 SLT Experiments 52
4.3.1 Quantitative Results 53
4.3.2 Qualitative Analysis 56
CHAPTER V CONCLUSION 58

LIST OF TABLES

Table		Page
3.1	Complexity comparison between recurrent and attention operations.	29
4.1	Comparison of our Sign Transformer Network variants on RWTH-PHOENIX-Weather 2014 in Word Error Rate % (the lower the better).	47
4.2	Comparison of our Sign Transformer Network with the Full Frame Word SubUnets from Camgoz et al. (2017) in Word Error Rate % (the lower the better). Both models are only trained on full frame images and use CTC for alignment.	47
4.3	Comparison of our best performing model with the published prior studies on RWTH-PHOENIX-Weather 2014 in Word Error Rate % (the lower the better).	49
4.4	Comparison of our encoder-decoder Sign Transformer networks with the baseline results (Camgoz et al., 2018) on RWTH-PHOENIX-Weather 2014T using BLEU and ROUGE scores % (the higher the better).	55
4.5	Comparison of the output translations from our encoder-decoder STN models to those shared by Camgoz et al. (2018).	57

LIST OF FIGURES

Figure		Page
1.1	American Sign Language manual alphabet.	2
1.2	The sign *hello* in Quebec Sign Language (QSL).	3
1.3	An example taken from Camgoz et al. (2018), showing the difference between the two tasks of Continuous Sign Language Recognition (CSLR) and Sign Language Translation (SLT).	6
2.1	Classical approach for the task of CSLR.	13
2.2	Iterative re-alignment algorithm from (Koller et al., 2017).	17
2.3	The SubUnets architecture proposed by Camgoz et al. (2017). The network is composed of three modules: Hand SubUnet (top stream), a Word SubUnet (bottom stream) and a second Word SubUnet (middle stream). IP refers to the Inner Product layer.	19
2.4	The two-Stream 3D CNN from Huang et al. (2018). The network takes as input a sequence of images of size 16. The output represents the global-local video representation.	20
2.5	The Hierarchical Attention Network (HAN) architecture proposed by Huang et al. (2018) for the task of CSLR. The input of the HAN is the global-local representation produced by the two-Stream 3D-CNN as depicted in Figure 2.4. The w refer to the word representations produced by the word encoder. The h refer to the latent representation and the y refer to the word probabilities.	21
2.6	An overview of the Sign2Gloss2Text approach proposed by Camgoz et al. (2018).	24
3.1	A hypothetical example demonstrating the self-attention mechanism for the word token *JFK*.	29
3.2	An overview of the Transformer Network from (Vaswani et al., 2017). The Input and Output embeddings use a word2vec network.	30

3.3 Multi-Head Attention from (Vaswani et al., 2017). 32

3.4 Overview of our Sign Transformer Network architecture. 36

3.5 Combination of both the full-frame and the handshape streams through a Context-Hand Attention layer. 38

3.6 Relative local context mechanism in the Context/Hand Attention layer. 39

3.7 Overview of our Hybrid STN approach. 42

4.1 The Word Error Rate learning curve of our STN variants for the task of CSLR on the RWTH-PHOENIX-Weather 2014 dataset. . 48

4.2 The top sequence is the output results of our STN network. The middle is for STN with the hand stream and the bottom is for STN with hand stream and the local context masking. Note that this example is randomly chosen and not cherry-picked. 51

4.3 Comparison of our STN model with different learning rate settings. 52

4.4 Comparison of our STN model with different number of attention layers. 53

4.5 Comparison of our STN model with different number of attention heads. 54

4.6 The BLEU-4 score learning curve of our encoder-decoder STN models for the task of SLT on the RWTH-PHOENIX-Weather 2014T dataset. 55

RÉSUMÉ

Outre les gestes de la main, la langue des signes utilise simultanément différents composants pour transmettre un message. À titre d'exemple, l'orientation des doigts, les mouvements des bras ou du corps ainsi que les expressions faciales. Parfois, un composant spécifique peut jouer un rôle majeur dans la modification de la signification du signe ou peut ne pas être requis pour interpréter un signe. Pour cela, il est primordial pour un système de reconnaissance de n'utiliser que les informations pertinentes pour traduire un signe. Dans ce contexte, nous avons élaboré le *Sign Transformer Network*, un réseau attentionnel pour traiter les deux tâches de: Reconnaissance Continue de la Langue des Signes et la Traduction en Langue des Signes. Il prend en entrée une séquence d'images qui désigne le signe à traduire et produit une traduction textuelle cohérente dans une langue parlée. Notre système est basé sur la nouvelle architecture neuronale *Transformer Network* qui a la capacitée de découvrir et d'apprendre, efficacement, les informations spatio-temporelles des données continues. Nous montrons qu'en utilisant simplement l'auto-attention pour la modélisation temporelle, nous surpassons presque toutes les études précédentes, prouvant la supériorité de l'auto-attention sur les réseaux traditionnels basés sur la récurrence.

Même si la langue des signes est multicanal (plusieurs canaux d'informations), les formes de mains représentent les entités centrales dans l'interprétation des signes. Afin d'interpréter correctement la signification d'un signe, les gestes de la main doivent être identifiés dans leur contexte approprié. En tenant compte de cela, nous utilisons le mécanisme d'auto-attention pour agréger efficacement les caractéristiques de la main avec leur contexte spatio-temporel approprié pour une

meilleure reconnaissance des signes. Ainsi, notre modèle est capable d'identifier les composants essentiels[1] de la langue des signes qui tournent autour de la main dominante et le visage. Nous testons notre modèle en utilisant la base de données RWTH-PHOENIX-Weather 2014 et sa variante RWTH-PHOENIX-Weather 2014T. Nous avons obtenu des résultats compétitifs sur les deux ensembles de données et surpassons de manière significative la plupart des approches existantes.

Mots-clés Intelligence artificielle; Apprentissage profond; Vision par ordinateur; Reconnaissance de l'action; estimation de la pose; Reconnaissance de la langue des signes; Traduction en langue des signes; Traitement du langage naturel; Auto-attention; Réseau de transformateurs.

[1]Un composant est jugé essentiel lorsqu'il est utilisé pour interpréter un signe donné.

ABSTRACT

Besides hand gestures, sign language simultaneously uses different components to convey a certain message. For example, the orientation of the fingers, the movements of the arms or body as well as the facial expressions. Sometimes a specific component can play a major role in changing the meaning of the sign or it may not be required to interpret a sign at other times. Accordingly, it is essential for a recognition system to use only the relevant information for sign prediction. In this context, we propose the Sign Transformer Network, an attentional network for both tasks: Continuous Sign Language Recognition and Sign Language Translation. It takes as input a sequence of images which designates the sign to translate and produces a coherent textual translation in a spoken language. Our system is based on the new neural architecture, Transformer Network, which has the capacity to efficiently discover and learn the spatio-temporal information of continuous data. We show that by merely using self-attention for temporal modelling, we outperform nearly all previous studies, proving the superiority of self-attention over traditional recurrent based networks.

Even though, Sign Language is multi-channel (multiple channels of information), handshapes represent the central entities in sign interpretation. Seeing handshapes in their correct context defines the meaning of a sign. Taking that into account, we utilize the self-attention mechanism to efficiently aggregate the hand features with their appropriate spatio-temporal context for better sign recognition. We found that by doing so, our model is able to identify the essential Sign

Language components[2] that revolve around the dominant hand and the face areas. We test our proposed approach on both RWTH-PHOENIX-Weather 2014 and its variant RWTH-PHOENIX-Weather 2014T, yielding competitive results on both datasets and significantly surpassing most state-of-the-art previous studies.

Keywords Artificial intelligence; Deep learning; Computer vision; Action Recognition; pose estimation; Sign Language Recognition; Sign Language Translation; Natural language processing; Self-Attention; Transformer Network.

[2]A component is considered essential when it is used to interpret a given sign.

CHAPTER I

INTRODUCTION

1.1 Context

Sign languages are the primary means of communication of the deaf and the hearing impaired. It is typically expressed in visual gestures and continuous signs. The language uses gestures to mimic or illustrate an object, a feeling, an expression or even an action. Like spoken languages, sign language is not universal. There is no unique language that is shared by deaf people around the world. Several distinct sign languages have evolved independently according to countries and even regions. There are more than 130 different sign languages around the world (glottolog, 2020). In the United States alone, it is estimated that more than 500,000 people use the American Sign Language (ASL) (Mitchell et al., 2006). According to Statistics Canada, nearly 25,000 people reported speaking a sign language in Canada (Canada, 2018). These languages can differ enormously in which each uses its own lexicon and manual alphabet. For instance, a person who uses the American Sign Language (ASL) will have difficulty communicating with another person who signs in the British Sign Language (BSL).

Fingerspelling is the process of spelling out (one letter at a time) words that have no existing sign such as proper names like a person's name, cities, products, etc. This method is carried out using the handshapes associated with the letters

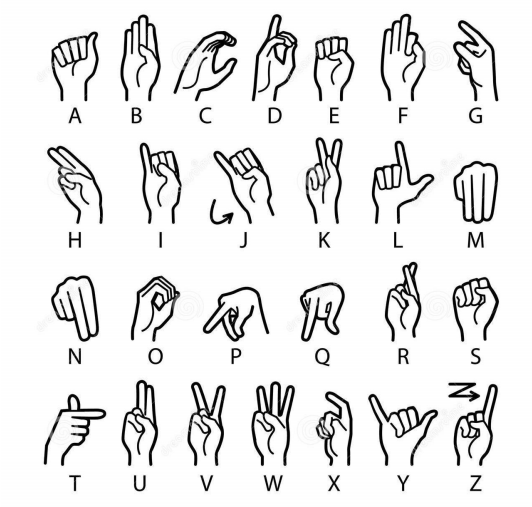

Figure 1.1 American Sign Language manual alphabet.

of the manual alphabet. Each sign language has its own manual alphabet. For example, as can be seen in Figure 1.1 (Rommeo79, 2019), the American Sign Language uses the ASL manual alphabet, which consists of 22 handshapes that– when held in certain positions and/or movements represent the 26 letters of the American alphabet(lifeprint, 2019a). Nonetheless, in order to express a distinct word, Sign Language mostly employ continuous and well-defined combinations of hand and body movements, instead of spelling the word letter by letter. For example, Figure 1.2 demonstrates the signing of the word *"Hello"* in Quebec Sign Language (Foundation des sourds du Quebec, 2019).

Sign languages are often defined as manual languages[1]. However, besides the hand articulations, non-manual components like facial expressions, arm, head, body movements and positions play a crucial part in Sign Languages (Crasborn, 2006). Any change in one of these components can alter the meaning of a sign. Usually, the handshape performed by the dominant hand carries most of the meaning of

[1]Manual languages are languages that use hands to interpret a word.

Figure 1.2 The sign *hello* in Quebec Sign Language (QSL).

the sign (Ding & Martinez, 2007). As for the other components, they provide a semantic context to it. For example, the signs for *"man"* and *"woman"* are very close as they both use the same handshape and body movement. Although, as mentioned in (Weronika Lass, 2019), male words tend to be signed close to the forehead while female signs are usually performed around the chin. Similarly, for the signs *"sit"* and *"chair"*, they both have the same handshape but slightly different hand movements, in which *"chair"* is the sign *"sit"* but twice (Valli & Lucas, 2000).

Facial expressions can also greatly affect the meaning of a handshape, and are mostly used to ask questions or express emotions. For instance, when signing a *"you"* sign, by raising the eyebrows, it becomes a question: *"Is it you?"* or by using a different facial expression it turns into *"It's you?!?!"* meaning that *"I am*

surprised that it is you" (lifeprint, 2019b).

These examples highlight the idea that interpreting a sign often requires recognizing the handshape accompanied by its contextual information. This context is not limited to the spatial information that resides around the handshape for a fixed point in time, but also the temporal information that consists of hand and body movements.

1.2 Motivations

Much of the current literature in Continuous Sign Language Recognition (CSLR) ignores the notion of incorporating contextual information to the handshape. Instead they either exclusively use the handshape features from the cropped hand images (Koller et al., 2016), or simply use global features by trying to learn a complete spatial representation of full body images (Koller et al., 2017; Wu et al., 2016).

Nevertheless, there have been some works that investigated the effect of combining information of the handshape with the other sign language modalities. For instance, Huang et al. (2018) use a two-stream 3D-CNN to extract global (full body) and local (hand) features and then fuse the representations in the last fully connected layers. Another work (Camgoz et al., 2017) involves the use of specialized sub-networks (SubUNets) to learn full frame and handshape information and then synchronize the modalities through a recurrent layer. The authors found that combining modalities in this manner outperforms the use of isolated SubUNets. However, it is important to note that sign language modalities mostly share complex non-linear relations. Because of that, these approaches may fail to successfully capture and aggregate the required handshape dependencies. This is especially true when using recurrent networks, as they are more biased by

close contexts, only considering short-term dependencies rather than the global long-term dependencies (Bengio et al., 1994). In this study, we propose a superior method that efficiently combines handshape features with their appropriate global context, and which yields better recognition results than the latter studies.

Virtually, all progress in CSLR has operated under the assumption that this is a gesture recognition task and that the main challenge is to simply learn the mapping of signs to their respective glosses [2]. Camgoz et al. (2018) argue that this is not the case, as Sign Language has it is own linguistic and grammatical properties. Therfore, we should take into account the word orders and grammar. That being the case, they introduce a new problem setup: Sign Language Translation (SLT), in which they approach it as a machine translation task, generating spoken language translations from sign language videos. And they, accordingly, employ an encoder-decoder RNN (Recurrent Neural Network) with an attention mechanism which is commonly used for machine translation tasks. The work of Camgoz et al. (2018) suggests to consider sign language recognition as a machine translation task instead of a simple gesture recognition. The main difference between both tasks (CSLR and SLT) is that CSLR regards learning only the monotonic alignment [3] between the source and target sequences. On the other hand, SLT also envisages the non-monotonic alignment between the sequences, in which a future word in a target sentence can align with an earlier word in the source sentence. This notion is clearly illustrated in Figure 1.3.

Alternatively, it has been shown that using an Attention Network [4] by itself can

---

[2]Gloss is an annotation that represent or explain a word in a text.

[3]Monotonic alignment refers to having two sequences of data that preserves the same order of alignment of their elements.

[4]The notion of attention is explained in Section 3.1 of Chapter 3

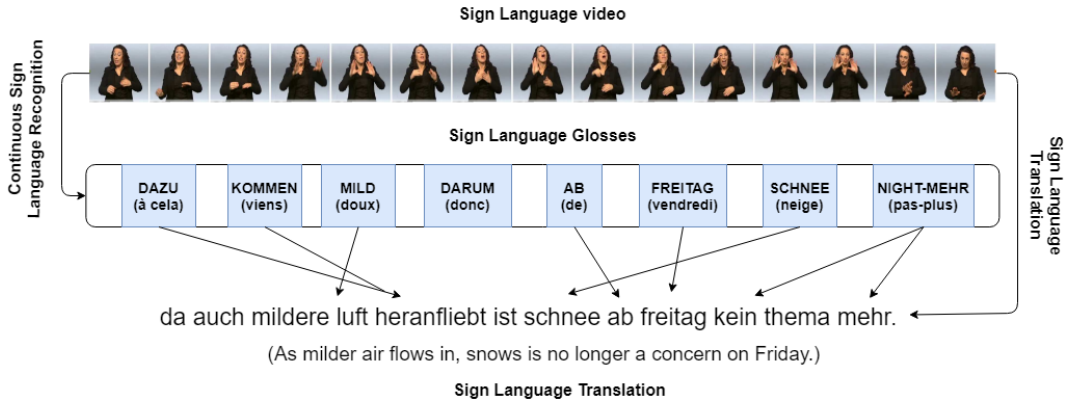

Figure 1.3 An example taken from Camgoz et al. (2018), showing the difference between the two tasks of Continuous Sign Language Recognition (CSLR) and Sign Language Translation (SLT).

also be efficient in machine translation. The Transformer Network from (Vaswani et al., 2017), which is based solely on self-attention, has been proven to be very effective in capturing dependencies. Just recently, Camgoz et al. (2020) proposed a transformer-based encoder-decoder architecture for the task of SLT which significantly exceeds the previous state-of-the-art performance from Camgoz et al. (2018) and sets a new baseline result for SLT. This proves the efficiency of such architecture for capturing temporal information. When signing, the handshape usually depends on some relevant contexts, rather than all context information. As a result, the transformer network is a more suitable choice for our problem setup, as it can efficiently combine hand features with their appropriate full-body information. This is because it is explicitly built to detect important dependencies, as opposed to their recurrent counterpart.

1.3 Contributions

In this thesis, we propose a novel approach based on the Transformer Network for seq2seq (sequence to sequence) sign language applications. Our proposed model has the ability to discover and learn spatio-temporal information from continuous data. The system will be adequate to both tasks: Continuous Sign Language Recognition (CSLR) and Sign Language Translation (SLT). It takes as input a sign video clip (sequence of images) and accordingly produce the corresponding translation in a spoken language. Unlike previous works, the originality of our approach lies in explicitly picking up and aggregating contextual information from the non-manual sign language components. Without any domain annotation, our approach is able to exclusively identify the most relevant features associated with the handshape when predicting a sign. The main contributions of this study can be summarized as follows:

- Devising an end-to-end framework for sequence to sequence Sign Language Translation and Recognition that utilizes self-attention for temporal modeling.
- Elaborating a more efficient method to incorporate handshapes with their spatiotemporal context for Sign Language Recognition.
- Outperforming previous studies that do not use a forced alignment[5] approach on both CSLR and SLT in terms of Word Error Rate as well as with BLEU and ROUGE Scores respectively.

[5]The notion of forced alignment will be explained in the Section 2.2.1 of Chapter 2.

1.4 Thesis plan

In this study, we focus primarily on the tasks of continuous sign language recognition and translation. The rest of this thesis is organized as follows:

- Chapter 2, presents a literature review on domain-related approaches. We start by broadly introducing early works applied to the task of Sign Language Recognition (SLR). Then, we specifically report recent and purely visual approaches assigned to both Continuous Sign Language Recognition (CSLR) and Sign Language Translation (SLT).
- Chapter 3 describes in details our proposed approaches for both CSLR and SLT.
- Chapter 4 is devoted to the presentation of our experimental results, and analyzes the performance of our proposed approach compared to state-of-the-art previous work.
- Finally, Chapter 5 presents an overall conclusion and discussion of the proposed approach, as well as future lines of research.

CHAPTER II

RELATED WORK

Sign Language Recognition (SLR) is multidisciplinary that involves a variety of research areas like computer vision, natural language processing, and sequence modelling. In general, SLR studies can be broadly divided into three main categories: sensor-based approaches, vision-based approaches, and hybrid approaches. An overview of related studies within these three main categories is provided in Section 2.1 of this chapter. While a comprehensive survey is beyond the scope of this chapter, we provide a critical review to put our work in perspective relative to existing main stream SLR approaches.

In Section 2.2, we present a detailed literature on the specific problem of dynamic sign recognition. And we specifically focus on recent vision-based studies that are applied for both Continuous Sign Language Recognition and Sign Language Translation. Finally, in Section 2.3, we provide an overview of studies that incorporate attention for neural networks, primarily focusing on close-related tasks, such as action recognition and language translation.

2.1 Sign Language Recognition

In recent years, there has been an increasing amount of literature on Sign Language Recognition systems. A great deal of the work done in this area uses

glove-based methods (Oz & Leu, 2011; Liang & Ouhyoung, 1998; Mehdi & Khan, 2002; Kuroda et al., 2004) or motion tracking gadgets and sensors (Brashear et al., 2005) to detect hand's gestures. Wu et al. (2015) use the Electromyography (EMG)[1] to reach high performance in recognizing the signs of American Sign Language (ASL) in real time. Zhang et al. (2009, 2011) perform hand gesture recognition based on the information fusion of a three-axis accelerometer (ACC) and (EMG) sensors. Georgi et al. (2015) merge the signals of both the inertial measurement unit (IMU)[2] and the Electromyography (EMG) to achieve a great recognition rate of 97.8% on a set of 12 gestures.

Due to the high cost of the motion tracking devices and the recent advancement of visual methods, many researchers started to shift to purely visual methods for capturing hand gestures. In this approach, the visual systems only exploit the extracted images as input data for gesture recognition. This eliminates the need for sensors and/or gloves and significantly reduces the costs of recognition systems (Ahmed et al., 2018). For example, Karami et al. (2011) designed a system for recognizing static gestures of Persian Sign Language (PSL) using the wavelet transform and neural networks. Their system is capable of recognizing 32 PSL alphabets with an average classification accuracy of 94.06% from digital images. Many other studies, such as Zafrulla et al. (2011); Lang et al. (2012b,a), use the *Microsoft Kinect* environment that is able to provide depth and colour data. This leads to more accurate and easier tracking of the hand and body movement. The obvious advantage of this kind of approach is that the person is not required to wear uncomfortable devices such as gloves or motion sensors when

---

[1]Electromyography (EMG) is a method used to detect the electrical activity of skeletal muscles.

[2]An Inertial Measurement Unit (IMU) is an electronic device which uses accelerometers to estimate body's force and angular rate.

signing. However, acquiring hand information is more difficult through mere visual techniques. The hands have a complex structure that is hard to capture and can be easily affected by image variations (e.g., different backgrounds, lightning). That is why image processing methods are generally needed for effective recognition when using visual methods.

There are works that combine vision-based approaches and sensors to acquire better recognition results. We refer to these approaches as Hybrid Systems. Gałka et al. (2016) argue that vision-based approaches are very sensitive to image variations. To alleviate this, the authors propose a more robust recognition system, which uses sensors and accelerometer gloves. They found that the measurements of these sensors greatly contribute to improving their prediction performance. The system achieves a significant accuracy of 99.75% on a set of 40 gestures. Another work presented by McGuire et al. (2004), uses gloves and Hidden Markov Models (HMM) to obtain recognition results of 94% accuracy on a vocabulary of 141 signs. Accordingly, hybrid Systems leverage the advantages of using both vision techniques and sensor-based methods, which results in more robust and accurate sign recognition.

2.2 Dynamic Sign Language Recognition

Sign language gestures are generally dynamic, including both spatial and temporal information. Some studies in the SLR literature simply focus on the spatial component of the gestures, recognizing only static signs. On the other hand, dynamic recognition requires modelling both the temporal and spatial dependencies of the signs. The research on dynamic sign language recognition can be mainly divided into two families: isolated and continuous recognition. Most works in this line of research employ HMMs to model the temporal information of each predicted

sign (Aran, 2008). Isolated recognition attempt to recognize a unique sign that is performed alone. Alternatively, continuous sign recognition deals with recognizing a sequence of signs. This introduces its own challenges, such as learning the implicit segmentation of the succeeding signs. Additionally, movements that occur during transitions between signs, can interfere and complicate sign recognition during continuous signing (Aran, 2008). In what follows, we will specifically focus on recent vision-based previous works applied for both tasks: Continuous Sign Language Recognition (CSLR) and (Sign Language Translation) SLT.

2.2.1 Continuous Sign Language Recognition

Until recently, most sign language recognition studies operated mainly on isolated signs. However, as labelled continuous datasets became more publicly available, research interest has started to shift towards CSLR (Camgoz et al., 2017). For instance, the dataset RWTH-PHOENIX-Weather 2014 (Koller et al., 2015) that quickly became a popular baseline for CSLR. The dataset offers sign language gloss annotation for a German weather forecast. In our work, we utilize this dataset to validate our own CSLR experiments and to compare our findings with the state of the art.

Most work in CSLR can be mainly divided into three stages: (1) the spatial modelling that aim to extract the spatial representations from the input images, (2) the temporal modelling that learns the temporal dependencies from the different sequence time steps, and finally, (3) the sequence alignment that involves learning the correspondence between the representations' sequence and the target sequence. This is clearly illustrated in Figure 2.1. There exists a fair amount of literature in the field that utilizes handcrafted feature representations (Monnier et al., 2015) and classical models like HMMs for sequence learning (Koller

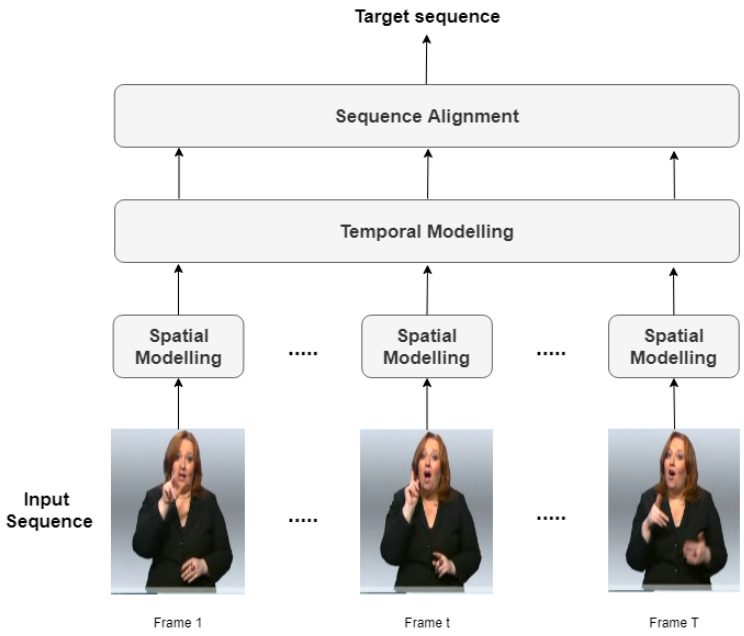

Figure 2.1 Classical approach for the task of CSLR.

et al., 2016). However, due to the revolutionary advances in deep learning, recent research has started to shift interest to using CNNs (Koller et al., 2016), and 3D-CNNs (Huang et al., 2018) as feature extractors, and recurrent networks for temporal modelling (Camgoz et al., 2017). Besides HMMs, a common strategy used for sequence to sequence modelling is the Connectionist Temporal Classification (CTC)[3] (Graves et al., 2006). CTC has been successfully applied for CSLR (Cui et al., 2017; Camgoz et al., 2017) and also for various other sequence to sequence problems including speech recognition (Graves et al., 2013) and handwriting recognition (Graves et al., 2008).

An early work from Koller et al. (2016), proposes a CNN-HMM approach for the

[3]Connectionist Temporal Classification (CTC) is a type of neural network usually applied as a loss function for training RNN-based networks. It is generally used to solve sequential problems with variant sequence lengths.

task of CSLR, leveraging CNNs for feature extraction and HMM for both sequence learning and temporal modelling. Given a sequence of features $x_1^T = \{x_1..x_T\}$, the model aim to infer the sequence of sign words $w_1^N = \{w_1..w_N\}$, in which T and N account for the length of source and target sequences respectively. The image features x_1^T are extracted using a CNN model. Respectively, the authors (Koller et al., 2016) opted for the GoogLeNet CNN architecture (Szegedy et al., 2015) because of its high efficiency as a feature extractors and its low computational capacity.

Koller et al. (2016) argue that sign images and sign words occur in a monotonous fashion. Therefore, the task of continuous sign recognition doesn't require learning the reordering of words, which explains their use of the HMM for sequence to sequence modelling. Their model is trained, in an end-to-end fashion, to maximize the target posterior probability $p(w_1^N|x_1^T)$. It finds the optimal word prediction that best matches the input image features, as can be seen in the following equation:

$$[w_1^N]^* = \arg\max_{w_1^N} p(w_1^N|x_1^T) \tag{2.1}$$

By applying the Bayes rule theorem on the posterior probability, we end up having:

$$[w_1^N]^* = \arg\max_{w_1^N} \frac{p(x_1^T|w_1^N).p(w_1^N)}{p(x_1^T)} \tag{2.2}$$

Removing the $p(x_1^T)$, since it doesn't account for any change with respect to w_1^N.

$$[w_1^N]^* = \arg\max_{w_1^N} p(x_1^T|w_1^N).p(w_1^N) \tag{2.3}$$

The posterior probability is estimated by the product of the prior word probability

$p(w_1^N)$ and the conditional probability $p(x_1^T|w_1^N)$. $p(w_1^N)$ is estimated through a language model, whereas $p(x_1^T|w_1^N)$ is modelled by a vision model, which is expressed by a HMM. The vision model consists of modelling a sequence of feature vectors given a sequence of words. On the other hand, the language model repose on modelling the word probabilities that can be estimated by the word occurrences in the training corpus. The overall network combines both the vision and language models to infer the maximum likelihood of the target word sequences w_1^N. The authors models the temporal aspect of the input through a HMM. Each word is expressed by predefined hidden states s. Following a first order markovian assumption, each hidden state s_t depends solely on its previous state s_{t-1}. Thus, the conditional probability can be expressed by the following equation:

$$p(x_1^T|w_1^N) = \sum_{s_1^T} \prod_t p(x_t|s_t, w_1^N).p(s_t|s_{t-1}, w_1^N) \tag{2.4}$$

Placing this expression in Equation 2.3, we obtain:

$$[w_1^N]^* = \arg\max_{w_1^N} \left\{ p(w_1^N) \max_{s_1^T} \left\{ \prod_t p(x_1^T|s_t, w_1^N).p(s_t|s_{t-1}, w_1^N) \right\} \right\} \tag{2.5}$$

Koller et al. (2016) train their proposed model on the cropped hand patches, only employing the dominant hand (right hand) as input. Respectively, they adopt a dynamic programming based approach to efficiently track the appropriate hand across the sequence of images. To account for data augmentation, the authors employ random cropping of hand patches and resize the input images to 224 $\times$ 224 sizes.

In another work, Koller et al. (2017) propose a CNN-RNN-HMM architecture and achieve state-of-the-art results in the task of CSLR, broadly outperforming the previous studies by a large margin. Contrary to their previous work (Koller et al.,

2016), the authors train their model on the full-frame RGB images which contain all the sign language visual information. They found that using full frames as input data considerably outperform merely using the dominant hand. They claim that because of the multi-channel nature of sign language, using only the dominant hand features will not result in an optimal recognition. As mentioned before, Koller et al. (2016) primarily rely on HMM to capture temporal information. Koller et al. (2017) claim that since sign language involves motion, capturing temporal variations through a HMM may not be sufficient for sign recognition. As a result, they employ a CNN-BLSTM (a CNN followed by a bidirectional LSTM layer) to model both spatial and temporal information of sign inputs. And they use the HMM only for the sequence to sequence alignment. Similarly to Koller et al. (2016), they exploit the deep CNN architecture GoogleNet, which is firstly pretrained on Imagenet (Deng et al., 2009).

On top of their CNN-RNN-HMM model, Koller et al. (2017) employ an iterative forced alignment algorithm that refines the label-to-image alignment and that broadly outperforms the previous work by a large margin. Forced alignment is usually applied in tasks like speech recognition. Instead of manual labelling, forced alignment is able to provide useful training targets, which are re-fined through iterative alignment. At each iteration, a speech recognition model produces a new set of training targets. The model will be trained on these new labels and re-align them for the next iteration. This training process is repeated until the resulting model stops getting better (dpwe, 2000).

Koller et al. (2017) follow a similar strategy for CSLR. They start by using a deep CNN-BLSTM network trained for per frame labelling, which is then embedded into an HMM for target sequence recognition. The resulting model iteratively re-align and corrects frame labels until they obtain the best recognition. The model continuously improves its performance by refining label-to-image prediction

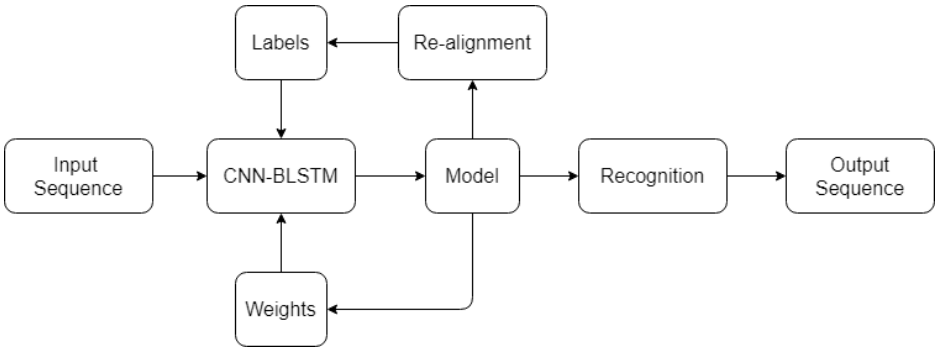

Figure 2.2 Iterative re-alignment algorithm from (Koller et al., 2017).

at each iteration. As shown in Figure 2.2, the CNN-BLSTM uses the previous iteration's model weights as initialization and profit from the new frame labels. Despite the promising results in sign recognition, their high dependency on the iterative forced alignment step is not disregardable. The key limitation in such method is that it enforces the independence between adjacent observations (frame and frame-label) (Kornai, 1996).

The previously mentioned publications use unique channel of information, whether it is the dominant hand or the full body representation. A recent study by Koller et al. (2019) employ the same forced alignment technique as (Koller et al., 2017) to maximize the target likelihood estimation. The authors adopt a multi-stream HMM approach to synchronize and learn from parallel and different sub-problems (sign language, mouth shape, and hand shape recognition). Respectively, using a parallel alignment approach produces better recognition performance than [7] which follows a similar CNN-LSTM-HMM setup but uses a single stream input.

Similarly, Camgoz et al. (2017) argue that unlike spoken languages, sign language is undoubtedly multi-channel. Information is carried through handshapes, body movements and facial expressions. Instead of relying on full frame representations, they propose an architecture that combines different modality information.

Accordingly, they use specialized sub-networks (SubUnets), where each is trained to model a specific expert knowledge domain. A hand SubUnet that is trained on cropped hand patches to recognize and align handshape sequences. A Word SubUnet that is trained on the full-frame images and that models the complete body representation. They combine and synchronize the different channels of information through an additional BLSTM layer. Camgoz et al. (2017) found that combining modalities in this manner outperform the use of isolated SubUnets. As depicted in Figure 2.3, the overall model is trained jointly using the Cross Entropy losses produced by the three loss layers (from the hand SubUnet, Word SubUnet and the second Word SubUnet). As discussed in the introduction section, combining channels' information in this manner usually results in a biased outcome. This is because recurrent networks usually focus on close context information and ignore long-ranged dependencies. Consequently, the aforementioned approach may fail to capture the required full-frame context information.

As can be seen in Figure 2.3, the overall architecture is composed of three main components. The Word SubUnet module that accepts full-frame images as inputs, thus learning the full-body information. The Hand SubUnet module that is equipped to merely learn the hand representations. And a second Word SubUnet that combines both full-body and hand information. Each module involves the use of CNNs to extract spatial features from input images and Bidirectional LSTMs for temporal modelling. Camgoz et al. (2017) employ a deeper bidirectional LSTM in the full-frame module to better model the full-body representations. For sequence to sequence alignment, they apply the CTC layer to recognize and time-align the target sequences. They also experiment with using the HMM for sequence modelling. They found that HMM considerably outperform the CTC, especially when used with a language model. For training, the authors exploit the CNN architecture CafeNet (Jia et al., 2014) because of its low computational ca-

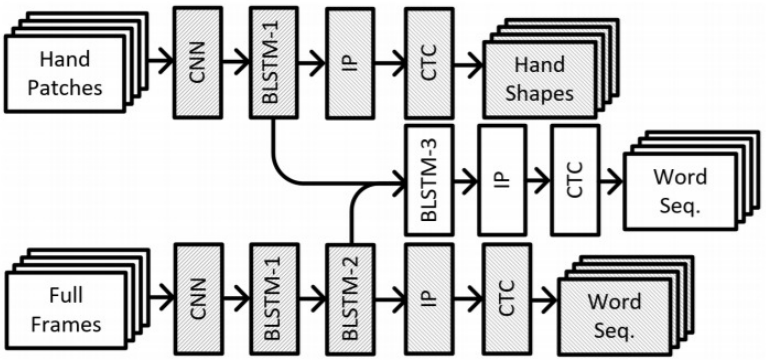

Figure 2.3 The SubUnets architecture proposed by Camgoz et al. (2017). The network is composed of three modules: Hand SubUnet (top stream), a Word SubUnet (bottom stream) and a second Word SubUnet (middle stream). IP refers to the Inner Product layer.

pacity. Respectively, the input images are resized to 227×227, whereas the input sequences are padded to 300. The model is trained end-to-end using the Adam optimization algorithm (Kingma & Ba, 2014).

Another work proposed by Huang et al. (2018) attempts to also combine full-body information with the hand representations for CSLR. The authors employ a two-stream 3D-CNN that extracts spatiotemporal features from a video of 16 frames. As can be seen in Figure 2.4, the upper stream extracts global information from the full-frame input sequence. On the other hand, the lower stream models the local (hand) information from the cropped hand patches. The last two fully connected layers fuse both local and global feature representations from both streams. The two-stream 3D-CNN is based on the C3D (Tran et al., 2015) architecture which contains eight convolutional layers and five pooling layers.

The authors (Huang et al., 2018) also propose a Hierarchical Attention Network (HAN) on top of the two-Stream 3D-CNN that incorporates an encoder-decoder LSTM structure with the attention mechanism. The training objective is to learn

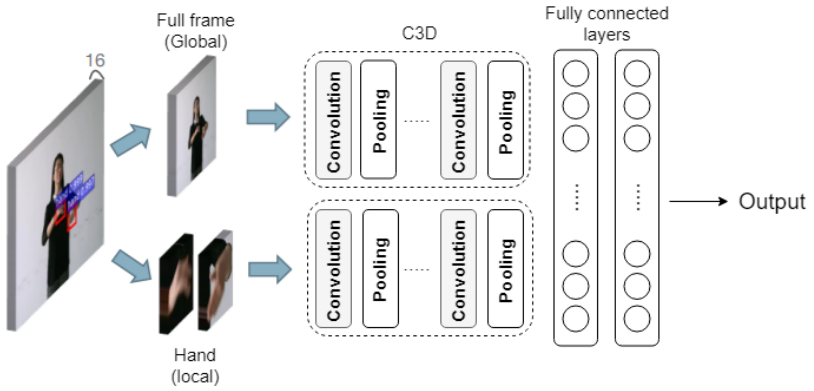

Figure 2.4 The two-Stream 3D CNN from Huang et al. (2018). The network takes as input a sequence of images of size 16. The output represents the global-local video representation.

the conditional probability $p(y|v)$. $y = \{y_1..y_N\}$ represent the target gloss words, while, $v = \{v_1..v_T\}$ express the global-local video representations which has been extracted from the two-stream 3D-CNN. The Hierarchical Attention Network is usually applied in Text Classification (Yang et al., 2016) to consider the hierarchical structure of documents (documents contain sentences which consists of words). It uses two levels of attention models, thus the name hierarchical attention network. At first, the network employ stacked recurrent networks on words followed by an attention layer to extract the sentence representation vectors. Then, the same procedure applied to the sentence vectors to generate a vector who conceives the meaning of the whole document. The final vector representation is then passed for text classification.

Respectively, Huang et al. (2018) apply a similar strategy for the task of CSLR using their HAN approach. As can be seen in Figure 2.5, the network encoder side is composed of two levels. The first (clip encoder) models words from the input video representations while the second (word encoder) models sentences from the

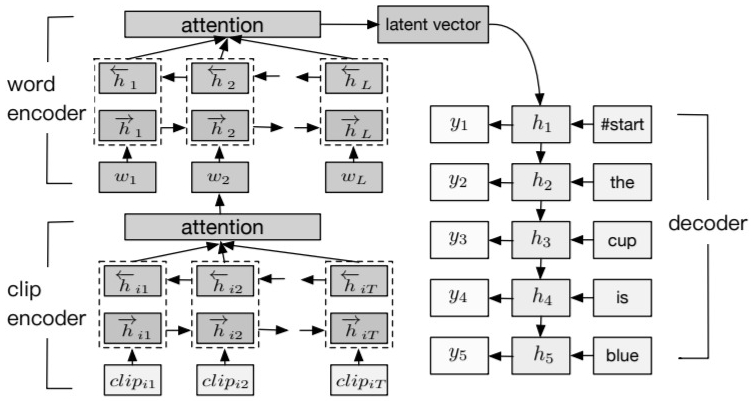

Figure 2.5 The Hierarchical Attention Network (HAN) architecture proposed by Huang et al. (2018) for the task of CSLR. The input of the HAN is the global-local representation produced by the two-Stream 3D-CNN as depicted in Figure 2.4. The w refer to the word representations produced by the word encoder. The h refer to the latent representation and the y refer to the word probabilities.

derived words vectors. The authors argue that this reflects the hierarchical nature of the task, in which video clips form words and words form sentences. The resulting latent vector representation is then passed to a decoder to produce the target sequence. The two level encoders of the network consists of a Bidirectional LSTM (BLSTM) with attention, whereas the decoder uses a single normal LSTM layer. A softmax activation function is applied on the decoder output vector h to generate the word probabilities y which will be used to train the model.

2.2.2 Sign Language Translation

Recently, Camgoz et al. (2018) propose to treat sign language recognition as a machine translation task. Consequently, they introduce the updated version of the Pheonix-2014 dataset : RWTH-PHOENIX-Weather 2014T. The dataset contains both gloss annotations and spoken language translations, making it more suitable

for Sign Language Translation. In our work, we demonstrate the effectiveness of our approaches through the use of this dataset on the task of SLT. And we compare our findings with the SLT baseline results from (Camgoz et al., 2018).

As mentioned previously, SLT tend to capture the non-monotonic relations between the source and target sequences, as opposed to CSLR. Camgoz et al. (2018) argue that the alignment between sign input sequence and the target translation is usually non-monotonic. Because of that, we can't use algorithms like CTC or HMM that only account for the monotonic alignment. Instead, Camgoz et al. (2018) adopt an encoder-decoder RNN structure that maps the input image sequence to the translation target sequence. The proposed approach models the conditional probability $p(y|x)$ producing target sequence $y = \{y_1..y_N\}$ from input sign frames $x = \{x_1..x_T\}$. N and T account for the lengths of both target and source sequences respectively. The authors apply CNNs to learn the spatial embeddings of sign images and recurrent networks for temporal modelling. They also utilize the attention mechanism in order to avoid the problem of the vanishing gradients [4]. They found that having an attention mechanism exceptionally improved the translation performance. Camgoz et al. (2018) refer to their proposed approach as Sign2Text.

Given a sign clip (sequence of input frames), Sign2Text first employ CNNs to extract the sequence representations $z = \{z_1..z_T\}$, which will be then passed to the encoder to model the temporal dependencies of sign frames. The encoder which is composed of a stack of recurrent units, produces a latent representation h_{sign} that conceives the entire input sequence information. The latent representation h_{sign} is then passed to the decoder, as its initial hidden state h_0. Similarly, the

[4]Vanishing gradient problem is usually encountered when training neural networks using gradient-based methods.

decoder is composed of a stack of recurrent units. Each recurrent unit accepts as input a hidden state h_{t-1} from the previous step and the word embeddings g_{t-1} of the previous predicted word y_{t-1}. And it accordingly produces the hidden state h_t along with the next word y_t, as expressed in the following equation:

$$y_t, h_t = \text{Decoder}(g_{t-1}, h_{t-1}) \tag{2.6}$$

The overall network (Sign2Text) is trained end-to-end using the conditional probability $p(y|x)$:

$$p(y|x) = \prod_{t=1}^{N} p(y_t|y_{1..t-1}, h_{\text{sign}}) \tag{2.7}$$

The accumulated error is measured by the cross entropy loss of each time step word y_t. The encoder and decoder modules are both composed of four stacked recurrent layers. Each layer contains 1000 hidden units. The authors (Camgoz et al., 2018) also utilize the AlexNet CNN architecture for feature extraction, which was initially pretrained on Imagenet. Their network is trained until training convergence using the Adam optimization method and which took on average 30 epochs.

Camgoz et al. (2018) proposed two types of approaches: (1) Sign2Text that directly translate sign language videos into spoken language and (2) Sign2Gloss2Text that operates on two stages. The first stage consist of using a forced alignment approach (Koller et al., 2017) as an SLR system to map input frames to gloss level representations $z = \{z_1..z_N\}$. The second stage aims to translate the gloss features to the target sequence through an encoder-decoder structure. The authors found that using the CNN-RNN-HMM (Koller et al., 2017) as a tokenization layer considerably improve translation performance. Figure 2.6 clearly illustrates

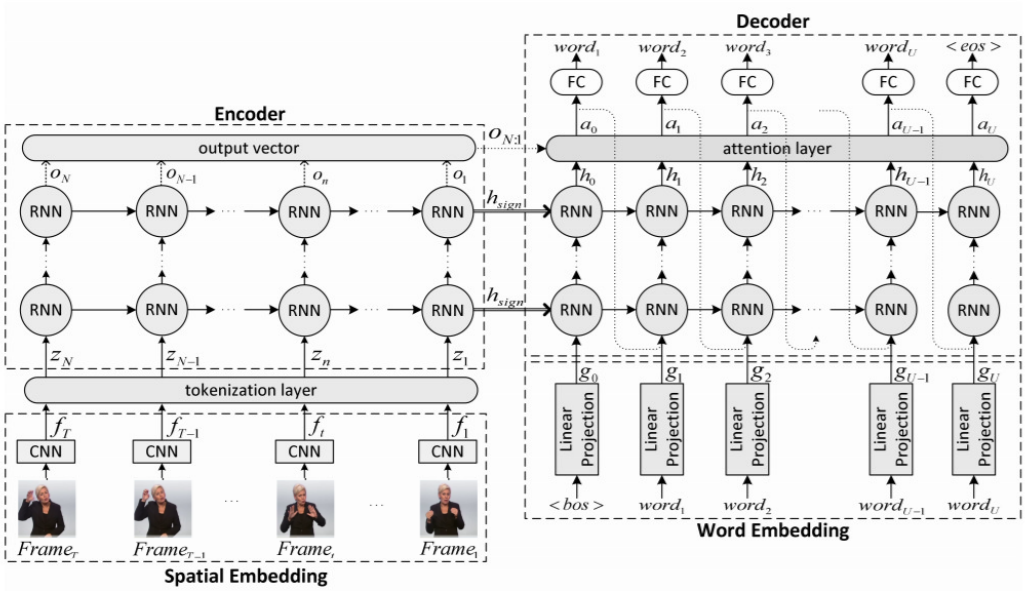

Figure 2.6 An overview of the Sign2Gloss2Text approach proposed by Camgoz et al. (2018).

the Sign2Gloss2Text approach. As mentioned before, recurrent networks usually fail to capture long range dependencies. Alternatively, some works have fixed the underlining problem by utilizing instead the attention mechanism. In the next section, we are going to briefly introduce some close related works that exploit attention for recognition.

2.3 Attention for Recognition

Recent machine translation research (Vaswani et al., 2017) achieved state of the art performance by merely using self-attention[5] , entirely replacing recurrent models. This attention mechanism has been successfully used in a variety of other tasks like video captioning (Chen et al., 2018) in which the authors use an encoder-decoder

[5]The notion of self-attention will be explained in detail in section 3.1

framework with the transformer network backbone to generate a text description of a given video. Another interesting work in the field of reading comprehension (Yu et al., 2018), utilizes the self-attention mechanism to compute the similarities between a pair of context and query words.

A separate study in action recognition (Girdhar et al., 2019) proposes a transformer-based model. They argue that human actions are recognizable from the state of the environment, apart from their own pose. For that, they use self-attention to aggregate features from the spatio-temporal context around the person to correctly classify a person's actions. This inspires us to exploit, in this study, self-attention to incorporate handshape features with their spatio-temporal dependencies. Doing so would add context from the global information to the handshape and ultimately contribute to improving sign recognition. To the best of our knowledge, no previous study has considered doing this for Sign Language Recognition or Translation. Furthermore, instead of the traditional recurrent networks, we utilize self-attention for temporal modelling, which will help efficiently capture temporal dependencies.

2.4 Conclusion

To summarize, in this chapter, we first presented an overview on previous studies applied for the task of Sign Language Recognition (SLR). These works can be generally categorized to three families: (1) approaches that use sensors and gloves for precise and effective tracking of hand and body movements, (2) purely visual methods, which do not require any external tracking devices, and (3) hybrid methods that combine vision-based systems with sensors and gloves. Secondly, we specifically presented a literature review on vision-based studies in the field of CSLR and then in the field of SLT, raising both their advantages and dis-

advantages. Thirdly, we showcase some close-related studies that leverages the self-attention mechanism for recognition.

The following chapter presents the different approaches that we have elaborated to solve both tasks: Continuous Sign Language Recognition and Sign Language Translation. We also propose a novel method to efficiently combine the hand-shapes with their non-manual dependencies for better sign recognition.

CHAPTER III

PROPOSED APPROACH: SIGN TRANSFORMER NETWORK

In this chapter, we describe the approaches that we have devised for both Continuous Sign Language Recognition (CSLR) and Sign Language Translation (SLT). First, in Section 3.2, we introduce our Sign Transformer model for the task of CSLR. The proposed model is based on the Transformer architecture that we describe in Section 3.1. Our model first extracts spatial features from the sign clip frames using 2D CNNs. Then, it employs self-attention for temporal modeling. Finally, the proposed model learns sequence alignment through a CTC layer. In Section 3.3, we add a secondary stream for the cropped handshape sequences and we combine the hand features with their spatio-temporal full-body context.

In Section 3.4, instead of considering the entire context information, we merely attend to information from the handshape local surroundings. This will allow the model to only focus on the required context, discarding unnecessary distant information. We explain the motivation behind such an approach and demonstrate its effectiveness in the experimentation section through quantitative and qualitative findings. In Section 3.5, we adapt our model for the task of Sign Translation by following an encoder-decoder structure. In Section 3.6, we train our latter model on two tasks, using a Hybrid Strategy: generating target translation and also recognizing and aligning Gloss words. Implementation and Training details are

given in Sections 3.7 and 3.8 respectively.

3.1 Background Information on the Transformer Network

For the sake of discussion and for better understandability of our proposed approach, we start by describing the Transformer architecture. This architecture was firstly proposed in (Vaswani et al., 2017) as an alternative for the traditional recurrent models. It is a sequence to sequence (Seq2Seq) architecture, in which an input sequence X is mapped into another output sequence Y. RNN-based models are considered a primary choice for this type of problem. RNN modules are very useful for remembering past context, which is very important for sequence modelling because word order is crucial for sequence understanding

The Transformer works in the same way as the conventional RNN encoder-decoder architecture. The input sentence $X = \{x_1, x_2, .., x_t\}$ is passed through N consecutive encoder layers which generate a sequence of representations $H = \{h_1, h_2, .., h_t\}$. The decoder uses the encoder output alongside the target sequence representation for next word prediction. Decoding is done in a self-regressive way, in which the predictions use outputs from earlier words. A detailed explanation of the overall architecture can be found in the original article (Vaswani et al., 2017).

The Transformer relies solely on self-attention to compute sequence representation, entirely repealing recurrence and convolutional operations. This contributes to considerably reduce the computational cost, as shown in the Table 3.1. As explained in (Vaswani et al., 2017), an attention layer works on a constant number of recurrence $O(1)$, when the sequence length is less than the dimensionality of representation (which is often the case), while a recurrent layer requires $O(n)$ sequential operations. In terms of computational complexity, attentional layers are much faster and more efficient than recurrent layers.

Table 3.1 Complexity comparison between recurrent and attention operations.

Layer type	Complexity	Sequential operation
Self attention	$O(n^2.d)$	$O(1)$
Recurrence	$O(n.d^2)$	$O(n)$

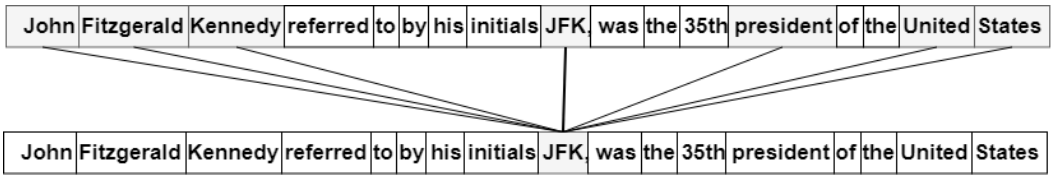

Figure 3.1 A hypothetical example demonstrating the self-attention mechanism for the word token *JFK*.

Attention looks at an input sequence and decides at each state which other parts of the sequence are important. This will provide contextual information about the current state. The Transformer uses a special type of attention, called self-attention, which means that we apply attention to the sequence itself. This notion can be easily understood from Figure 3.1. As seen from this figure, the word *JFK* takes most of the contextual information from the words *John*, *Fitzgerald* and *Kennedy*, since these are the initials of this word, but also the words *president* and *United States*, because they describe more the nature of the word. It should also be noted that it makes sense that the word has a strong similarity with itself.

As shown in Figure 3.2, the overall architecture of the Transformer follows an encoder-decoder structure (the encoder and the decoder, represented respectively in the left and right halves of the architecture). The model uses stacked attention layers followed by feed forward (linear) layers. We explain in detail in the following sections, the different Transformer Network components.

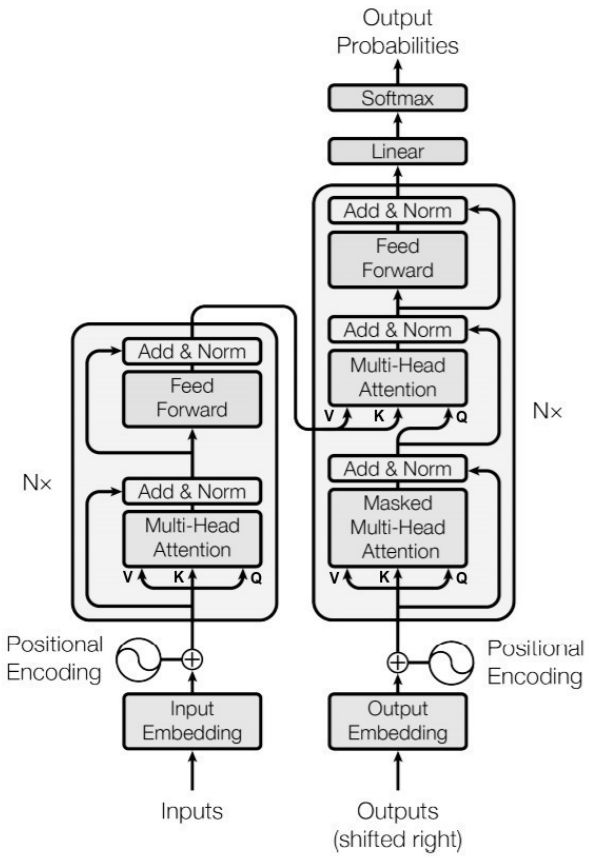

Figure 3.2 An overview of the Transformer Network from (Vaswani et al., 2017). The Input and Output embeddings use a word2vec network.

.

3.1.1 Word Embedding

Learning word embeddings represent the first crucial step in machine translation tasks. This usually consists of using *word2vec* method to learn word representations. The *word2vec* technique aim to learn how to represent words by estimating the probability of having a certain context (being surrounded by neighbouring words). This can generally be achieved by using a simple neural network to learn the word embeddings.

An alternative and naive approach consists of using a *one-hot* encoding, which consists of representing the words through binary representations. This is achieved by setting a value of 1 to the corresponding word index in the Bag of words and 0s elsewhere. This usually leads to a representation vector with a high dimensionality when the Bag of Words is considered large. Alternatively, *word2vec* is capable of producing word embeddings with lower dimensionality. As shown in Figure 3.2, the Tranformer uses two (*word2vec*) embeddings layers: *Input Embedding* and *Output Embedding* which correspond respectively to encoding of input and target sequences.

3.1.2 Attention Mechanism

The Transformer first linearly project the input sequence to three identical sequences: Query Q, Key K and Value V vectors. The architecture then uses the scaled dot product (see Equation 3.1) to calculate the attention scores (similarity) of each feature with the rest of the features in the sequence by multiplying Q with K. The attention scores represent the degree of focus (attention) applied on other words when encoding the current word. The values are then scaled using the square root of the representation dimension d_K. According to the authors (Vaswani et al., 2017), this helps prevent numerical instabilities. The output is passed to a *Softmax* to normalize the scores and to make sure they add up to a total value of 1. The resulting vector is then calculated as the weighted sum of the values V conditionally on the attention scores. Doing this will make sure to keep the important words and get rid of the irrelevant words.

$$Attention(Q, K, V) = softmax\Big(\frac{QK^T}{\sqrt{d_k}}\Big)V \tag{3.1}$$

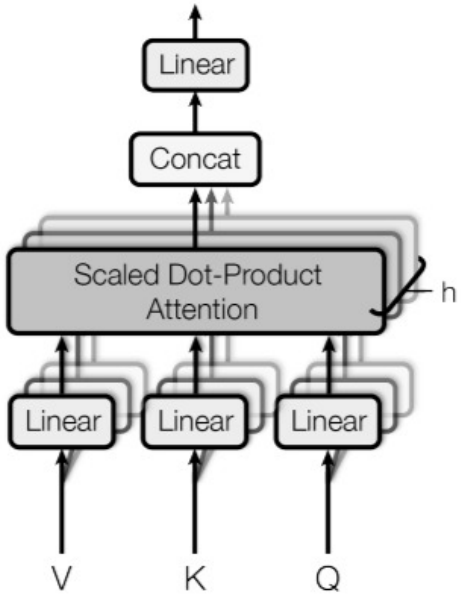

Figure 3.3 Multi-Head Attention from (Vaswani et al., 2017).

3.1.3 Multi-Head Attention

Instead of applying the attention mechanism to the entire representation space, word representations are first projected linearly to the vectors Q, K and V with a lower dimensionality, h times. In this case, we apply the dot product to the different set of representations to create the h attention vectors. We concatenate the h attention representations to produce the final attention vector, which is then passed to a linear layer. This procedure is illustrated in Figure 3.3.

Instead of having a single attention head, Q, K and V are divided into several attention heads. According to Vaswani et al. (2017), this allows the model to learn information from different representation spaces. When following a multi-head attention strategy, each head has a reduced dimensionality, so the total cost of

computation is the same as using a single attention head with full dimensionality[1].

3.1.4 Positional Encoding

Moving away from recurrent and convolutional models, we end up losing positional information. Accordingly, to preserve the sequence order, the authors incorporate positional encodings to the feature representations through the use of a sinusoid-wave-based function. The positional encoding vector is added to the representation vector of the input sequence, as illustrated in Figure 3.2. The positional encoding vector must have the same dimensionality as the input representation, so that they can be added together. According to the authors (Vaswani et al., 2017), this allowed the model to easily learn the information of absolute and relative positions.

3.1.5 Encoder Module

In summary, the encoder module consists, first of all, of an embedding layer, which takes the input sequence of words and produces the words embeddings using a regular linear layer. The representation vector is then aggregated with the positional encoding, in order to add the position information. The resulting vector is transmitted to N consecutive encoder stacks, where each module contains a multi-head attention layer, followed by a two layered neural network. Each of these sublayers has a residual connection around it followed by a normalization layer (Ba et al., 2016), as described by Equation 3.2. Residual connections help avoid the problem of vanishing gradient in deep neural networks. As for the normalization layer, it has the effect of stabilizing the learning process.

---

[1] `https://www.tensorflow.org/tutorials/text/transformer`

$$x = \text{Normalization Layer}(x + \text{Encoder stack}(x)) \tag{3.2}$$

3.1.6 Decoder Module

Similarly, the decoder uses the same components as the encoder. However, the only difference is that the decoder uses two types of multi-head attention. The first is the masked multi-head attention applied to the target output sequence, which uses an anticipation mask to avoid looking at future sequence words. The second is a regular multi-head attention and which consider the output of the encoder as the K and V representations. On the other hand, the query vector Q represent the linear projection of the output of the previous masked multi-head attention sublayer. The input vector is applied to N consecutive decoder stacks to produce the output representation. The resulting vector is then passed to a linear layer followed by a Softmax to generate the target word probabilities. This is clearly illustrated in Figure 3.2.

3.2 The Proposed Sign Transformer Network

We propose a Transformer-based model for the task of Continuous Sign Language Recognition, in which we refer to it as the Sign Transformer Network. Our model is designed to accept a clip as input and accordingly produce the output words in a spoken language, making it a sequence to sequence task. Subsequently, our model is trained to recognize and time-align the target glosses from the sign language clip.

The first step consist of learning the sign images representation. For this, we use a regular 2D CNNs to extract the spatial representations $H = \{h_1, h_2, ..h_{T_x}\}$ from the individual images of the video $X = \{x_1, x_2, ..x_{T_x}\}$, as can be seen in the

following equation:

$$h_t = \text{Spatial Representation}(x_t) \tag{3.3}$$

In recognition tasks, framewise aligning each timestep in the input sequence $X = \{x_1, x_2, ..x_T\}$ with its corresponding ground-truth annotation can be time-consuming and computationally expensive to train. This is especially true in the case of CSLR as the input sequence's length can be much more superior than that of the target sequence, with $T_x >> T_y$. The Connectionist Temporal Classification (CTC) proposed by Graves et al. (2006), is a popular solution to overcome such a problem. The CTC considers all possible alignment in each timestep. It introduces an extended target word vocabulary $L' = L \cup \{\epsilon\}$ where ϵ represents the blank label that accounts for silence and which is to be removed from the output while decoding.

The CTC objective loss function is defined as $\mathcal{L}_{CTC} = -\log P(Y|X)$ where $P(Y|X)$ is the conditional probability, and can be described as:

$$P(Y|X) = \sum_{a \in L'} \prod_{1 \leq t \leq T} P_t(a_t|X) \tag{3.4}$$

Computing the score for each alignment can be computationally intensive. To alleviate this, CTC uses a dynamic programming algorithm to efficiently and promptly compute $\mathcal{L}_{CTC}$ (Hannun, 2017).

Recurrent networks are usually used to estimate posteriors $P_t(a_t|X)$ for each timestep $t_{1 \leq t \leq T}$, in which a_t represent the latent representation produced by the network. However, RNNs normally fail to capture global dependencies, especially when dealing with long sequences. Therefore, for sequence to sequence model-

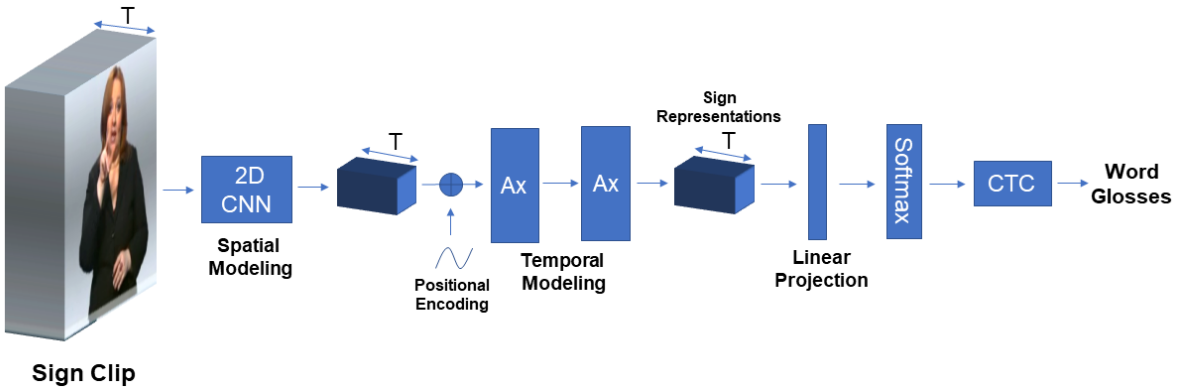

Figure 3.4 Overview of our Sign Transformer Network architecture.

ing, we replace traditional recurrent methods with self-attention, as they have proven to be more resilient to the vanishing gradient problem. Similarly, as used in the encoder side of the Transformer network, we use stacked self-attention layers to learn temporal dependencies and map the input frame features viewed as $X \in R^{T \times D}$ to another sequence of equal length, in which D represents the feature dimentionality.

In summary, our model produces image representations through a 2D CNN on the individual frames of a given sign clip. The resulting matrix is passed to succeeding Ax Attention units followed by a linear projection and a Softmax to output the word probabilities and finally a CTC layer in order to generate the gloss words. The overall architecture is highlighted in Figure 3.4. Similarly to the original Transformer architecture, the Ax Unit represents the encoder stack, which is composed of a multi-head self-attention mechanism followed by a fully connected layer. We apply a layer Norm (Ba et al., 2016) first and then a residual connection for both the multi-head attention and the fully connected layers as opposed to the original Transformer paper.

3.3 Context-Hand Attention Layer

Although our model may be able to learn all sign language modalities simply from the full-frame sequences, it would be of special interest to investigate the impact of fusing hand and global (full-body) features. Handshapes require spatio-temporal information; as a result, each hand feature needs to attend to context across time and not just the current context frame. Respectively, we design two-stream sub-networks: a Context Stream that is trained on the full-frame sequences and a Hand Stream which is trained on the cropped hand images. This will allow the first to learn to recognize global information and as a result the overall context of the signs. The second will be equipped to merely learn the handshape information. We combine the two through a third module that we refer to as a Context-Hand Attention Layer, in which the key and value features of shape ($T \times d_k$) are computed as linear projections of the full-frame sequence representation, while the Query features Q_{hand} of shape ($T' \times d_k$) are obtained through the hand sequence projections.

In the Context-Hand attention layer, we apply the dot-product to get the attention values $A_{hand/context}$ of the Q_{hand} features over the $K_{context}$ features. Then, we perform a weighted averaging of the resulting matrix over the $V_{context}$ features to get the updated hand representations Q'_{hand}. We apply a Layer-Norm operation on the hand query and then add it to the original hand features. The resulting feature is passed to another Norm Layer and a 2 layer Feed-Forward Network (FFN), to eventually produce the final hand query Q''_{hand}. The process is described in the following set of equations:

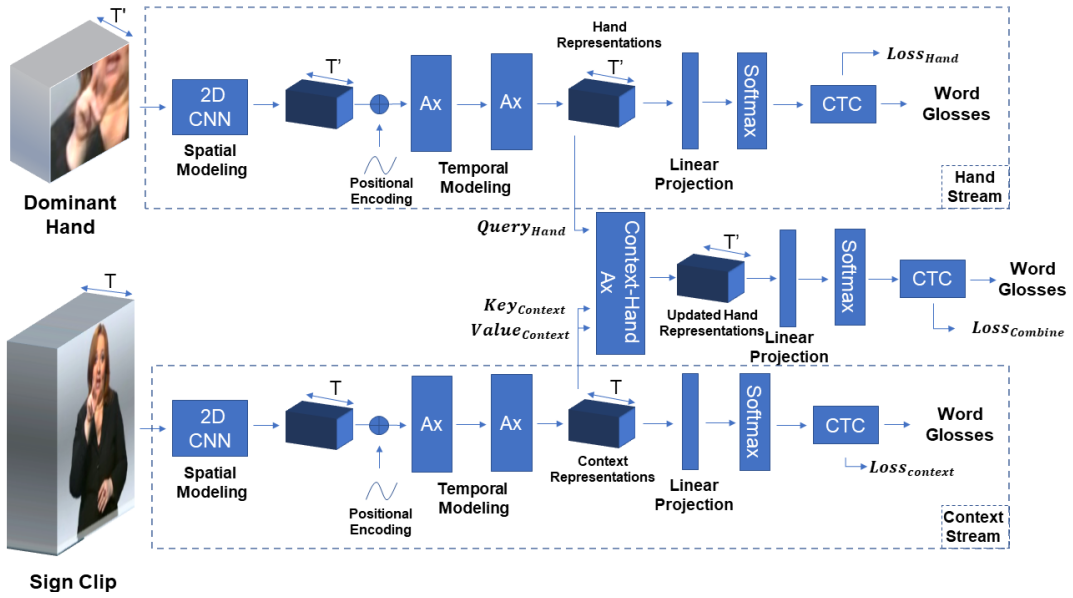

Figure 3.5 Combination of both the full-frame and the handshape streams through a Context-Hand Attention layer.

$$A_{hand/context} = Softmax\left(\frac{Q_{hand}K_{context}^{T}}{\sqrt{d_k}}\right)V_{context} \tag{3.5}$$

$$Q'_{hand} = Q_{hand} + \text{Normalization Layer}(A_{hand/context}) \tag{3.6}$$

$$Q''_{hand} = Q'_{hand} + \text{Normalization Layer}(\text{FFN}(Q'_{hand})) \tag{3.7}$$

$$\tag{3.8}$$

The updated hand features Q''_{hand} are passed to a linear layer followed by a Softmax and then to a CTC to generate the word glosses. Similarly to Camgoz et al. (2017), the overall network is trained end-to-end using the three loss layers: $\mathcal{L}_{hand}$, $\mathcal{L}_{context}$ and $\mathcal{L}_{combine}$. The proposed architecture is illustrated in Figure 3.5.

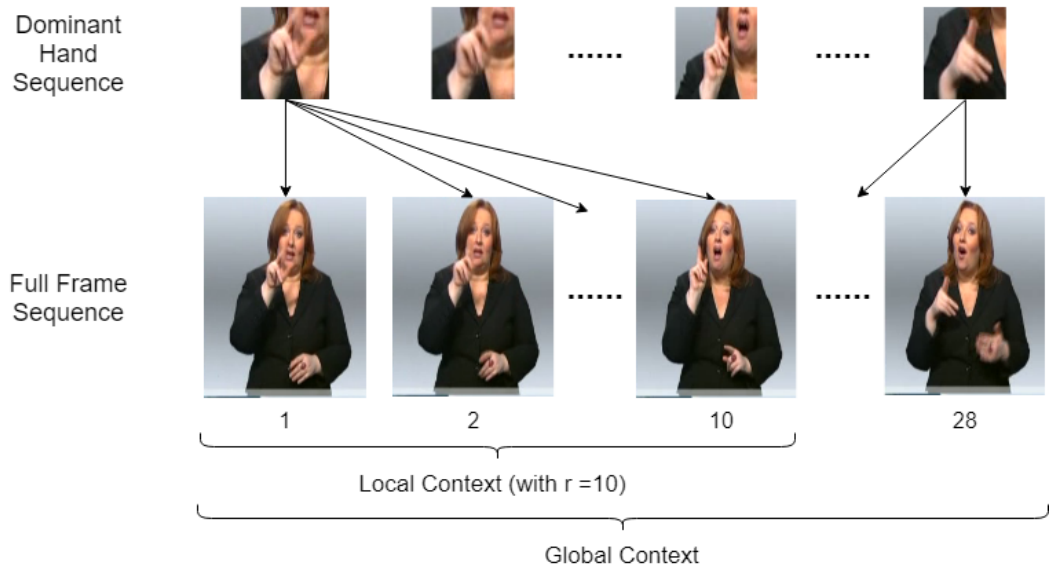

Figure 3.6 Relative local context mechanism in the Context/Hand Attention layer.

3.4 Relative Local Context Masking

Typically, self-attention allows the model to observe the entire sequence when measuring the attention at each sequence step. This can be very helpful for recognition, as Sign Language possesses a complex temporal structure and having this type of attention can help capture entangled dependencies. However, while handshapes expect temporal context, it is unclear to what extent this context may span. Required information can be limited to a local temporal window. So, instead of taking information from the entire sequence, we set a relative window and we only attend to the context around the current handshape frame. A similar idea has been explored for Acoustic Models (Sperber et al., 2018).

This can be especially beneficial when dealing with long sequences where early handshapes don't usually require information from distant context frames. We

set a relative window of size r and we apply a mask $M_{rel} \in R^{T' \times T}$ to the attention scores to mask out unwanted values in which:

$$M_{rel} = \begin{cases} 1 & \text{if } |j-k| < r \\ 0 & \text{else} \end{cases}$$

This process is also described in Figure 3.6. As can be seen from this figure, the hand shape in each state considers the contextual information from the images in the relative window $r \leq 10$ and ignores that of the distant images.

Having attention restricted to a local region may result in missing some required long-range dependencies. Therefore, we avoid using such a mechanism on self-attention layers in both the context and hand streams as they are responsible for learning global information from the overall sequences. And we solely use it in the Context-Hand attention layer to incorporate handshapes to their local related contexts.

3.5 Encoder-Decoder Self-Attention

We re-purpose our Sign Transformer Network for the task of sign translation by including a decoder module and following an encoder-decoder structure. The encoder part takes the image embeddings and produces a latent sequence representation to be passed to the decoder to generate the target spoken sentences. The model follows the conventional transformer network architecture. The decoding is carried on in an auto-regressive manner. In each time-step, the model predicts the target next $word_i$ by only seeing the previous $word_{1..i}$. On that account, we add two special tokens: a $< sos >$ that marks the beginning of a sentence and a $< eos >$ that indicates the end of a sentence and which halts the decoding proce-

dure. Since self-attention allows the model to see the entire sequence and hence to prevent it from peaking at the expected future target words, the model uses a look-ahead mask. In each time-step, the decoder predicts the posterior $P_{dec}(Y|X)$ which is then used to measure the objective loss function $\mathcal{L}_{dec} = -\log P_{dec}(Y|X)$, where $P_{dec}(Y|X)$ is given by:

$$P_{dec}(Y|X) = \prod_{i=1}^{L} p(y_i|y_1, y_2, .., y_{i-1}) \tag{3.9}$$

3.6 Hybrid Training

Apart from $\mathcal{L}_{dec}$, we add a second objective function $\mathcal{L}_{CTC}$ to produce gloss words. Our model is trained on two different tasks: generating target translation, and also recognizing and aligning gloss words. CTC enforces learning the monotonic alignment between frame and gloss sequences and helping to speed up the training process (Watanabe et al., 2017). Since we are training with a hybrid strategy , our model has to learn a representation for both tasks and hence reduces the risk of overfitting (Ruder, 2017). As a result, this can help better estimate the sequence alignment and remarkably improve the translation performance. We refer to this approach as Hybrid STN and it is clearly illustrated in Figure 3.7. We use stacked Ax layers to encode the input frame sequences and as described in the original Transformer paper (Vaswani et al., 2017), we add a multi-head attention sub-layer in our decoder attention stacks to perform attention on the output of the encoder. It is important to note that the gloss and translation sequences may share the same semantics but they use different target word labels.

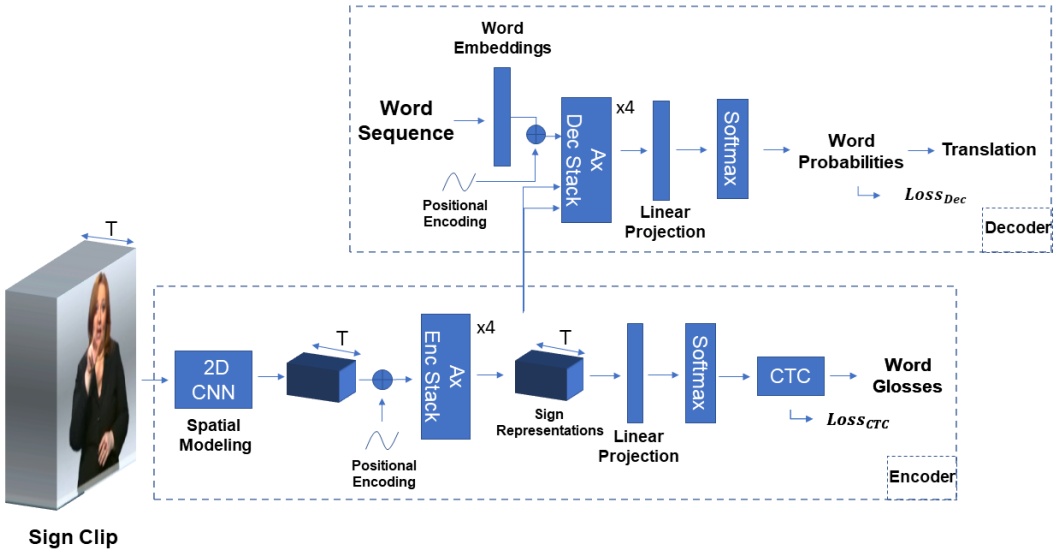

Figure 3.7 Overview of our Hybrid STN approach.

3.7 Implementation Details

We start by extracting T frames (mostly 64) from the original video clip, in order to reduce computational complexity and discard frame redundancy. We resize the input full-frame images to a spatial resolution of 224×224 and we normalize them by subtracting the dataset's image mean which has been provided by Camgoz et al. (2018). We also resize the hand patches to a lower resolution of 112×112. For spatial embeddings for both full-frame and hand images, we utilize the MobileNetV2 (Sandler et al., 2018) architecture, due to its low-latency and high efficiency as a feature extractor. We drop its last fully connected layers and use the rest of the layers for feature embeddings. All our networks use a feature dimensionality of $d_k = 128$ and 10 attention heads.

For networks that are trained for SLR, we use 2 layers of self-attention. As for the encoder-decoder networks that are trained for SLT, we instead use 4 layers. The rest follow the default recommended setup provided by Vaswani et al. (2017): A

2 layer position-wise feed-forward network with a dimensionality of $d_{ff} = 2,048$ and a positional encoding sinusoidal function that encodes relative and absolute information. We based our networks' implementations on (Klein et al., 2017).

3.8 Training Details

We initialize all of our networks' layers using Xavier from (Glorot & Bengio, 2010), except for the image embedding layers which have been pre-trained on ImageNet (Deng et al., 2009). We use the Adam Kingma & Ba (2014) optimization method with its default parameters and a learning rate of 10^{-4} for our CSLR experiments and 10^{-5} for SLT. We also use gradient clipping with a threshold of 1 to avoid the exploding gradients problem. We avoid overfitting by employing data augmentation through random x-y image translation and using a dropout probability of 0.3. Dropout is a regularization technique that prevents over-fitting when training. This involves randomly ignoring units (neurons) in a neural network, which reduces the interdependence between neurons during training (Srivastava et al., 2014).

In the case of SLT experiments, we use label smoothing (Müller et al., 2019) of $\alpha = 0.1$ instead of the cross-entropy as it offers a regularization effect and results in a slightly better performance. Label smoothing is usually applied for multi-class classification problems, in which the cross-entropy is used as the loss function. Normally in such tasks, we minimize the objective function by training the network to maximize the log-likelihood of the target class. Consequently, the model will be trained on predicting hard targets expecting true labels to have a value of 1, while a value of 0 otherwise. In classification tasks, ground truth training labels can have erroneous labels. As a result, training the model on this misguiding labels can lead to overfitting.

To overcome this problem and instead of training on hard targets, we relax the model confidence by reducing the loss target predictions of 1s by a small value of α . And respectively increase the target predicting of 0s by α. As can be seen in the following equation:

$$\text{Soft Labels} = \text{Hard Labels} * (1 - \alpha) + \alpha/\text{number of classes} \tag{3.10}$$

The Hard Labels represent the one hot class prediction of all the target vocabulary tokens. This method is referred to as Label Smoothing, it is used to prevent the model from becoming over-confident during training, which can then lead for better generalization (Müller et al., 2019).

Following Camgoz et al. (2018), that found that having a small batch size increases translation performance; we likewise train all our models using a batch size of 2. Since we are performing batch training, and considering that input clips may vary in size, sequences in every batch are padded to equal lengths (maximum sequence length in the batch). As a result, we utilize a mask on the input sequences to avoid attending on padded elements. In the case where we use relative local masking, we merge both the padding mask with M_{rel}. So that, besides the distant elements, we also avoid looking at the padded elements. We train our networks for an average of 100 epochs as they stop improving or until train perplexity convergence. We evaluate our model every epoch on the validation set and report the best performing model.

CHAPTER IV

EXPERIMENTAL RESULTS

In this chapter, we perform a set of experiments to evaluate the efficacy of our proposed approach on the two distinct tasks: Continuous Sign Language Recognition (in Section 4.1) and Sign Language Translation (in Section 4.3). Accordingly, for each task, we introduce the dataset which we use to compare our findings with the state-of-the-art. In Section 4.2, we empirically validate our STN architecture by comparing it with different hyper-parameter choices.

4.1 CSLR Experiments

For our first set of experiments, we use the RWTH-PHOENIX-Weather 2014 corpus for the task of CSLR. The dataset offers sign clips (sequence of frames) and their corresponding gloss annotation. It contains 9 different signers and a training set of 5,672 sequences, thereby helping with model generalization. The textual annotation consists of a vocabulary L of 1,231 words, with only 410 singletons (words that appears only once in the training set). The dataset provides full frames alongside the cropped dominant hand (right hand), which yields a favourable test bed for our central approach.

4.1.1 Quantitative Results

As shown in Table 4.1, we start by comparing our approaches' results for the RWTH-PHOENIX-Weather 2014 dataset on both the validation and test sets and using the Word Error Rate:

$$WER = \frac{\#\text{deletions} + \#\text{insertions} + \#\text{substitutions}}{\#\text{number of reference observations}} \tag{4.1}$$

WER is a metric that is usually used to evaluate speech recognition systems. It operates by adding up the substitutions, insertions, and deletions that occur in the recognized sequence. A substitution occurs when a word gets totally replaced by a different word. An insertion arises when a non-existing word is added. And deletion is when a word is not correctly recognized.

Instead of using a greedy approach for decoding, in which we simply take the word with the highest probability, we consider using a beam decoder with a beam width of 10 to get our final output sequence in both training and evaluation. Beam search decoding is mostly used for machine translation tasks. It considers the k best output sequences, in which k represents the beam width. This is carried on by considering the combinations of the k most likely words at each time step. This usually produces better prediction results than when using the greedy search approach. We found in our experiments that having a higher beam value did not improve performance and rather tremendously increased decoding time. Taking this into account, we use the same beam search strategy for all our experiments.

The goal of this first set of experiments is to evaluate and compare our different STN variants in order to showcase the effectiveness of our central approach (STN with the hand stream). And, then, compare our best performing approach with the state-of-the-art previous works to prove our proposed model superiority.

Table 4.1 Comparison of our Sign Transformer Network variants on RWTH-PHOENIX-Weather 2014 in Word Error Rate % (the lower the better).

	Dev	Test
STN	35.33	35.45
+ Hand Stream	33.68	34.12
+ **Relative Local Masking**	**32.74**	**33.29**

Table 4.2 Comparison of our Sign Transformer Network with the Full Frame Word SubUnets from Camgoz et al. (2017) in Word Error Rate % (the lower the better). Both models are only trained on full frame images and use CTC for alignment.

	Dev	Test
STN	**35.3**	**35.4**
Word SubUNet Camgoz et al. (2017)	43.9	43.1

To this end, we consider at first our regular STN model. Similarly to the Full Frame Word SubUnets architecture from (Camgoz et al., 2017), we train our first STN model on full frames, utilizing CNNs for spatial embeddings, CTC for alignment, and replacing BLSTM recurrent layers with self-attention. As can be seen in Table 4.2, our approach significantly surpasses (Camgoz et al., 2017), proving the advantage of using self-attention over RNN-based networks for temporal modelling.

In the next step, we study the effect of incorporating handshape features with full frame context information by adding a hand stream module and a Context-Hand attention layer. This leads to a considerable boost in performance as shown in Table 4.1. But more importantly and as manifested in Figure 4.1, adding

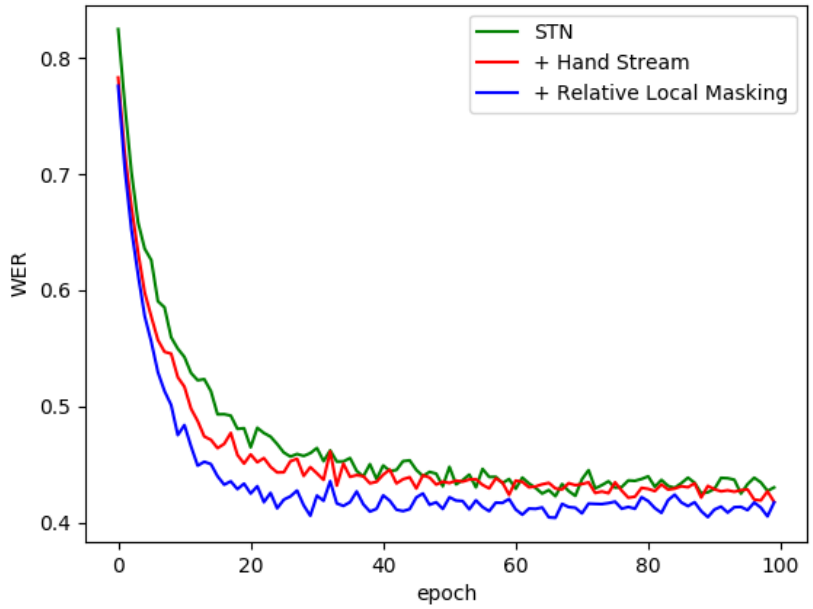

Figure 4.1 The Word Error Rate learning curve of our STN variants for the task of CSLR on the RWTH-PHOENIX-Weather 2014 dataset.

handshape features also improves training and accelerates model convergence. This empirically showcases the usefulness of combining the dominant hand with the overall context derived from the nonmanual components of the sign. Note that both hand and full-frame sequences have equal lengths ($T = T'$).

Finally, and as demonstrated in Table 4.1, limiting attention and fusing handshapes with their local related context leads to a significant improvement in performance, surpassing all our previous approaches. This supports the idea that handshapes, when accompanied by their proper contexts, can notably help in recognition.

Table 4.3 shows a comparison of our best performing model with previous publications. For a fair comparison, we only compare with competing methods that

Table 4.3 Comparison of our best performing model with the published prior studies on RWTH-PHOENIX-Weather 2014 in Word Error Rate % (the lower the better).

		Dev	Test
ours	**STN**	**32.7**	**33.2**
Huang et al. (2018)	3D two-Stream CNN/HAN	-	38.3
Cui et al. (2017)	CNN-RNN	39.4	38.7
Koller et al. (2016)	Hybrid CNN-HMM	38.3	38.8
Camgoz et al. (2017)	CNN-BLSTM-HMM(LM)	40.8	40.7
Camgoz et al. (2017)	CNN-BLSTM-CTC	43.1	42.1
Koller et al. (2016)	CNN-HMM	47.1	45.1
Koller et al. (2015)	Statistical approach	57.3	55.6

are trained on ground truth target glosses and we don't account for approaches that rely on forced alignment to train on per-frame labels (Koller et al., 2017, 2019). Accordingly, we opted to compare in this study with methods that employ similar training configurations, in order to showcase the efficiency of our proposed approach.

4.1.2 Qualitative Analysis

Apart from obtaining promising results, it is of interest to visualize the areas on which the model focuses for signs prediction. Accordingly, we use Grad-Cam from (Rs et al., 2017) to produce the localization heatmaps in which bright pixels represent a positive influence and have great importance on the predicted sign. We extract the heatmaps from the frame embedding activations, highlighting important regions, which the model uses to predict a particular sign.

Respectively, we first compute the gradients $\frac{\partial y^c}{\partial A}$ for a predicted word y^c with respect to a feature map A^k from the CNN activations (before the pooling operation of the MobileNetV2 architecture). By averaging these gradients, we obtain the attention weights a_k^c, which represent the importance of the feature map k for a given predicted word y^c. To produce the heatmap features, we multiply the importance weights a_k^c with the activations A^k, followed by a ReLU operation [1]. This is clearly expressed in the following equation:

$$\text{Heatmap} = \text{ReLU}\left(\sum_k a_k^c A^k\right) \tag{4.2}$$

As can be seen in Figure 4.2, our models primarily focus on the dominant hand (right hand) and the face area which reinforces the intuition that our proposed model is able to identify the essential components for sign interpretations.

4.2 Hyperparameters Validation

Our proposed STN models are mainly trained with a learning rate of 0.0001 and use a 10-head, 2-layer setup by default. In this section, we compare our default architecture with other hyper-parameter settings, focusing primarily on key elements like the learning rate, number of attention layers and number of attention heads.

As can be seen from Figure 4.3, training our proposed model with higher learning rates ($lr = 0.1$ and $lr = 0.01$) results in gradient exploding. Consequently, it produces very high Word Error Rates that remains constant and non-improving all long training. On the other hand, training with a very low learning rate

[1] ReLU (Rectified Linear Units) is an activation function that simply replaces negative values with zeros.

Figure 4.2 The top sequence is the output results of our STN network. The middle is for STN with the hand stream and the bottom is for STN with hand stream and the local context masking. Note that this example is randomly chosen and not cherry-picked.

($lr = 0.00001$) slows down training and decreases convergence rate. Secondly, and as illustrated in Figure 4.4, increasing the amount of attention layers results in overfitting and ultimately reduce model performance. Contrarily, training with a 2-layered setup in our default STN results in better recognition performance than when using a single attention layer.

Finally, and as depicted in Figure 4.5, training with a multi-head strategy can greatly improve model performance. This confirms the idea raised by Vaswani et al. (2017), which states that doing so allows the model to learn from different representation spaces. Also, as seen from the last figure, increasing the amount of attention heads doesn't always enhance prediction. Our default STN setup with only 10 attention heads is slightly better than that with 20 heads. It is important to note that the amount of attention heads must evenly divide the full-dimensionality space (which is in our case $D = 1,280$). For instance, when using 10 attention heads, each one has a reduced dimensionality of $d_k = 128$

$(D/10 = d_k)$.

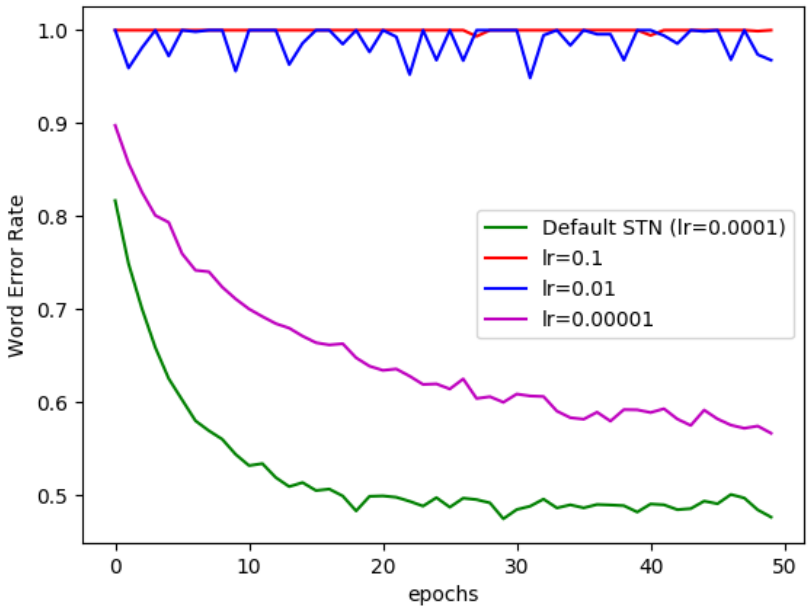

Figure 4.3 Comparison of our STN model with different learning rate settings.

4.3 SLT Experiments

Given the satisfactory performance of using self-attention for CSLR, for our second set of experiments, we consider the dataset RWTH-PHOENIX-Weather 2014T for the task of Sign Language Translation. We compare our results with the baseline defined by Camgoz et al. (2018). Unlike the previous dataset, this one provides both gloss and translation annotations. The dataset consists of an extended vocabulary of 2,887 distinctive words and a train set of 7,096 sequences.

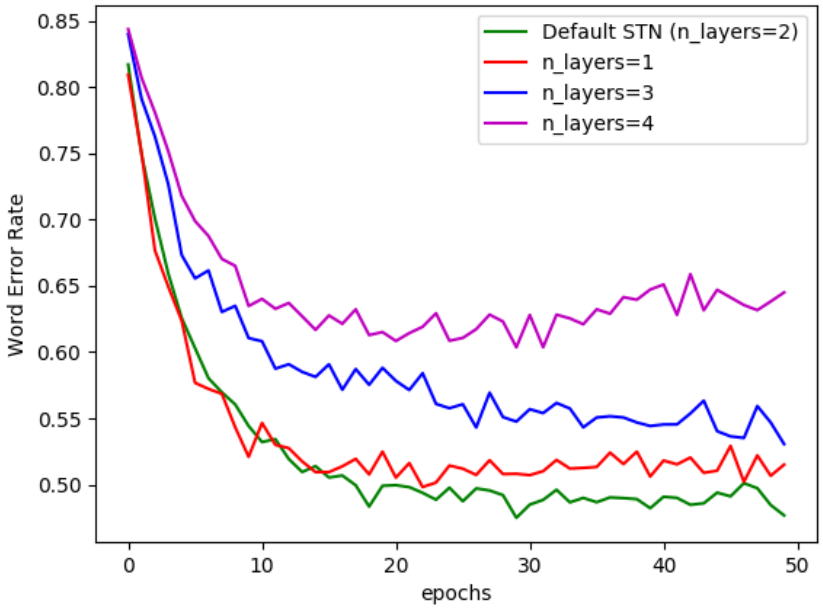

Figure 4.4 Comparison of our STN model with different number of attention layers.

4.3.1 Quantitative Results

We apply a beam search of size 3 to produce the output translations, which we found to be the optimal beam width for SLT. To evaluate our models' translations, we use the BLEU (Papineni et al., 2002) and ROUGE (Lin, 2004) metrics. These metrics are usually used to evaluate text output for NLP applications. The BLEU score measures how many words in the generated output appeared in the reference text. On the contrary, ROUGE compute how many words in the reference translation exists in the generated output. First, we consider using an encoder-decoder framework, mapping input sign clips to their corresponding spoken translation. Afterward and as specified in (Camgoz et al., 2018), the dataset might be too small to allow our Sign Transformer network to generalize well. Respectively, we

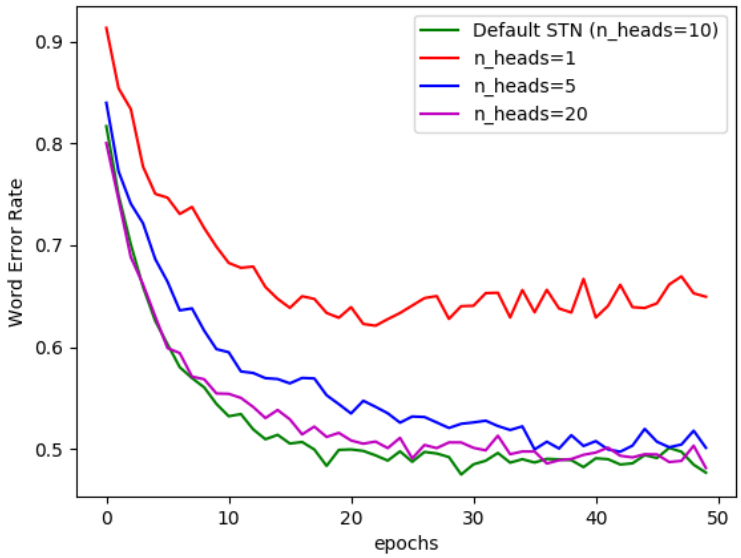

Figure 4.5 Comparison of our STN model with different number of attention heads.

train our model on both gloss words and translation prediction. We refer to this approach as Hybrid STN. As depicted in Figure 4.6, this remarkably improves model performance and speeds up training convergence.

We contrast our findings with (Camgoz et al., 2018) models in Table 4.4. Firstly, we compare our encoder-decoder STN with their Sign2Text (S2T) that directly maps sign language clips to their corresponding spoken language translations. Our proposed model achieves a compelling margin improvement of 2.6% in BLUE-4. Secondly, we compare our Hybrid STN with their second model Sign2Gloss2Text (S2G2T), in which both use an additional gloss level supervision that significantly helps with training. As discussed in the Related Work chapter, S2G2T uses a forced alignment approach (Koller et al., 2017) as a Tokenization Layer that maps input frames to gloss level representations and performs a gloss-level prediction. Contrarily, our model directly maps sign language frames to output translation.

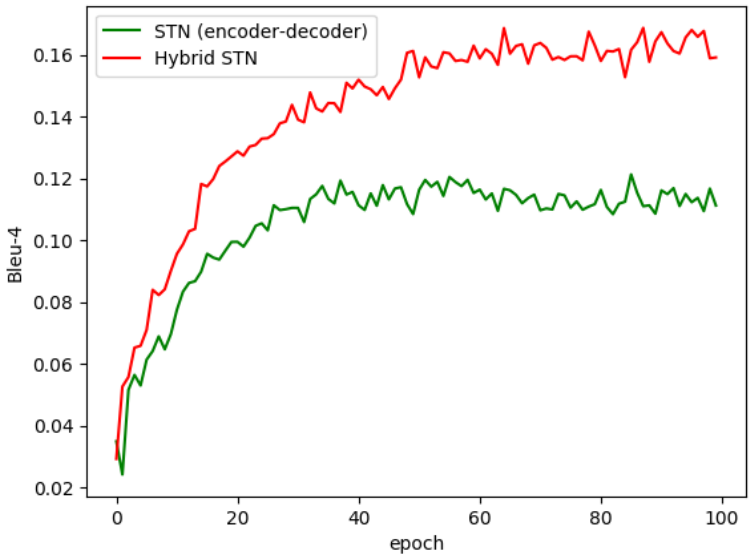

Figure 4.6 The BLEU-4 score learning curve of our encoder-decoder STN models for the task of SLT on the RWTH-PHOENIX-Weather 2014T dataset.

Table 4.4 Comparison of our encoder-decoder Sign Transformer networks with the baseline results (Camgoz et al., 2018) on RWTH-PHOENIX-Weather 2014T using BLEU and ROUGE scores % (the higher the better).

	Dev					Test				
	Rouge	Bleu-1	Bleu-2	Bleu-3	Bleu-4	Rouge	Bleu-1	Bleu-2	Bleu-3	Bleu-4
STN (Encoder-Decoder)	**35.29**	**35.49**	**22.10**	**15.58**	**12.14**	**34.53**	**35.59**	**22.15**	**15.70**	**12.18**
S2T (Camgoz et al., 2018)	31.80	31.87	19.11	13.16	9.94	31.80	32.24	19.03	12.83	9.58
Hybrid STN	41.27	39.74	27.41	20.61	16.54	41.18	40.82	28.20	21.16	16.93
S2G2T (Camgoz et al., 2018)	**44.14**	**42.88**	**30.30**	**23.02**	**18.40**	**43.80**	**43.29**	**30.39**	**22.82**	**18.13**

4.3.2 Qualitative Analysis

Even though metrics like BLEU or ROUGE are useful indicators to measure translation quality, they can be misleading with regard to overall accuracy (Kirti Vashee, 2019). As a result, to better qualify our outputs, we analyze in Table 4.5 our translation results with those shared by Camgoz et al. (2018). As can be seen from the translation samples, our STN hybrid yields better quality translations that are semantically closer to the Ground Truth sentences than those generated by the S2G2T model (Camgoz et al., 2018). This can be especially seen in the first example sentence, in which S2G2T completely fails to predict the target word *kuhler* (cooler) and instead predict the word *wechselhaft* (unpredictable) twice. As for the second sentence example, the S2G2T model completely misses the context of the sentence as opposed to ours.

Table 4.5 Comparison of the output translations from our encoder-decoder STN models to those shared by Camgoz et al. (2018).

Approach	Translation
Ground Truth	die neue woche beginnt wechselhaft und kuhler.
	(the new week starts unpredictable and cooler.)
STN (ours)	die neue woche startet wechselhaft und wieder kühler .
	(the new week starts unpredictable and cooler again.)
Hybrid STN (ours)	die neue woche beginnt wechselhaft und wieder kühler .
	(the new week starts unpredictable and cooler again.)
S2T from (Camgoz et al., 2018)	am montag uberall wechselhaft und kuhler.
	(on monday everywhere unpredictable and cooler.)
S2G2T from (Camgoz et al., 2018)	die neue woche beginnt wechselhaft und wechselhaft.
	(the new week starts unpredictable and unpredictable.)
Ground Truth	im suden und sudwesten gebietsweise regen sonst recht freundlich.
	(in the south and southwest locally rain otherwise quite friendly.)
STN (ours)	am sonntag gebietsweise regen im bergland ist es sehr windig.
	(It is very windy on Sunday in the mountainous regions.)
Hybrid STN (ours)	im südwesten regnet es zum teil kräftig im osten sonst ist es freundlich.
	(in the southwest it is raining heavily in the east, otherwise it is friendly.)
S2T from (Camgoz et al., 2018)	von der sudhalfte beginnt es vielerorts.
	(from the southpart it starts in many places.)
S2G2T from (Camgoz et al., 2018)	am freundlichsten wird es im suden.
	(the friendliest it will be in the south.)
Ground Truth	und nun die wettervorhersage für morgen samstag den zweiten april .
	(and now the weatherforecast for tomorrow saturday the second april.)
STN (ours)	und nun die wettervorhersage für morgen samstag den fünften märz .
	(and now the weather forecast for tomorrow saturday the fifth of march.)
Hybrid STN (ours)	und nun die wettervorhersage für morgen samstag den zweiten märz.
	(and now the weather forecast for tomorrow saturday the second of march.)
S2T from (Camgoz et al., 2018)	und nun die wettervorhersage fur morgen freitag den sechsundzwanzigsten märz.
	(and now the weatherforecast for tomorrow friday the twentysixth march.)
S2G2T from (Camgoz et al., 2018)	und nun die wettervorhersage fur morgen samstag den siebzehnten april .
	(and now the weatherforecast for tomorrow saturday the seventeenth april.)

CHAPTER V

CONCLUSION

Unlike spoken languages, sign language is multi-channel. It simultaneously exploits multiple components to convey a certain message, in which the dominant hand represents the central and most crucial factor for sign interpretation. To this regard, some existing works in CSLR have considered the effect of fusing handshape features with other non-manual modalities. These approaches usually fail to successfully capture the required handshape dependencies. To address this problem, we have proposed in this study a novel approach that efficiently identifies and combines the relevant non-manual features with the performed handshape for better sign prediction.

Accordingly, our approach is an attention-based network that has the capacity to efficiently discover and learn spatio-temporal information from continuous data. It takes as input a sequence of sign images and consequently generates the corresponding target words. We first proposed to replace traditional recurrent networks with attentional layers for temporal modelling, proving the advantage of the latter for CSLR. Next, we explored the idea of combining hand features with full-body representation through the use of a hand-context module, which leverages attention to aggregate handshapes with their spatio-temporal context. We, furthermore, extended our proposed approach by fusing hand representations with

only their related local context. The experiments empirically showcase the advantage of using such an approach for sign interpretation. We evaluated our networks on the RWTH-PHOENIX-Weather 2014 dataset and we found that our proposed model surpasses most state-of-the-art previous studies.

Secondly, we adopted our proposed model to the task of Sign Language Translation, by adding a decoder module and following an encoder-decoder structure. We also employed a hybrid strategy by training our model on both generating gloss words and spoken language translation. Such an approach allowed for better generalization and helped avoid overfitting. We trained and tested our encoder-decoder models on the RWTH-PHOENIX-Weather 2014T dataset for the task of SLT. We also compared our findings with the baseline results from Camgoz et al. (2018). The experiments showed that the proposed approach yielded high-quality translation results.

In contrast to existing methods on both CSLR and SLT tasks, our approach has several practical improvements. The proposed method leverages the self-attention mechanism instead of the conventional recurrent network for temporal modelling. Attention is considered more resilient for capturing temporal dependencies than their recurrent counterpart. Respectively, this will help better express transitional movements that consist of hand and body motions. Moreover, we found that by leveraging attention to combine hand query features with their respective temporal full-body context tremendously improved sign recognition. In fact, our approach is able to automatically identify the essential sign language components that revolve around the dominant hand and the face areas. We believe that such evidence confirms the efficiency of the proposed approach in both Sign language Recognition and Translation.

For future studies, it will be of interest to investigate the effect of using a forced

alignment algorithm on top of our architecture, similarly to (Koller et al., 2017, 2019). Relying on forced alignment may significantly improves recognition as shown in (Koller et al., 2017) and it is a popular solution to overcome weak supervision by iteratively refining and training on label-to-image prediction. We can also employ HMMs instead of CTC for sequence alignment, as they have been proven to be superior in (Camgoz et al., 2017). Another important venue to explore is to furthermore extend this work by adding a hand detection module that extracts the dominant hand features without the need to provide cropped hand patches. This will also allow us to investigate the effect of combining hand features with their context in our STN (encoder-decoder) models for the task of Sign Language Translation.

BIBLIOGRAPHY

Ahmed, M. A., Bahaa, B., Zaidan, A., Salih, M. & Lakulu, M. M. (2018). A review on systems-based sensory gloves for sign language recognition state of the art between 2007 and 2017. *Sensors*, *18*, 2208. http://dx.doi.org/10.3390/s18072208

Aran, O. (2008). Vision based sign language recognition: modeling and recognizing isolated signs with manual and non-manual components. *Bogazi» ci University*.

Ba, J., Kiros, J. & Hinton, G. (2016). Layer normalization.

Bengio, Y., Simard, P. & Frasconi, P. (1994). Learning long-term dependencies with gradient descent is difficult. *IEEE transactions on neural networks / a publication of the IEEE Neural Networks Council*, *5*, 157–66. http://dx.doi.org/10.1109/72.279181

Brashear, H., Starner, T., Lukowicz, P. & Junker, H. (2005). Using multiple sensors for mobile sign language recognition. pp. 45 – 52. http://dx.doi.org/10.1109/ISWC.2003.1241392

Camgoz, N. C., Hadfield, S., Koller, O. & Bowden, R. (2017). Subunets: End-to-end hand shape and continuous sign language recognition. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pp. 3075–3084. http://dx.doi.org/10.1109/ICCV.2017.332

Camgoz, N. C., Hadfield, S., Koller, O., Ney, H. & Bowden, R. (2018). Neural sign language translation. In *2018 IEEE/CVF Conference on Computer Vision*

and Pattern Recognition, pp. 7784–7793. http://dx.doi.org/10.1109/CVPR.2018.00812

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

Canada, S. (2018). Caractéristiques linguistiques des canadiens. Retrieved on 2018-07-23 from https://www12.statcan.gc.ca/census-recensement/2011/as-sa/98-314-x/98-314-x2011001-fra.cfm

Chen, M., Li, Y., Zhang, Z. & Huang, S. (2018). Tvt: Two-view transformer network for video captioning. In *Asian Conference on Machine Learning*, pp. 847–862.

Crasborn, O. (2006). *Nonmanual structures in sign languages*.

Cui, R., Liu, H. & Zhang, C. (2017). Recurrent convolutional neural networks for continuous sign language recognition by staged optimization. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1610–1618. http://dx.doi.org/10.1109/CVPR.2017.175

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K. & Li, F. F. (2009). Imagenet: a large-scale hierarchical image database. pp. 248–255. http://dx.doi.org/10.1109/CVPR.2009.5206848

Ding, L. & Martinez, A. M. (2007). Recovering the linguistic components of the manual signs in american sign language. In *2007 IEEE Conference on Advanced Video and Signal Based Surveillance*, pp. 447–452. http://dx.doi.org/10.1109/AVSS.2007.4425352

dpwe (2000). What is forced alignment? http://www1.icsi.berkeley.edu/Speech/faq/forcedalign.html.

Foundation des sourds du Quebec (2019). Dictionnaire. https://www.courslsq.net/ewac/lsq/dictionary.php.

Gałka, J., Mąsior, M., Zaborski, M. & Barczewska, K. (2016). Inertial motion sensing glove for sign language gesture acquisition and recognition. *IEEE Sensors Journal*, *16*, 1–1. http://dx.doi.org/10.1109/JSEN.2016.2583542

Georgi, M., Amma, C. & Schultz, T. (2015). Recognizing hand and finger gestures with imu based motion and emg based muscle activity sensing. pp. 99–108. http://dx.doi.org/10.5220/0005276900990108

Girdhar, R., João Carreira, J., Doersch, C. & Zisserman, A. (2019). Video action transformer network. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 244–253. http://dx.doi.org/10.1109/CVPR.2019.00033

Glorot, X. & Bengio, Y. (2010). Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pp. 249–256.

glottolog (2020). Pseudo family: Deaf sign language. Retrieved on 01-03-2020 from https://glottolog.org/resource/languoid/id/deaf1237

Graves, A., Fernández, S., Gomez, F. & Schmidhuber, J. (2006). Connectionist temporal classification: Labelling unsegmented sequence data with recurrent neural 'networks. volume 2006, pp. 369–376. http://dx.doi.org/10.1145/1143844.1143891

Graves, A., Liwicki, M., Fernández, S., Bertolami, R., Bunke, H. & Schmidhuber, J. (2008). A novel connectionist system for unconstrained handwriting recognition. *IEEE transactions on pattern analysis and machine intelligence*, *31*(5), 855–868.

Graves, A., Mohamed, A.-r. & Hinton, G. (2013). Speech recognition with deep recurrent neural networks. *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*, *38*. http://dx.doi.org/10.1109/ICASSP.2013.6638947

Hannun, A. (2017). Sequence modeling with ctc. *Distill*. https://distill.pub/2017/ctc, http://dx.doi.org/10.23915/distill.00008

Huang, J., Zhou, W., Zhang, Q. & Li, H. (2018). Video-based sign language recognition without temporal segmentation.

Jia, Y., Shelhamer, E., Donahue, J., Karayev, S., Long, J., Girshick, R., Guadarrama, S. & Darrell, T. (2014). Caffe: Convolutional architecture for fast feature embedding. *MM 2014 - Proceedings of the 2014 ACM Conference on Multimedia*. http://dx.doi.org/10.1145/2647868.2654889

Karami, A., Zanj, B. & Kianisarkaleh, A. (2011). Persian sign language (psl) recognition using wavelet transform and neural networks. *Expert Syst. Appl.*, *38*, 2661–2667. http://dx.doi.org/10.1016/j.eswa.2010.08.056

Kingma, D. P. & Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Kirti Vashee (April 12, 2019). Understanding mt quality: Bleu scores. https://www.sdl.com/blog/understanding-mt-quality-bleu-scores.html.

Klein, G., Kim, Y., Deng, Y., Senellart, J. & Rush, A. M. (2017). Opennmt: Open-source toolkit for neural machine translation. In *Proc. ACL*. http:

//dx.doi.org/10.18653/v1/P17-4012. Retrieved from https://doi.org/10.18653/v1/P17-4012

Koller, O., Camgoz, N., Ney, H. & Bowden, R. (2019). Weakly supervised learning with multi-stream CNN-LSTM-HMMs to discover sequential parallelism in sign language videos. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *PP*. http://dx.doi.org/10.1109/TPAMI.2019.2911077

Koller, O., Forster, J. & Ney, H. (2015). Continuous sign language recognition: Towards large vocabulary statistical recognition systems handling multiple signers. *Computer Vision and Image Understanding*, *141*, 108–125. http://dx.doi.org/10.1016/j.cviu.2015.09.013

Koller, O., Ney, H. & Bowden, R. (2016). Deep hand: How to train a cnn on 1 million hand images when your data is continuous and weakly labelled. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3793–3802. http://dx.doi.org/10.1109/CVPR.2016.412

Koller, O., Zargaran, S. & Ney, H. (2017). Re-sign: Re-aligned end-to-end sequence modelling with deep recurrent cnn-hmms. http://dx.doi.org/10.1109/CVPR.2017.364

Koller, O., Zargaran, S., Ney, H. & Bowden, R. (2016). Deep sign: Hybrid cnn-hmm for continuous sign language recognition. http://dx.doi.org/10.5244/C.30.136

Kornai, A. (1996). Extended finite state models of language. *Natural Language Engineering*, *2*(4), 287–290.

Kuroda, T., Tabata, Y., Goto, A., Ikuta, H., Murakami, M. et al. (2004). Consumer price data-glove for sign language recognition. In *Proc. of 5th Intl Conf. Disability, Virtual Reality Assoc. Tech., Oxford, UK*, pp. 253–258.

Lang, S., Block, M. & Rojas, R. (2012a). Sign language recognition using kinect. In *International Conference on Artificial Intelligence and Soft Computing*, pp. 394–402. Springer.

Lang, S., Block-Berlitz, M. & Rojas, R. (2012b). Sign language recognition using kinect. pp. 394–402. http://dx.doi.org/10.1007/978-3-642-29347-4_46

Liang, R.-H. & Ouhyoung, M. (1998). A real-time continuous gesture recognition system for sign language. pp. 558 – 567. http://dx.doi.org/10.1109/AFGR.1998.671007

lifeprint (2019a). American sign language: Fingerspelling numbers: Introduction. https://www.lifeprint.com/asl101/fingerspelling/fingerspelling.htm.

lifeprint (2019b). American sign language: Nonmanual markers (nmms). http://www.lifeprint.com/asl101/pages-layout/nonmanualmarkers.htm.

Lin, C.-Y. (2004). ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pp. 74–81., Barcelona, Spain. Association for Computational Linguistics. Retrieved from https://www.aclweb.org/anthology/W04-1013

McGuire, R., Hernandez-Rebollar, J., Starner, T., Henderson, V., Brashear, H. & Ross, D. (2004). Towards a one-way american sign language translator. pp. 620–625. http://dx.doi.org/10.1109/AFGR.2004.1301602

Mehdi, S. A. & Khan, Y. N. (2002). Sign language recognition using sensor gloves. pp. 2204 – 2206 vol.5. http://dx.doi.org/10.1109/ICONIP.2002.1201884

Mitchell, R., Young, T., Bachleda, B. & Karchmer, M. (2006). How many people use asl in the united states? why estimates need updating. *Sign Language Studies*, *6*. http://dx.doi.org/10.1353/sls.2006.0019

Monnier, C., German, S. & Ost, A. (2015). A multi-scale boosted detector for efficient and robust gesture recognition. pp. 491–502. http://dx.doi.org/10.1007/978-3-319-16178-5_34

Müller, R., Kornblith, S. & Hinton, G. E. (2019). When does label smoothing help? In *Advances in Neural Information Processing Systems*, pp. 4696–4705.

Oz, C. & Leu, M. (2011). American sign language word recognition with a sensory glove using artificial neural networks. *Eng. Appl. of AI*, *24*, 1204–1213. http://dx.doi.org/10.1016/j.engappai.2011.06.015

Papineni, K., Roukos, S., Ward, T. & Zhu, W. J. (2002). Bleu: a method for automatic evaluation of machine translation. http://dx.doi.org/10.3115/1073083.1073135

Rommeo79 (2019). Vector language of deaf-mutes hand. american sign language asl alphabet. https://www.dreamstime.com/vector-language-deaf-mutes-hand-american-sign-asl-alphabet-art-image123911624.

Rs, R., Cogswell, M., Das, A., Vedantam, R., Parikh, D. & Batra, D. (2017). Grad-cam: Visual explanations from deep networks via gradient-based localization. pp. 618–626. http://dx.doi.org/10.1109/ICCV.2017.74

Ruder, S. (2017). An overview of multi-task learning in deep neural networks. *arXiv preprint arXiv:1706.05098*.

Sandler, M., Howard, A., Zhu, M., Zhmoginov, A. & Chen, L.-C. (2018). Mobilenetv2: Inverted residuals and linear bottlenecks. pp. 4510–4520. http://dx.doi.org/10.1109/CVPR.2018.00474

Sperber, M., Niehues, J., Neubig, G., Stüker, S. & Waibel, A. (2018). Self-attentional acoustic models. pp. 3723–3727. http://dx.doi.org/10.21437/Interspeech.2018-1910

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I. & Salakhutdinov, R. (2014). Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, *15*(56), 1929–1958. Retrieved from http://jmlr.org/papers/v15/srivastava14a.html

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V. & Rabinovich, A. (2015). Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1–9.

Tran, D., Bourdev, L., Fergus, R., Torresani, L. & Paluri, M. (2015). Learning spatiotemporal features with 3d convolutional networks. pp. 4489–4497. http://dx.doi.org/10.1109/ICCV.2015.510

Valli, C. & Lucas, C. (2000). *Linguistics of American sign language: an introduction*. Gallaudet University Press.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł. & Polosukhin, I. (2017). Attention is all you need. In *Advances in neural information processing systems*, pp. 5998–6008.

Watanabe, S., Hori, T., Kim, S., Hershey, J. & Hayashi, T. (2017). Hybrid ctc/attention architecture for end-to-end speech recognition. *IEEE Journal of Selected Topics in Signal Processing*, *PP*, 1–1. http://dx.doi.org/10.1109/JSTSP.2017.2763455

Weronika Lass (2019). American sign language - more than just hands. https://www.omniglot.com/language/articles/morethanjusthands.htm.

Wu, D., Pigou, L., Kindermans, P., Le, N. D., Shao, L., Dambre, J. & Odobez, J. (2016). Deep dynamic neural networks for multimodal gesture segmentation and

recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *38*(8), 1583–1597. http://dx.doi.org/10.1109/TPAMI.2016.2537340

Wu, J., Tian, Z., Sun, L., Estevez, L. & Jafari, R. (2015). Real-time american sign language recognition using wrist-worn motion and surface emg sensors. pp. 1–6. http://dx.doi.org/10.1109/BSN.2015.7299393

Yang, Z., Yang, D., Dyer, C., He, X., Smola, A. & Hovy, E. (2016). Hierarchical attention networks for document classification. pp. 1480–1489. http://dx.doi.org/10.18653/v1/N16-1174

Yu, A. W., Dohan, D., Luong, M.-T., Zhao, R., Chen, K., Norouzi, M. & Le, Q. V. (2018). Qanet: Combining local convolution with global self-attention for reading comprehension. *arXiv preprint arXiv:1804.09541*.

Zafrulla, Z., Brashear, H., Starner, T., Hamilton, H. & Presti, P. (2011). American sign language recognition with the kinect. pp. 279–286. http://dx.doi.org/10.1145/2070481.2070532

Zhang, X., Xiang, C., Li, Y., Lantz, V., Wang, K. & Yang, J. (2011). A framework for hand gesture recognition based on accelerometer and emg sensors. *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on*, *41*, 1064 – 1076. http://dx.doi.org/10.1109/TSMCA.2011.2116004

Zhang, X., Xiang, C., Wang, W.-h., Yang, J., Lantz, V. & Wang, K. (2009). Hand gesture recognition and virtual game control based on 3d accelerometer and emg sensors. pp. 401–406. http://dx.doi.org/10.1145/1502650.1502708