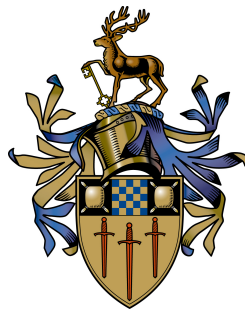


# Neural Sign Language Recognition and Translation

Necati Cihan Camgöz

Submitted for the Degree of  
Doctor of Philosophy  
from the  
University of Surrey



Centre for Vision, Speech and Signal Processing  
Faculty of Engineering and Physical Sciences  
University of Surrey  
Guildford, Surrey GU2 7XH, U.K.

October 2020

© Necati Cihan Camgöz 2020



# Abstract

Sign languages have been studied by computer vision researchers for the last three decades. One of the end goals of vision-based sign language research is to build systems that can understand and translate sign languages to spoken/written languages or vice versa, to create a more natural medium of communication between the hearing and the Deaf. However, most research to date has mainly focused on isolated sign recognition and spotting, neglecting the underlying rich grammatical and linguistic structures of sign language that differ from spoken language. More recently, Continuous Sign Language Recognition (CSLR) has become feasible with the availability of large benchmark datasets, such as the RWTH-PHOENIX-Weather-2014 Dataset (PHOENIX14), and the development of algorithms that can learn from weak annotations. Although, CSLR is able to recognize sign gloss sequences, further progress is required to produce meaningful spoken/written language interpretations of continuous sign language videos.

In this thesis, we introduce the Sign Language Translation (SLT) problem and lay groundwork for future research on this topic. The objective of SLT is to generate spoken/written language translations from continuous sign language videos, taking into account the different word orders and grammar. We evaluate our approaches on the RWTH-PHOENIX-Weather-2014T (PHOENIX14T) dataset, the first and the currently only publicly available Continuous SLT dataset aimed at vision based sign language research. It provides spoken language translations and gloss level annotations for German Sign Language videos of weather broadcasts. We lay down several evaluation protocols to underpin future research in this newly established field.

In the first contribution chapter of this thesis, we formalize SLT in the framework of Neural Machine Translation (NMT) and propose the first SLT approach, *Neural Sign Language Translation*. We combine Convolutional Neural Networks (CNNs) and attention-based encoder-decoder models, which allows us to jointly learn the spatial representations, the underlying language model, and the mapping between sign and spoken language. We investigate different configurations of the proposed network with both end-to-end and pretrained settings (using expert gloss annotations). In our experiments, recognizing glosses and then translating them to spoken languages (Sign2Gloss2Text) drastically outperforms an end-to-end direct translation approach (Sign2Text). Sign2Gloss2Text utilizes a state-of-the-art CSLR model to predict gloss sequences from sign language videos and then solves SLT as text-to-text translation problem. This suggests that using gloss level intermediate representations, essentially dividing the process into two stages, is necessary to train accurate SLT models.

Glosses are incomplete text-based representations of continuous multi-channel visual signals, that are sign languages. Thus, the best performing two step configuration of *Neural Sign Language Translation* has an inherent information bottleneck limiting translation. To address this issue, in the second contribution chapter of this thesis we formulate SLT as a multi-task learning problem. We introduce a novel transformer based architecture, *Sign Language Transformers*, that jointly learn CSLR and SLT while being trainable in an end-to-end manner. This is achieved by using a Connectionist Temporal Classification (CTC) loss to bind the recognition and translation problems into a single unified architecture. This joint approach does not require any ground-truth

timing information, simultaneously solving two co-dependant sequence-to-sequence learning problems and leads to significant performance gains. We report state-of-the-art CSLR and SLT results achieved by our Sign Language Transformers. Our translation networks outperform both sign video to spoken language and gloss to spoken language translation models, in some cases more than doubling the performance of *Neural Sign Language Translation* (Sign2Text configuration - 9.58 vs. 21.80 BLEU-4 Score).

Models we introduce in both first and second contribution chapters heavily rely on gloss information, either in the form of direct supervision or for pretraining. To realize large scale sign language translation, that is on par with their spoken/written language counterparts, we require more parallel datasets. However, annotating sign glosses is a laborious task and acquiring such annotations for large datasets is infeasible. To address this issue, in our last contribution chapter we propose modelling SLT based on sign articulators instead of glosses. Contrary to previous research, which mainly focused on manual features, we incorporate both manual and non-manual features of the sign. We utilize hand shape, mouthings and upper body pose representations to model sign in a holistic manner.

We propose a novel transformer based architecture, called *Multi-Channel Transformers*, aimed at sequence-to-sequence learning problems where the source information is embedded over several channels. This approach allows the networks to model both the inter and the intra relationship between asynchronous source channels. We also introduce a channel anchoring loss to help our models preserve channel specific information while also regulating training against overfitting.

We apply multi-channel transformers to the task of SLT and realize the first multi-articulatory translation approach. Our experiments on PHOENIX14T demonstrate that our approach achieves on par or better translation performance against several baselines, overcoming the reliance on gloss information which underpin previous approaches. Now we have broken the dependency upon gloss information, future work will be to scale learning to larger datasets, such as broadcast footage, where gloss information is not available.

**Key words:** Sign Language Recognition, Sign Language Translation, Multi-Articulatory Sign Language Translation, Neural Machine Translation, Deep Learning, Sequence-to-sequence Learning

Email: [n.camgoz@surrey.ac.uk](mailto:n.camgoz@surrey.ac.uk)

WWW: [www.cihancamgoz.com](http://www.cihancamgoz.com)

## Acknowledgements

I would first like to thank my advisors, Prof. Richard Bowden and Dr. Simon Hadfield. You two have patience of saints. Thank you for being excellent mentors as well as great friends. I would also like thank you for teaching me how to take the reins of my own research and allowing me to pursue my ideas. I will always be grateful for everything you have done for me. I will try to repay my debt by teaching the skills and the lessons I learned from you two to the next generations.

I would also like to thank Prof. Lale Akarun, who introduced me to the field of computer vision and sign language research. Her recommendation was the main reason I applied to this PhD position and it was one of the best decisions of my life. I would also like of thank all of the member of the Computer Engineering Department of Bogazici University and more specifically the Perceptual Intelligence Laboratory, which was literally my home for a better part of two years.

Next, I would like to thank Oscar Koller. I think the quote “If I have seen further it is by standing on the shoulders of Giants.” best describes our relationship. If I have realized SLT it is by using the dataset your team collected. Thank you for all of our discussions, your critical comments and your support. It is a joy to collaborate with you and I can't wait to see what we will do next.

I would also like to thank member of the Oculus Hand Tracking team. Thank you Rob Wang for giving me the chance to be a part of your team. I would also like to thank Lingling Tao, Yuting Ye, Chris Twigg for being amazing mentors. You shown me how research can be applied to build amazing products that revolutionize the world.

I would also like to thank all of the past and present members of the CogVis Lab and CVSSP. It has been an amazing 4 years. I would also like to thank my collaborators, including but not limited to members of the SMILE, Content4ALL and ExTOL Projects.

Last but not least, I would like to thank Nimet Kaygusuz for always being there for me and keeping my sanity intact. I wouldn't be the person that I am today without you. In all honesty, I probably wouldn't be here at all.

This PhD was funded by the SNSF Sinergia project “Scalable Multimodal Sign Language Technology for Sign Language Learning and Assessment (SMILE)” grant agreement number CRSII2.160811 and the University of Surrey - Overseas Research Scholarship. I would also like to thank NVIDIA Corporation for their GPU grants.



# Contents

<b>Nomenclature</b>	<b>xi</b>
<b>Symbols</b>	<b>xiii</b>
<b>Declaration</b>	<b>xxi</b>
<b>1 Introduction</b>	<b>1</b>
<b>2 Computer Vision based Sign Language Research</b>	<b>11</b>
2.1 Sign Language Recognition . . . . .	11
2.2 Sign Language Translation . . . . .	14
<b>3 Modern Deep Learning Approaches and Their Applications</b>	<b>17</b>
3.1 Convolutional Neural Networks . . . . .	17
3.2 Sequence-to-Sequence Learning . . . . .	19
3.3 Neural Machine Translation . . . . .	20
<b>4 Sign Language Translation Dataset and Evaluation Protocols</b>	<b>27</b>
4.1 Evaluation Protocols . . . . .	28
4.2 Performance Metrics . . . . .	30
4.2.1 Word Error Rate (WER) . . . . .	30
4.2.2 Bilingual Evaluation Understudy (BLEU) . . . . .	30
4.2.3 Recall-Oriented Understudy for Gisting Evaluation (ROUGE) . . . . .	31

---

<b>5</b>	<b>Neural Sign Language Translation</b>	<b>33</b>
5.1	Methodology . . . . .	34
5.1.1	Spatial and Word Embeddings . . . . .	34
5.1.2	Tokenization Layer . . . . .	36
5.1.3	Attention-based Recurrent Encoder-Decoder Networks . . . . .	36
5.2	Quantitative Experiments . . . . .	43
5.2.1	Gloss2Text: Simulating Perfect Recognition . . . . .	44
5.2.2	Sign2Text: From Sign Video To Spoken Text . . . . .	49
5.2.3	Sign2Gloss2Text: Gloss Level Supervision . . . . .	49
5.3	Qualitative Examples . . . . .	50
5.4	Closing Remarks . . . . .	51
<b>6</b>	<b>Sign Language Transformers</b>	<b>55</b>
6.1	Methodology . . . . .	57
6.1.1	Spatial and Word Embeddings . . . . .	59
6.1.2	Sign Language Recognition Transformers . . . . .	60
6.1.3	Sign Language Translation Transformers . . . . .	61
6.2	Quantitative Experiments . . . . .	63
6.2.1	Implementation and Evaluation Details . . . . .	63
6.2.2	Text-to-Text Sign Language Translation . . . . .	64
6.2.3	<i>Sign2Gloss</i> . . . . .	65
6.2.4	<i>Sign2Text</i> and <i>Sign2(Gloss+Text)</i> . . . . .	67
6.3	Qualitative Examples . . . . .	69
6.4	Closing Remarks . . . . .	69
<b>7</b>	<b>Multi-Channel Transformers</b>	<b>73</b>
7.1	Methodology . . . . .	74
7.1.1	Channel and Word Embeddings . . . . .	76
7.1.2	Multi-Channel Encoder Layer . . . . .	76
7.1.3	Multi-Channel Decoder Layer . . . . .	79
7.1.4	Loss Functions . . . . .	81
7.2	Implementation and Evaluation Details . . . . .	82



---

7.3	Quantitative Experiments . . . . .	84
7.3.1	Channel Feature and Word Embeddings . . . . .	85
7.3.2	Single Channel Baselines . . . . .	87
7.3.3	Early and Late Fusion of Sign Channels . . . . .	88
7.3.4	Multi-Channel Transformers . . . . .	88
7.4	Qualitative Examples . . . . .	91
7.5	Closing Remarks . . . . .	95
<b>8</b>	<b>Conclusions and Future Work</b>	<b>99</b>
8.1	Possible Future Work . . . . .	100
	<b>Bibliography</b>	<b>103</b>



# Nomenclature

<b>ASL</b>	American Sign Language
<b>BSL</b>	British Sign Language
<b>BSLCP</b>	British Sign Language Corpus Project
<b>BLEU</b>	Bilingual Evaluation Understudy
<b>BLSTM</b>	Bidirectional Long Short-Term Memory
<b>CNN</b>	Convolutional Neural Network
<b>CRF</b>	Conditional Random Field
<b>CSLR</b>	Continuous Sign Language Recognition
<b>CTC</b>	Connectionist Temporal Classification
<b>DL</b>	Deep Learning
<b>DGS</b>	German Sign Language - Deutsche Gebärdensprache
<b>DSGS</b>	Swiss German Sign Language - Deutschschweizer Gebärdensprache
<b>FC</b>	Fully Connected
<b>GNMT</b>	Google's Neural Machine Translation
<b>GAN</b>	Generative Adversarial Network
<b>GPU</b>	Graphics Processing Unit
<b>GRU</b>	Gated Recurrent Unit
<b>HMM</b>	Hidden Markov Model
<b>HOG</b>	Histograms of Oriented Gradients
<b>IP</b>	Inner Product
<b>ISL</b>	Irish Sign Language
<b>LCS</b>	Longest Common Subsequence

---

<b>LSTM</b>	Long Short-Term Memory
<b>NMT</b>	Neural Machine Translation
<b>OCR</b>	Optical Character Recognition
<b>PHOENIX12</b>	RWTH-PHOENIX-Weather-2012 Dataset
<b>PHOENIX14</b>	RWTH-PHOENIX-Weather-2014 Dataset
<b>PHOENIX14T</b>	RWTH-PHOENIX-Weather-2014T Dataset
<b>RELU</b>	Rectified Linear Units
<b>RNN</b>	Recurrent Neural Network
<b>ROUGE</b>	Recall-Oriented Understudy for Gisting Evaluation
<b>SGD</b>	Stochastic Gradient Descent
<b>SLA</b>	Sign Language Assessment
<b>SLR</b>	Sign Language Recognition
<b>SLT</b>	Sign Language Translation
<b>SIFT</b>	Scale Invariant Feature Transform
<b>SURF</b>	Speeded Up Robust Features
<b>SMPL</b>	Skinned Multi-Person Linear Model
<b>MANO</b>	hand Model with Articulated and Non-rigid defOrmations
<b>BERT</b>	Bidirectional Encoder Representations from Transformers
<b>WER</b>	Word Error Rate
<b>SLRT</b>	Sign Language Recognition Transformer
<b>SLTT</b>	Sign Language Translation Transformer

# Symbols

## Introduced in Chapter 3

$\mathcal{X}$	Source sequence in a sequence-to-sequence learning task.
$x_t$	$t^{\text{th}}$ token of a single channel source sequence $\mathcal{X}$
$T$	Cardinality of the source sequence
$\mathcal{Y}$	Target sequence in a sequence-to-sequence learning task.
$y_u$	$u^{\text{th}}$ token of a target sequence $\mathcal{Y}$
$U$	Cardinality of a target sequence
$Q$	Query matrix used in dot-product attention function
$K$	Key matrix used in dot-product attention function
$K^T$	Transpose of the key matrix $K$
$V$	Value matrix used in dot-product attention function
$d_m$	Number of hidden units of a transformer model

## Introduced in Chapter 4

$ \cdot $	Cardinality operation
$\mathcal{G}^*$	Ground truth sign gloss sequence
$\mathcal{G}^d$	A set of deleted glosses in predicted gloss sequence
$\mathcal{G}^i$	A set of inserted glosses in predicted gloss sequence
$\mathcal{G}^s$	A set of substituted glosses in predicted gloss sequence
$\tilde{\mathcal{S}}$	Predicted spoken language sentence
$\mathcal{S}$	Ground truth spoken language sentence
$\tilde{\mathcal{S}}^n$	A set of n-grams extracted from $\tilde{\mathcal{S}}$

---

$\mathcal{S}^n$	A set of n-grams extracted from $\mathcal{S}$
$p_n$	$n$ -gram precision of a predicted sentence $\tilde{\mathcal{S}}$ given the reference sentence $\mathcal{S}$
BP	Brevity penalty used for calculating BLEU scores
$p_{lcs}$	Longest Common Subsequence based precision measure used for calculating ROUGE-L score
$r_{lcs}$	Longest Common Subsequence based recall measure used for calculating ROUGE-L score

### Introduced in Chapter 5

$\mathcal{V}$	An ordered series of frames belonging to a video
$I_t$	A video frame, <i>i.e.</i> an image, at time step $t$
$w_u$	$u^{th}$ word in a sentence
$f_t$	Spatial representation of a video frame at time step $t$ ( $I_t$ )
$m_u$	Word embedding of the $u^{th}$ word in a sentence ( $w_u$ )
$f_{1:T}$	Spatial representation of a video ( $\mathcal{V}$ ) with $T$ number of frames
$z_{1:J}$	Tokenized representation of spatial features of a sign video ( $f_{1:T}$ )
$\mathcal{B}$	A mapping function from tokenized representation of a video ( $z_{1:J}$ ) to the target sentence ( $\mathcal{S}$ )
$o_j^e$	Encoder output given the previous hidden state $h_{j+1}^e$ and input $z_j$
$h_j^e$	Encoder's hidden state after observing inputs $z_{J:j}$
$h_{sign}^e$	Encoder's hidden state after observing all the inputs $z_{J:1}$
$o_u^d$	Decoder output given the previous hidden state $o_{u-1}^d$ and previously predicted word's embedding $m_{u-1}$
$h_u^d$	Decoder's hidden state after observing the word embeddings $m_{1:u-1}$
< bos >	Special token indicating the beginning of a sentence
< eos >	Special token indicating the end of a sentence
$c_u$	Context vector that is used for predicting $w_u$ , which is a weighted sum of the encoder outputs $o_{j:1}^e$
$\gamma_j^u$	Attention weights given the encoder output $o_j^e$ and the decoder output $o_u^d$
$\tilde{o}_u^d$	Decoder output after it is combined with context vector $c_u$
$G_u^f$	Forget Gate of an LSTM unit

---

$G_u^i$	Input Gate of an LSTM unit
$G_u^o$	Output Gate of an LSTM unit
$C_u$	Cell State of an LSTM unit
$\tilde{C}_u$	Candidate Cell State of an LSTM Unit
$G_u^r$	Update Gate of a GRU
$G_u^h$	Output Gate of a GRU
$\sigma$	Sigmoid activation function
$\tanh$	Tanh activation function
$\cdot^*$	Element-wise multiplication

### Introduced in Chapter 6

$\mathcal{G}$	An ordered series of sign glosses
$N$	Cardinality of a sign gloss sequence $\mathcal{G}$
$\hat{f}_t$	Positionally encoded frame representation $f_t$
$\hat{m}_u$	Positionally encoded word embedding $m_u$
$z_{1:T}$	Spatio-temporal representations of video frames $I_{1:T}$
$\mathcal{L}_R$	Recognition loss value
$\lambda_R$	Recognition loss scalar weight
$\mathcal{L}_T$	Translation loss value
$\lambda_T$	Translation loss scalar weight
$\mathcal{L}$	Total loss value calculated by taking a weighted sum of $\mathcal{L}_R$ and $\mathcal{L}_T$

### Introduced in Chapter 7

$X_i$	$i^{th}$ source channel sequence with a cardinality of $T_i$
$x_{i,t}$	$t^{th}$ token of the $i^{th}$ source channel sequence $X_i$
$\hat{x}_{i,t}$	Positionally encoded linear projection of target token $x_{i,t}$
$y_u$	$u^{th}$ token of the target sequence
$\hat{y}_u$	Positionally encoded linear projection of target token $y_u$
$W_i^{\alpha,\beta}$	Weight matrix in $\alpha$ module used for extracting $\beta$ values from $i^{th}$ source channel

---

$b_i^{\alpha,\beta}$	Bias term in $\alpha$ module used for extracting $\beta$ values from $i^{th}$ source channel
$Q_i^\alpha$	Scaled dot-product attention query matrix in $\alpha$ module extracted from $i^{th}$ source channel
$K_i^\alpha$	Scaled dot-product attention key matrix in $\alpha$ module extracted from $i^{th}$ source channel
$V_i^\alpha$	Scaled dot-product attention value matrix in $\alpha$ module extracted from $i^{th}$ source channel
$\mathcal{L}_{A,i}$	Anchoring loss value for $i^{th}$ source channel
$\lambda_{A,i}$	Anchoring loss weight for $i^{th}$ source channel



# List of Figures

1.1	A sample sequence from RWTH-PHOENIX-Weather-2014T Dataset (PHOENIX14T) which depicts the differences between Continuous Sign Language Recognition and Sign Language Translation. (Note that the free form German translations and the German Sign Language - Deutsche Gebärdensprache (DGS) glosses are provided by the dataset and we added their English translations.) . . . . .	3
1.2	An overview of the Neural Sign Language Translation [24] approach.	6
1.3	An overview of the Sign Language Transformers [27] approach, jointly trained to perform Continuous Sign Language Recognition (CSLR) and Sign Language Translation (SLT). . . . .	7
1.4	An overview of the proposed Multi-channel Transformer [26] architecture applied to the multi-articulatory SLT problem. . . . .	9
3.1	A generic Recurrent Neural Network (RNN)-based encoder-decoder architecture for machine translation. . . . .	21
3.2	Dot product attention-based encoder-decoder architecture proposed by Luong <i>et al.</i> [113] . . . . .	22
3.3	Transformer Networks proposed by Vaswani <i>et al.</i> [187] . . . . .	23
5.1	An overview of our Sign Language Translation approach that generates spoken language translations of sign language videos. . . . .	35
5.2	Visualizations of Long Short-Term Memory [82] unit and Gated Recurrent Unit [38]. . . . .	38
5.3	Visualizations of the attention mechanisms used in this chapter. . . . .	41
5.4	Effects of Beam Width on Gloss2Text performance. . . . .	48
6.1	A detailed overview of a single layered Sign Language Transformer. (SE: Spatial Embedding, WE: Word Embedding , PE: Positional Encoding, FF: Feed Forward) . . . . .	58
7.1	An overview of the proposed Multi-channel Transformer [26] approach.	75
7.2	A detailed overview of the proposed multi-channel attention modules.	77



# List of Tables

4.1	Key statistics of the RWTH-PHOENIX-Weather-2014T Dataset. . . .	28
4.2	Example $n$ -gram extractions with the lengths of one, two and three. . .	31
5.1	Gloss2Text: Effects of using different recurrent units on translation performance. . . . .	45
5.2	Gloss2Text: Attention Mechanism Experiments. . . . .	46
5.3	Gloss2Text: Batch Size Experiments. . . . .	47
5.4	Sign2Text: Attention Mechanism Experiments. . . . .	49
5.5	Effects of different tokenization schemes for sign to text translation. .	50
5.6	Samples where models failed to correctly translate the target sentence.	53
6.1	(Top) New baseline results for text-to-text tasks on Phoenix2014T [24] using transformer networks and (Bottom) Our best performing Sign Language Transformers compared against the state-of-the-art. . . . .	66
6.2	Impact of the Spatial Embedding Layer variants. . . . .	67
6.3	Impact of different numbers of layers . . . . .	67
6.4	Training Sign Language Transformers to jointly learn recognition and translation with different weight on recognition loss. . . . .	68
6.5	Generated spoken language translations by our models - I. . . . .	70
6.6	Generated spoken language translations by our models - II. . . . .	71
7.1	Effects of using different embedding setups on hand channel features to spoken language translation performance. . . . .	86
7.2	Single channel Sign Language Translation (SLT) baselines using different network architectures, . . . . .	89
7.3	SLT performance of early and late channel fusion approaches. . . . .	90
7.4	Multi-channel Transformer based multi-articulatory SLT results. . . .	92

7.5	Spoken language translations produced by our best Multi-Channel Transformer model - Good Translation Examples. . . . .	93
7.6	Spoken language translations produced by our best Multi-Channel Transformer model - Mediocre and Poor Translation Examples. . . . .	94
7.7	Spoken language translations produced by RNN-based (Chapter 5) and transformer-based (Chapter 6 and Chapter 7) approaches presented in this thesis. . . . .	96

# Declaration

Some elements of this work have appeared or will appear in the following publications. Contributions of this thesis and how each publication relates to its chapters is discussed in Chapter 1. The publications where I was the first author and that are key publications supporting this thesis:

Section 5 Necati Cihan Camgoz, Simon Hadfield, Oscar Koller, Hermann Ney, and Richard Bowden. Neural Sign Language Translation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

Section 6 Necati Cihan Camgoz, Oscar Koller, Simon Hadfield, and Richard Bowden. Sign Language Transformers: Joint End-to-end Sign Language Recognition and Translation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020

Section 7 Necati Cihan Camgoz, Oscar Koller, Simon Hadfield, and Richard Bowden. Multi-channel Transformers for Multi-articulatory Sign Language Translation. In *16th European Conference on Computer Vision Workshops (ECCVW)*, 2020

The publications where I was one of the primary authors and are relevant to this thesis but are excluded because they “did not fit the story”:

- Sarah Ebling, Volker Hegelheimer, Necati Cihan Camgoz, and Richard Bowden. Use of New Technologies in L2 Assessment. In *Handbook of Language Assessment across Modalities*. Oxford University Press, 2020
- Kearsy Cormier, Neil Fox, Bencie Woll, Andrew Zisserman, Necati Cihan Camgoz, and Richard Bowden. ExTOL: Automatic recognition of British Sign Language using the BSL Corpus. In *Proceedings of the Sign Language Translation and Avatar Technology (SLTAT)*, 2019
- Sarah Ebling, Necati Cihan Camgoz, Penny Boyes Braem, Katja Tissi, Sandra Sidler-Miserez, Stephanie Stoll, Simon Hadfield, Tobias Haug, Richard Bowden, Sandrine Tornay, Marzieh Razavi, and Mathew Magimai-Doss. SMILE Swiss German Sign Language Dataset. In *Proceedings of the International Conference on Language Resources and Evaluation (LREC)*, 2018

- 
- Necati Cihan Camgoz, Simon Hadfield, Oscar Koller, and Richard Bowden. SubUNets: End-to-end Hand Shape and Continuous Sign Language Recognition. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2017.
  - Necati Cihan Camgoz, Simon Hadfield, and Richard Bowden. Particle Filter based Probabilistic Forced Alignment for Continuous Gesture Recognition. In *Proceedings of the IEEE International Conference on Computer Vision Workshops (ICCVW)*, 2017
  - Necati Cihan Camgoz, Simon Hadfield, Oscar Koller, and Richard Bowden. Using Convolutional 3D Neural Networks for User-Independent Continuous Gesture Recognition. In *Proceedings of the IEEE International Conference on Pattern Recognition Workshops (ICPRW)*, 2016

Other Sign Language Recognition, Translation and Production publications I have contributed to during my PhD but are not affiliated with this thesis:

- Benjamin Saunders, Necati Cihan Camgoz, and Richard Bowden. Adversarial Training for Multi-Channel Sign Language Production. In *Proceedings of the British Machine Vision Conference (BMVC)*, 2020
- Benjamin Saunders, Necati Cihan Camgoz, and Richard Bowden. Progressive Transformers for End-to-End Sign Language Production. In *16th European Conference on Computer Vision (ECCV)*, 2020
- Ogulcan Ozdemir, Ahmet Alp Kindiroglu, Necati Cihan Camgoz, and Lale Akarun. BosphorusSign22k Sign Language Recognition Dataset. In *9th Workshop on the Representation and Processing of Sign Languages*, 2020
- Stephanie Stoll, Necati Cihan Camgoz, Simon Hadfield, and Richard Bowden. Text2Sign: Towards Sign Language Production using Neural Machine Translation and Generative Adversarial Networks. *International Journal of Computer Vision (IJCV)*, 2020
- Oscar Koller, Necati Cihan Camgoz, Richard Bowden, and Hermann Ney. 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 (TPAMI)*, 2019
- Sandrine Tornay, Marzieh Razavi, Necati Cihan Camgoz, Richard Bowden, and Mathew Magimai-Doss. HMM-based Approaches to Model Multichannel Information in Sign Language Inspired from Articulatory Features-based Speech Processing. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2019
- Stephanie Stoll, Necati Cihan Camgoz, Simon Hadfield, and Richard Bowden. Sign Language Production using Neural Machine Translation and Generative Adversarial Networks. In *Proceedings of the British Machine Vision Conference (BMVC)*, 2018

- 
- Necati Cihan Camgoz, Ahmet Alp Kindiroglu, and Lale Akarun. Sign Language Recognition for Assisting the Deaf in Hospitals. In *Proceedings of the International Workshop on Human Behavior Understanding (HBU)*, 2016

Other work I have contributed to during my PhD but are not directly relevant to sign language research and are not affiliated with this thesis:

- Matthew Vowels, Necati Cihan Camgoz, and Richard Bowden. NestedVAE: Isolating Common Factors via Weak Supervision. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020
- Matthew Vowels, Necati Cihan Camgoz, and Richard Bowden. Gated Variational AutoEncoders: Incorporating Weak Supervision to Encourage Disentanglement. In *Proceedings of the IEEE International Conference on Automatic Face and Gesture Recognition (FG)*, 2020
- Terry Windeatt, Cemre Zor, and Necati Cihan Camgoz. Approximation of Ensemble Boundary using Spectral Coefficients. *IEEE Transactions on Neural Networks and Learning Systems (TNNLS)*, 30(4), 2019
- M. Amac Guvensan, A. Oguz Kansiz, Necati Cihan Camgoz, H. Irem Turkmen, A. Gokhan Yavuz, and M. Elif Karsligil. An Energy-Efficient Multi-Tier Architecture for Fall Detection on Smartphones. *Sensors*, 17(7), 2017





# Chapter 1

## Introduction

Sign Languages are the natural medium of communication of the Deaf. Despite the common belief, sign languages are not transcriptions of spoken languages but languages in their own right. Like any spoken language, they form organically in a region and evolve through time.

Each sign language has its own linguistic rules and grammatical structures as well as having a unique vocabulary that does not have a one-to-one correspondence to spoken language. Unlike their spoken counter parts, which make use of the sound patterns, sign languages employ multiple complementary channels to convey information [185]. These channels, also known as *articulators* in linguistics [116], can be grouped under two main categories with respect to their role in conveying information: *manual* and *non-manual* features [17, 18].

*Manual Features* are the hand shapes and hand movements used for performing a sign while *Non-Manual Features* are the actions that are produced by other body parts, such as head movements, upper body posture, eye gaze, facial expressions and mouthings<sup>1</sup>. Although manual features can be considered as the dominant part of the sign morphology, they alone do not encapsulate the full context of the conveyed information. To give clarity, emphasis and additional meaning, signers use non-manual features, which are rich multi-modal linguistic tools that undertake numerous roles within the syntax of sign languages.

---

<sup>1</sup>Mouthings are lip patterns that accompany a sign.

The multi-channel nature of sign languages make sign language research a challenging and intriguing field for both linguists and computer scientists. Linguistic research concentrates on understanding the nature of sign languages. Sign language linguists collect extensive corpora and analyze them in-depth to derive the underlying rules [153, 78]. While linguists focus on explaining the workings of sign languages, computer scientists aspire to develop systems that are capable of recognizing, understanding and producing sign languages.

One of the main goals of computational sign language research is to build systems that can translate from sign languages to spoken languages or vice versa, thus creating a more natural medium of communication for the Deaf to converse with the hearing. Computer science researchers, mainly from the computer vision community, have been working towards developing such applications for the last three decades [164, 165, 43]. There have been several important advancements, such as the availability of large scale datasets [65, 33] and the shifting of research focus towards recognising the sequence of sign glosses<sup>2</sup> (Continuous Sign Language Recognition (CSLR)) [97] from isolated sign language recognition [8] and spotting [41, 19]. However, the move from recognition to translation is still in its infancy [24].

The distinction between CSLR and Sign Language Translation (SLT) is important as the grammar of sign and spoken languages are very different. As can be seen in Figure 1.1, these differences include (to name a few): different word ordering, multiple channels used to convey concurrent information and the use of direction and space to convey the relationships between objects. Put simply, the mapping between spoken languages and sign is complex and there is no simple word-to-sign mapping. Thus, further progress is required to realize systems that would be able to recognize and understand continuous sign language streams and then translate them into spoken languages in real-time in a user-independent manner.

The motivation behind this thesis is to lay the ground work for this future sign language translation research. We start by surveying the computer vision based sign language recognition and translation literature in *Chapter 2*. Guided by the previous research, we

---

<sup>2</sup>Sign glosses are spoken language words that match the meaning of signs and, linguistically, manifest as minimal lexical items.

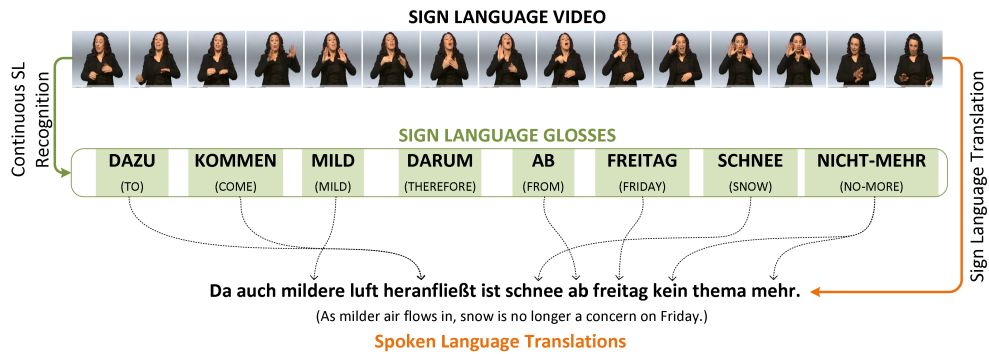


Figure 1.1: A sample sequence from PHOENIX14T which depicts the differences between Continuous Sign Language Recognition and Sign Language Translation. (Note that the free form German translations and the DGS glosses are provided by the dataset and we added their English translations.)

hypothesise that such translation systems will require completion of several sub-tasks, which are currently unsolved:

**Sign Segmentation:** Firstly, the system needs to detect sign sentences, which are commonly formed using topic-comment structures [173], from continuous sign language videos. This is trivial to achieve for text based machine translation tasks [123], where the models can use punctuation marks to separate sentences. Speech-based recognition and translation systems, on the other hand, look for pauses, *e.g.* silent regions, between phonemes to segment spoken language utterances [186, 205]. There have been studies in the literature addressing automatic sign segmentation [93, 150, 155, 15, 35]. However, to the best of our knowledge, there is no study which utilizes sign segmentation for realizing continuous SLT.

**Sign Language Recognition and Understanding:** Following successful segmentation, the system needs to understand what information is being conveyed within a sign sentence. Current approaches tackle this by recognizing sign glosses and other linguistic components. Such methods can be grouped under the banner of CSLR [97, 23]. From a computer vision perspective, this is the most challenging task. Considering the input of the system is high dimensional spatio-temporal data, *i.e.* sign videos, models are required that understand what a signer looks like and how they interact and move within their 3D signing space. Moreover, the model needs to comprehend what these

aspects mean in combination. This complex modelling problem is exacerbated by the asynchronous multi-articulatory nature of sign languages [149, 168]. Although there have been promising results towards CSLR, the state-of-the-art [96] can only recognize sign glosses and operate within a limited domain of discourse, namely weather forecasts [65].

**Sign Language Translation:** Once the information embedded in the sign sentence is understood by the system, the final step is to generate spoken language sentences. As with any natural language, sign languages have their own unique linguistic and grammatical structures, which often do not have a one-to-one mapping to their spoken language counterparts. As such, this problem truly represents a machine translation task. Initial studies conducted by computational linguists have used text-to-text statistical machine translation models to learn the mapping between sign glosses and their spoken language translations [120]. However, glosses are simplified representations of sign languages and linguists are yet to come to a consensus on how sign languages should be annotated.

In this thesis we focus on the latter two sub-tasks, namely continuous sign language recognition and translation. These tasks are sequence-to-sequence learning problems by their nature. Additionally, SLT is a special case of machine translation, where the input domain is sign videos instead of spoken language in text form. Both sequence-to-sequence learning and machine translation fields have seen significant improvements over the last decade with the introduction of modern Deep Learning (DL) approaches [70]. Hence, in *Chapter 3*, we give a brief background on the state-of-the-art in DL and its application to computer vision, sequence-to-sequence learning and machine translation fields.

Inspired by the recent developments in the aforementioned fields, we propose combining Convolutional Neural Networks (CNNs) [107] with sequence-to-sequence learning methods [72, 123] to realize end-to-end sign language recognition and translation. However, to be able to develop such methods we first needed a dataset with continuous sign language videos and their spoken language translations.

Although there are vast quantities of sign language interpretations broadcast everyday,

---

they lack the alignment between sign sentences and their corresponding translations (This relates back to the Sign Segmentation task which is not investigated in the scope of this thesis). There are also linguistic corpora with sign gloss annotations and spoken language translations. However, these are either sparsely annotated [153] or the number of annotated samples are too few with respect to corpora’s domain of discourse [153, 78]. To address this, we present the first publicly available sign language translation dataset, the RWTH-PHOENIX-Weather-2014T Dataset (PHOENIX14T). Curated by our collaborates in RWTH-Aachen, it is an extended version of the popular RWTH-PHOENIX-Weather-2014 Dataset (PHOENIX14). Aimed at computer vision researchers, PHOENIX14T is composed of parallel sign videos, gloss annotations (in DGS) and their spoken language translations (in German). In *Chapter 4*, we give further statistics of the dataset, share the performance metrics used, namely Bilingual Evaluation Understudy (BLEU) [132] and Recall-Oriented Understudy for Gisting Evaluation (ROUGE) [110] scores, and introduce the evaluation protocols developed to underpin future SLT research.

Using PHOENIX14T, in *Chapter 5*, we investigate the feasibility of training DL based methods to generate spoken language sentences from sign language videos. We utilize attention-based encoder-decoder architectures [113, 11], which were state-of-the-art in the field of Neural Machine Translation (NMT) at the time of the research, and combine them with CNNs to realize the first continuous sign video to spoken language translation approach, *Neural Sign Language Translation*. Although, the trained model is able to generate meaningful spoken language translations, its translation performance is drastically lower than other NMT baselines (9.58 BLEU-4). We believe this is due to the fact that going from sign language videos to spoken language sentences is a challenging task and our model was not able to generalize well on the limited available data. To address this issue we introduce gloss level intermediate representations in the form of a tokenization layer to ease the translation task. An overview of this approach is visualized in Figure 1.2. We utilize state-of-the-art CSLR models to predict gloss sequences from sign language videos. We then use these gloss predictions to train gloss to spoken language translation models, which yield much higher translation accuracy (18.13 BLEU-4).

We hypothesise that the main reason of the performance gain is the additional gloss

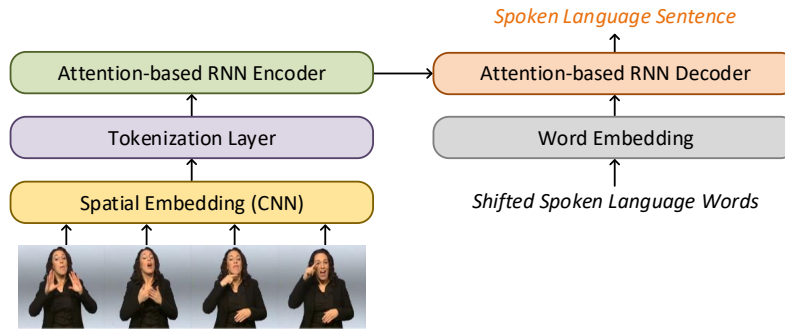


Figure 1.2: An overview of the Neural Sign Language Translation [24] approach.

level supervision that has been introduced by the CSLR models. Furthermore, changing the input sequences from spatial frame representations to fewer one-hot vector representations of sign glosses eases the translation task. However, utilizing this two step approach has its drawbacks. As our models condition translation on the predicted sign glosses, their performance is limited by the accuracy of the preceding CSLR module. This approach also imposes an information bottleneck as gloss annotations do not fully encapsulate the rich linguistic tools and grammatical structures of the sign language.

To be able to utilize gloss supervision without creating an information bottleneck, we reformulate SLT as a multi-task learning problem and introduce Sign Language Transformers in *Chapter 6*. We propose assigning recognition and translation tasks to the sub-networks of the state-of-the-art transformer models. We name these sub-networks as Sign Language Recognition Transformer (SLRT) and Sign Language Translation Transformer (SLTT) with respect to their task. The SLRT is a Connectionist Temporal Classification (CTC) based transformer encoder model which learns to predict sign gloss sequences given the spatial representations of sign video frames. The spatio-temporal representations learned by SLRT are then passed to an autoregressive transformer decoder model, SLTT, which is trained to generate one-spoken word at a time. An overview of the Sign Language Transformer model can be seen in Figure 1.3. We train our networks using gloss annotations and the spoken language translations provided by PHOENIX14T, in an end-to-end manner. We report state-of-the-art CSLR accuracy while significantly improving the translation performance over the Neural Sign Language Translation approach (9.58 vs. 21.32 BLEU-4). Furthermore, we empirically

show that joint training benefits both the CSLR (24.59 vs. 24.49 WER) and the SLT (20.17 vs. 21.32 BLEU-4) sub-tasks.

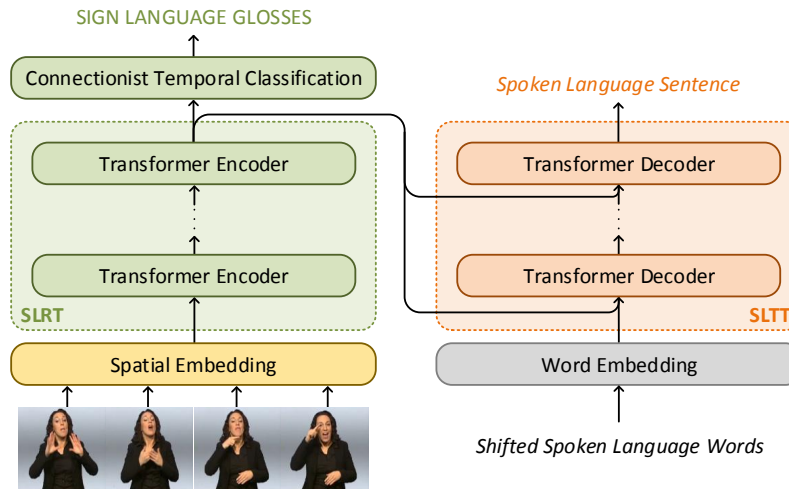


Figure 1.3: An overview of the Sign Language Transformers [27] approach, jointly trained to perform CSLR and SLT.

Encouraged by the translation performance of the Sign Language Transformer, a preliminary study investigates their effectiveness on a larger corpus, namely British Sign Language Corpus Project (BSLCP) [153]. Compared to PHOENIX14T, BSLCP has more annotations (10207 vs. 8257 sequences) and covers a larger domain of discourse (free form conversation and interviews vs. weather forecast interpretations). We conduct experiments using our best performing setup and report results which were drastically worse than the ones we obtain on PHOENIX14T (4.00 vs. 24.54 BLEU-4, See [46] for further information on this study). We think this performance disparity is due to the number of annotations being relatively small with respect to the large domain BSLCP covers (*e.g.* 2890 out of 5348 unique gloss tokens only occur once).

To realize sign language translation systems that are on par with their spoken and written language counter parts, which can translate content from any domain, the SLT field requires more parallel datasets. We believe the best option towards large scale translation is to exploit the sign language interpretations and the accompanied subtitles from daily broadcasts. However, there are two main difficulties one needs to address to be able to train current state-of-the-art approaches using broadcast data: (1) *Alignment*:

The subtitles and the sign language interpretations follow roughly the same temporal information order. However, they are generally not temporally aligned. Thus curating parallel datasets, which current approaches require, from broadcast data will require an additional, perhaps semi-automatic, alignment step. (2) *Annotation*: Broadcasters only provide the spoken language subtitles and the sign language videos. This means that datasets created from the broadcast data will lack the sign gloss annotations which current models heavily rely upon. Given gloss annotation is a laborious task and requires specific sign language expertise, it is not feasible to annotate datasets in large quantities. Furthermore, depending on gloss annotations pose further limitations as these glosses are often language specific, making translation systems and their CSLR sub-modules only applicable to the language they were trained on. Hence, in the final contribution chapter we focus on eliminating the current approach’s dependence on sign gloss annotations. We take inspiration from subunit based Sign Language Recognition (SLR) literature [42, 44, 179]. We propose utilizing specialized models that were trained on individual sign articulators, *e.g.* hand shape recognition networks [101, 23], and then use their learnt representations in combination to model sign in a holistic manner. This approach will also enable us to make use of datasets and models from other relevant computer vision fields, such as human pose estimation [81] and lip reading [37]. Furthermore, these specialized articulator networks will be applicable to modelling other sign languages as well.

To combine subunit representations of multiple articulators and to eliminate the dependence on gloss annotations, in *Chapter 7* we propose a novel deep learning architecture, named Multi-channel Transformers. Aimed at multi-channel sequence-to-sequence problems, our approach builds on the idea of self-attention which learns the contextual relationship within sequences. Given representations of multiple asynchronous subunit articulators, our networks explicitly model inter and intra channel contextual relationships, while being trained to perform sign language translation. An overview of the proposed Multi-channel Transformers in the context of SLT can be seen in Figure 1.4.

We conduct experiments on PHOENIX14T to evaluate the multi-articulatory SLT performance of our approach. We use models that were trained for human pose estimation [81, 207], hand shape classification [101, 96] and viseme recognition [99, 96]



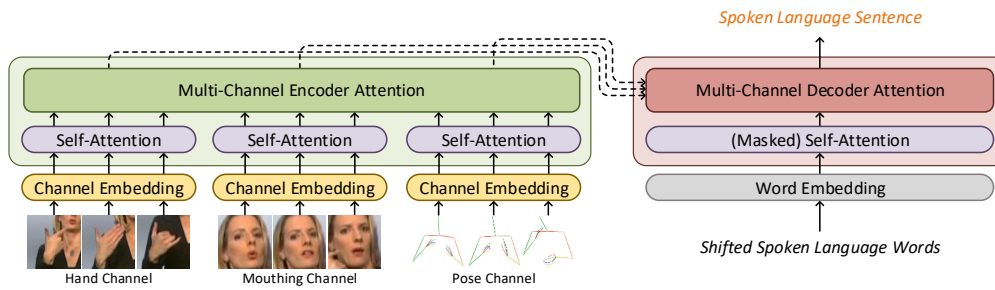


Figure 1.4: An overview of the proposed Multi-channel Transformer [26] architecture applied to the multi-articulatory SLT problem.

to extract representations of manual and non-manual components of sign. We examine the effectiveness of our approach in contrast to naive fusion techniques (early and late) and report superior translation accuracy. Furthermore, when compared against models which utilize gloss level supervision, our approach achieve promising translation performance (18.31 vs. 19.08 BLEU-4). These results not only indicates the importance of modelling the contextual relationship within and between sign channels, but also suggests that it is possible to realize SLT without gloss supervision.

We finally conclude this thesis in Chapter 8 by sharing the conclusions drawn from the work and closing remarks. We also talk about the open questions and future directions of computer vision based sign language research.



## **Chapter 2**

# **Computer Vision based Sign Language Research**

In this chapter we survey the literature on vision based sign language recognition and translation. We first give a brief historical overview of sign language research, starting from recognizing isolated signs to the more recent multi-articulatory Continuous Sign Language Recognition (CSLR) methods. We then review the previous work on Sign Language Translation (SLT) and discuss the contributions of this thesis to the field.

### **2.1 Sign Language Recognition**

Sign Language Recognition (SLR) has been studied by computer vision researchers for the last three decades [177, 165] and it has produced several real life applications such as: TESSA [47], a post office translation application, Dicta-Sign [60], a sign language wiki system, SignTutor [9], an interactive sign language tutoring system, HospiSign [25, 174], a hospital information kiosk for the deaf as well as searchable sign language dictionaries [45, 61]. However, most of these prototypical systems to date focused on isolated sign recognition and spotting.

There are various factors that contribute to previous work focusing on isolated sign. First and foremost is that collection and annotation of CSLR data is a laborious task.

Although there are datasets available from linguistic sources [78, 153] and sign language interpretations from broadcast footage [41, 19], they are either sparsely or weakly annotated and lack sequence level alignment information which SLR methods require. In addition, such datasets lack the human pose information which legacy SLR methods heavily relied on. This has resulted in many researchers collecting their own isolated sign language datasets [127, 25, 204]. Most of these dataset were gathered using consumer depth cameras [209] to be able to utilize the then popular fast and real-time human pose estimation methods [156]. However, the use of controlled environments and a limited vocabulary inhibit the end goal of sign language translation. Another factor is that until recently, a baseline dataset for SLR, had not been established. This led researchers to work on their own small datasets specific for their applications [130, 193, 206, 128], making most of the research incomparable, hence robbing the field of competitive progress.

With the developments in the field of weakly supervised learning [44, 19, 138, 98] and breakthroughs in the field of human body pose estimation [34, 196, 29], working on linguistic data and sign language interpretation from broadcast footage has become a feasible option. Although there were some efforts to develop pose estimation techniques specialized for sign language videos [34], general purpose, Convolutional Neural Network (CNN)-based approaches quickly became the norm [196]. OpenPose library [29, 28] is the most commonly used technique to estimate signers' pose from the videos. Although OpenPose models were not trained on sign language data, due to the utilized bootstrapping techniques [160], they are able to successfully estimate the pose of articulated hand shapes which are common in sign. In their latest work, Hidalgo *et al.* combined the OpenPose body and hand networks into a single architecture, which drastically improved the real-time pose estimation performance [81]. More recently, monocular 3D pose estimation techniques started becoming popular [200]. Compared to multi-view approaches [117, 183], which might not always be applicable, monocular techniques [135, 200] enable the extraction of valuable 3D pose information by fitting canonical body and hand models [89], such as Skinned Multi-Person Linear Model (SMPL) [111] and hand Model with Articulated and Non-rigid defOrmations (MANO) [146], using 2D joint pixel coordinates on the image plane. However, these approaches

---

are still far from being real-time, running on the seconds per frame scale on a modern computer.

These developments were followed by Forster *et al.*'s release of RWTH-PHOENIX-Weather-2012 Dataset (PHOENIX12) [64] and its extended version the RWTH-PHOENIX-Weather-2014 Dataset (PHOENIX14) [65], which contained German Sign Language - Deutsche Gebärdensprache (DGS) interpretations of weather forecasts. With the availability of the PHOENIX datasets, research interest started to shift towards CSLR, and PHOENIX14 quickly became a baseline for continuous SLR, pushing the field towards the goal of continuous SLT to spoken language.

As signs are spatio-temporal constructs, generally all SLR methods consist of two steps: (1) Extraction of spatial features from video frames and (2) Temporal modelling of these representations. Legacy approaches [42, 43], relied heavily on hand crafted features to represent the manual and non-manual aspects of the sign. These features are then modelled using graphical models such as Hidden Markov Models (HMMs) [97, 179], Conditional Random Fields (CRFs) [141] or template-based approaches to capture the temporal changes in the sign sequences [19, 40, 127, 128].

With the rise of deep learning, enthusiasm revived and accelerated the field [22, 101, 103]. The recognition of limited domain but continuous real-life sign language became feasible [23, 48, 86, 102, 49]. Using Deep Learning (DL) methods, which can learn spatio-temporal representations from data with minimal human intervention, researchers swiftly adopted CNNs and Recurrent Neural Network (RNN) based approaches. Koller *et al.* [102] proposed using CNNs to model both the signer as a whole and their hand shapes [101, 103]. They further employed a HMM based forced alignment approach to train their networks from weakly annotated video sequences [102]. Building on this, Camgoz *et al.* [23] and Cui *et al.* [48, 49] proposed utilizing the Connectionist Temporal Classification (CTC) loss function [72], which marginalizes over all possible video and annotation alignments during training, thus eliminating the costly iterative forced alignment procedure.

Concurrently, driven by linguistic evidence [13, 197, 136], the field realized that sign language recognition needs to focus on more than just the hands. Earlier works looked at

several articulators separately, such as the face in general [189, 7, 97], head pose [114], the mouth [7, 99, 100], eye-gaze [30], and body pose [137, 34]. The current state-of-the-art CSLR approaches are based on multi-stream architectures utilizing both manual and non-manual features of the sign [96, 212]. They report significant improvements over methods which only utilize manual features, thus indicating the necessity of incorporating non-manual features.

## 2.2 Sign Language Translation

Generally speaking, much of the available sign translation literature falsely declares sign recognition as sign translation [63, 194, 76, 75]. There have been earlier attempts to realize SLT by computational linguists. However, existing work has solely focused on the text-to-text translation problem and has been very limited in size, averaging around 3000 total words [121, 166, 167, 154]. Using statistical machine translation methods, Stein *et al.* [167] proposed a weather broadcast translation system from spoken German into DGS and vice versa, using the PHOENIX12 [64] dataset. Another method translated air travel information from spoken English to Irish Sign Language (ISL), spoken German to ISL, spoken English to DGS, and spoken German to DGS [120]. Ebling [56] developed an approach to translate written German train announcements into Swiss German Sign Language - Deutschschweizer Gebärdensprache (DSGS). While non-manual information has not been included in most previous systems, Ebling & Huenerfauth [59] proposed a sequence classification based model to schedule the automatic generation of non-manual features after the core machine translation step.

Conceptual video based SLT systems were introduced in the early 2000s [20]. There have been studies, such as [31], which propose recognizing signs in isolation and then constructing sentences using a language model. However, end-to-end SLT from continuous sign videos has not been realized until this thesis.

The most important obstacle to vision based SLT research has been the availability of suitable datasets. There are datasets with spoken language translations available from linguistic sources [153, 78] and sign language interpretations from broadcasts

---

[41]. However, the available annotations are either weakly aligned (subtitles) or too few to build models which would work on a large domain of discourse. Furthermore, the relationship between sign sentences and their spoken language translations are non-monotonic, as they have different ordering. Also, sign glosses and linguistic constructs do not necessarily have a one-to-one mapping with their spoken language counterparts. This made the use of available CSLR methods [103, 102] (that were designed to learn from weakly annotated data) infeasible, as they are built on the assumption that sign language videos and corresponding annotations share the same temporal order.

To address these issues, in collaboration with RWTH-Aachen, we introduce the first publicly available SLT dataset, RWTH-PHOENIX-Weather-2014T Dataset (PHOENIX14T) [24], which is an extension of the popular PHOENIX14 CSLR dataset (See Chapter 4). We approach the SLT task as a spatio-temporal Neural Machine Translation (NMT) problem, which we term '*Neural Sign Language Translation*'. We propose a system using CNNs in combination with attention-based NMT methods [113, 11] to realize the first end-to-end SLT models. Although the end-to-end approach produce meaningful translations, utilizing an initial CSLR tokenization step, and thus converting the problem into a text-to-text translation task, yields significantly improved results (See Chapter 5). However, sign languages are visual constructs and trying to represent them with text introduces an information bottleneck. To address this issue we reformulate SLT as a multi-task problem to incorporate gloss supervision without limiting the information flow. We utilize state-of-the-art transformer models and train sub-networks of our *Sign Language Transformer* models jointly to predict sign gloss sequences and spoken language sentences. This end-to-end approach outperform previous translation benchmarks and is the current state-of-the-art on PHOENIX14T (See Chapter 6).

Following our work, Orbay and Akarun [129] investigated different tokenization methods on PHOENIX14 and showed that a pretrained hand shape recognizer [101] outperforms simpler approaches and reaches 14.6 BLEU-4. While they also investigated transformer architectures and multiple hands as input, the results under-performed. Ko *et al.* [95] describe a non-public dataset covering sign language videos, gloss annotation and translation. Their method relies on detected body key-points only. It hence misses the important appearance based characteristics of sign.

Overall, previous work in the space of SLT had two major short-comings: (1) The beauty of translations is the abundance of available training data as they can be created in real-time by interpreters. Glosses are expensive to create and limit data availability. No previous work was able to achieve state-of-the-art performance while not relying on glosses. (2) So far SLT has never considered multiple articulators. To address these issues, we introduce Multi-channel Transformers to realize the first multi-articulatory SLT without utilizing gloss level supervision (See Chapter 7). We propose utilizing specialized models from related fields, such as human pose estimation and hand shape recognition, and model the inter and intra contextual relationship of articulator channels. We evaluate our approach on PHOENIX14T and train our models using human pose, hand shape and viseme representations. Our networks achieve superior translation performance over single channel models indicating the importance of considering multiple articulators. Furthermore, we report comparable results against models which utilize gloss supervision, suggesting that the field’s dependency on gloss annotations may come to an end soon.



## **Chapter 3**

# **Modern Deep Learning**

## **Approaches and Their Applications**

Deep Learning (DL) [70] has gained popularity and achieved state-of-the-art performance in various fields such as Computer Vision [105], Speech Recognition [5] and more recently in the field of Machine Translation [123]. To realize Continuous Sign Language Recognition (CSLR) and Sign Language Translation (SLT) models are required that can recognize the spatio-temporal linguistic constructs of sign languages and their mapping to spoken languages. Hence, in this chapter we give a brief background on the state-of-the-art in DL and its applications to computer vision, sequence-to-sequence learning and neural machine translation.

### **3.1 Convolutional Neural Networks**

In the computer vision field, DL methods, more specifically Convolutional Neural Networks (CNNs) [107], first became popular for the object recognition task [53, 105]. Prior to DL methods, researchers were devising hand crafted features, such as the Histograms of Oriented Gradients (HOG) [51], Scale Invariant Feature Transform (SIFT) [112] or Speeded Up Robust Features (SURF) [12], to represent spatial information in images. These features were not task specific and were biased by human intuition. DL

approaches like CNNs obviated the necessity of designing such features, as they are able to learn spatial representations from images while being trained to accomplish a task such as object recognition.

Inspired by the feline visual cortex, the first CNN, Neurocognitron, was introduced in the early 1980's [67]. Lecun *et al.* [107] later proposed LeNet and utilized back propagation [108] to read hand written letters and digits. Although there has been continuous work on CNNs throughout the years, it was not popular in computer vision for decades due to the lack of large scale dataset neural networks require to generalize and efficient hardware to train those networks on. However, they gained popularity after Krizhevsky *et al.* [105] drastically outperformed its competitors with the AlexNet architecture in the ImageNet 2012 challenge [53]. From this point on CNNs became an ubiquitous tool in computer vision research and it is now being used in various fields, such as speech recognition [2], image captioning [201], visual question answering [6], semantic segmentation and [147] much more [70].

There have been several improvements in CNNs architectures since AlexNet [105] in recent years. One of the main objectives for developing new models was to have deeper architectures. However, as noted by Simonyan *et al.*, going deeper doesn't necessary improve the performance and in some cases worsens it [162]. One of the main factors to this was the issue of vanishing gradients. As the networks got deeper, the error signals got weaker due to the non-linear nature of activation functions. To overcome this Szegedy *et al.* [176] proposed having multiple losses for the same objective spread through the network, thus allowing the error signals to propagate further. Another approach to overcome the vanishing gradient problem was to utilize skip connections proposed by He *et al.* [79]. These skip connections, commonly known as residual connections, allowed the training of networks much deeper than was previously possible, even going up to 1000 layers [80]. However, the performance gain by building deeper networks saturates. To address this, Huang *et al.* [85] proposed densely connected wide networks and reported improved results over ResNets trained with residual connections. While networks continued to become deeper and wider, researchers started to realize most of the parameters in these networks are actually redundant [66]. More recently Tan *et al.* [178] extensively studied the balancing of network depth, width and resolution

---

and proposed EfficientNets which obtained state-of-the-art performance with fewer parameters.

Two of the most relevant fields to sign language recognition are human action recognition [141] and gesture recognition [143]. With the emergence of CNN methods and availability of large dataset [163, 192, 148], the research into action recognition started to become dominated by deep architectures [161, 92, 182]. Similarly in the field of gesture recognition, research focus shifted towards DL solely relying on the representation of capabilities of the deep architectures [32, 140, 124].

In this thesis, we utilize CNNs as a Spatial Embedding layer to extract representations of signers from video frames. We explore a variety of architectures, ranging from the earlier AlexNet (Chapter 5), the more recent Inception models (Chapters 6 and 7) and the state-of-the-art EfficientNets (Chapters 6).

## 3.2 Sequence-to-Sequence Learning

One of the most important breakthroughs in DL was the development of sequence-to-sequence learning approaches. Strong annotations are hard to obtain for sequence-to-sequence tasks, in which the objective is to learn a mapping between two sequences. Earlier methods in the field of speech recognition proposed combining neural networks with graphical models, such as Hidden Markov Model (HMM), to learn to recognize and align speech signals and their phoneme representations [145]. To be able to train from weakly annotated data in an end-to-end manner, Graves *et al.* proposed the Connectionist Temporal Classification (CTC) loss function [72], which considers all possible alignments between two sequences while calculating the error. CTC quickly became a popular loss function for many sequence-to-sequence applications. It has obtained state-of-the-art performance on several tasks in speech [74, 5] and handwriting recognition [73]. Computer vision researchers adopted CTC and applied it to weakly labeled visual problems, such as sentence-level lip reading [10], action recognition [84], hand shape recognition [23] and CSLR [23, 48, 49]. In this thesis, we utilize CTC loss in our multi-task Sign Language Transformer approach to train our transformer encoder models using weakly labelled gloss annotations (See Chapter 6).

Another common sequence-to-sequence task is machine translation, which aims to develop methods that can learn the mapping between two languages. The field of machine translation, as with any other field which employs statistical modelling, has been drastically changed by the development of modern DL approaches [70]. Although CTC has proven to be successful in several tasks, it is not suitable for machine translation as CTC assumes source and target sequences share the same order. Furthermore, CTC does not explicitly model the conditional dependence within target sequences, which in turn does not allow networks to learn an implicit target language model. This has led to the development of Encoder-Decoder Network architectures [91] and the emergence of the Neural Machine Translation (NMT) field [123], which will be discussed in detail in the next section.

### 3.3 Neural Machine Translation

The objective of machine translation is to learn the conditional probability  $p(\mathcal{Y}|\mathcal{X})$  where  $\mathcal{X} = (x_1, \dots, x_T)$  is a sentence from source language with  $T$  tokens and  $\mathcal{Y} = (y_1, \dots, y_U)$  is the desired corresponding translation of said sentence in the target language. To learn this mapping using neural networks, Kalchbrenner *et al.* [91] proposed using an *encoder-decoder* architecture, where the source sentence is encoded into a fixed sized “context” vector which is then used to decode the target sentence. They realized this using a Convolutional n-gram Model (CGM) for the encoder and the hybrid approach consisting of an inverse CGM and a Recurrent Neural Network (RNN) as the decoder.

Cho *et al.* [36] and Sutskever *et al.* [172] further improved the encoder-decoder architectures by assigning the encoding and decoding stages of translation to individual specialized RNNs. The context vector in this case is the hidden state of the encoder after processing the last source token. This context vector is then used to initialize the decoder’s hidden state which is trained to generate the target sentence in an auto-regressive manner. The decoding process starts with a unique beginning of sentence ( $\langle \text{bos} \rangle$ ) token and continues until another unique end-of-sentence ( $\langle \text{eos} \rangle$ ) token is produced. A visualization of this approach can be seen in Figure 3.1.

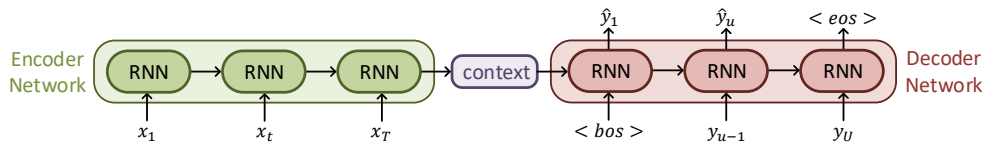


Figure 3.1: A generic RNN-based encoder-decoder architecture for machine translation.

Although encoder-decoder networks improved machine translation performance, there remains the issue of an information bottleneck caused by encoding the source sequence into a fixed sized vector and the long term dependencies between source and target sequences. As RNNs accumulate the context vector by considering one token at a time, the information contribution of the earlier tokens in the sequence vanish as the source sentence comes to an end. Although there have been practical solutions to this, such as source sentence reversing [172], the context vector is still of fixed size, and thus cannot perfectly encode arbitrarily long input sequences.

To overcome the information bottleneck imposed by using the last hidden state of the RNN as the context vector, Bahdanau *et al.* [11] proposed an attention mechanism, which was a breakthrough in the field of NMT. The idea behind the attention mechanism is to use a soft-search over the encoder outputs at each step of target sentence decoding. This was realized by conditioning target word prediction on a context vector which is a weighted sum of the source sentence representations. The weighting in turn is done by a learnt scoring function which measures the relevance of the encoder outputs and the decoders current hidden state. Luong *et al.* [113] further improved this approach by proposing a dot product attention (scoring) function as:

$$\text{context} = \text{softmax}(QK^T) V \quad (3.1)$$

where *Queries* correspond to the hidden state of the decoder at a given time step, and *Keys* and *Values* represent the encoder outputs (See Figure 3.2). We utilize both of these attention mechanisms in realising the first end-to-end SLT approach, Neural Sign Language Translation [24], and explain them in further detail in Section 5.1.

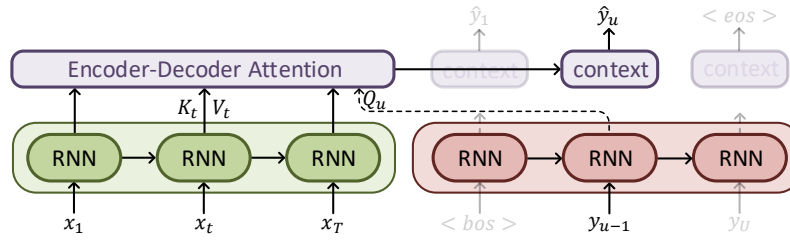


Figure 3.2: Dot product attention-based encoder-decoder architecture proposed by Luong *et al.* [113]

Various attention based architectures have been proposed for NMT, such as Google’s Neural Machine Translation (GNMT) [199] which combines bi-directional and uni-directional encoders in a deep architecture. One of the key differences implemented in GNMT was only using the attention function to pass information between the encoder and the decoder networks, thus ignoring the fixed length representation stored in the recurrent units’ hidden state. GNMT has been further extended to do Zero-Shot translation and was able to translate between language pairs it was not explicitly trained on [88].

Similar sequence-to-sequence and attention based approaches have been applied to various computer vision tasks. Xu *et al.* proposed the “Show, Attend and Tell” architecture, where they extract spatial features from images using CNNs and use a sequence-to-sequence model to map image patch features to captions [201]. Donahue *et al.* introduced the CNN-Long Short-Term Memory (LSTM) hybrid “Long-term Recurrent Convolutional Network” architecture for action recognition and video description generation [55]. Venugopalan *et al.* further extended this approach by utilizing RNN-based encoder-decoder models in combination with CNNs and applied it to various video-to-text tasks [188]. Chung *et al.* built up on the video-to-text approach and introduced “Watch, Listen, Attend, and Spell” architecture and applied it to multi-modal lip-reading sentences in the wild task [37].

More recently, researchers proposed a convolution based sequence-to-sequence learning approach [68] for machine translation. The motivation behind this approach was to allow the encoder to have larger receptive fields in the temporal domain as it goes

deeper in the network. This approach yielded better performance compared to RNN based methods while being able to train and infer faster.

One of the most recent breakthroughs in NMT was the introduction of transformer networks [187]. Vaswani *et al.* proposed modelling the contextual relationship within source and target sequences using dedicated attention mechanisms. To realize this, they proposed a new self-attention mechanism, which refines the source and target token representations by looking at the context they have been used in. Combining encoder and decoder self-attention layers with an encoder-decoder attention, Vaswani *et al.* proposed Transformer networks, a fully connected network (as opposed to being RNN-based) which has revolutionized the field of machine translation. They also utilize multi-head attentions by separating their inputs into multiple chunks and learning different attention weights for each chunk, allowing the networks to learn multiple contextual relationships between words. A simplified visualization of the transformer architecture can be seen in Figure 3.3.

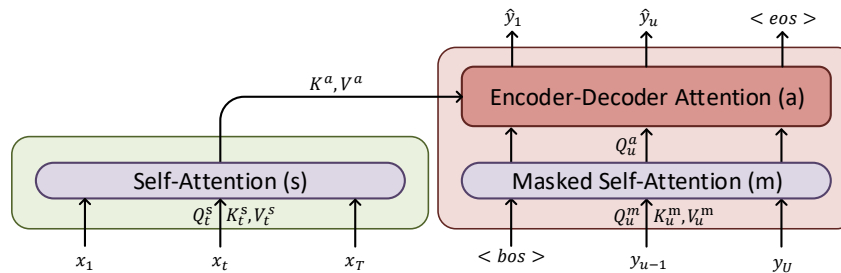


Figure 3.3: Transformer Networks proposed by Vaswani *et al.* [187]

In contrast to RNN-based models, transformers obtain  $Q$ ,  $K$  and  $V$  values by using individually learnt linear projection matrices at each attention layer. Vaswani *et al.* also introduced the “scaled” dot-product attention as:

$$\text{context} = \text{softmax} \left( \frac{QK^T}{\sqrt{d_m}} \right) V \quad (3.2)$$

where  $d_m$  is the number of hidden units of the model. The motivation behind the scaling operation is to counteract the effect of gradients becoming extremely small in cases where the number of hidden units is high and in-turn, the dot products grow large [187].

In addition to NMT, transformers have achieved success in various other tasks. Dai *et al.* further extended the transformer architecture and introduced Transformer-XLs, which was able to learn longer temporal dependencies compared to both RNN and transformer based approaches [50]. Devlin *et al.* proposed an unsupervised pre-training approach, Bidirectional Encoder Representations from Transformers (BERT), to learn sentence representations from large scale textual datasets [54]. Zhang *et al.* incorporated Knowledge Graphs into the training of BERT models to enhance language representation and reported significant improvements on various knowledge driven tasks [208]. More recently, Clark *et al.* proposed ELECTRA, which replaces the masked language modelling pre-training scheme of BERT with a modified token detection approach to improve sample efficiency [39]. Transformers have also been used for computer vision tasks. Tsai *et al.* proposed multi-modal transformers for multi-modal language understanding, working on unaligned video, text and audio sequences [184]. Riberio *et al.* introduced Sketchformers, which encodes free-hand sketch input into a latent vector space for multiple down-stream tasks, such as sketch classification, sketch based image retrieval, and the reconstruction and interpolation of sketches [144]. Similar to the self-attention layer of transformer networks, Wang *et al.* proposed non-local blocks for activity recognition from videos [195]. Sun *et al.* modified the BERT architecture for visual-linguistic tasks, and introduced VideoBERT, which achieved state-of-the-art performance on video captioning [171].

In Chapter 6, we propose a multi-task reformulation of CSLR and SLT based on transformer networks. For CSLR, we train transformer encoders using CTC loss to predict sign gloss sequences from continuous sign language videos. We then pass the spatio-temporal representations learnt by the encoder to a transformer decoder, which is trained to generate one spoken language word at a time, to realize SLT. We report significant performance improvements over RNN-based models, and display the benefits of the proposed multi-task learning framework.

We further extend the transformer network architecture and adapt it to the task of multi-articulatory SLT in Chapter 7. In accordance with the original motivation of the transformer, we believe information coming from different source channels should be modeled in the context of that channel (*i.e.* channel-wise self-attention). However, we



---

further propose a multi-channel attention layer to refine the representations of each source channel in the context of other source channels. We also adapt the encoder-decoder attention layer to be able to use multiple source channel representations. In the case of SLT, these channels correspond to sign articulators, namely hands, body and mouthings.



## Chapter 4

# Sign Language Translation Dataset and Evaluation Protocols

In this chapter we present the RWTH-PHOENIX-Weather-2014T Dataset (PHOENIX14T)<sup>1</sup> [24], a Continuous Sign Language Recognition (CSLR) and Sign Language Translation (SLT) corpus, which we utilize to evaluate the performance of our approaches. Curated by our collaborators in RWTH-Aachen, PHOENIX14T is an extension of the RWTH-PHOENIX-Weather-2014 Dataset (PHOENIX14), which has become the primary benchmark for CSLR in recent years. PHOENIX14T constitutes a parallel corpus including German Sign Language - Deutsche Gebärdensprache (DGS) videos, sign gloss annotations and also German translations (spoken by the news anchor), which makes it the only available dataset suitable for training and evaluating joint SLR and SLT techniques. Due to different sentence segmentation between spoken language and sign language, it was not sufficient to simply add a translation tier to PHOENIX14. Instead, the segmentation boundaries also have been redefined. Wherever the addition of a translation layer necessitated new sentence boundaries, forced alignment approaches have been used [101] to compute the new boundaries.

In addition to changes in boundaries, PHOENIX14T has a marginally decreased sign gloss vocabulary due to some improvements in the normalization schemes. This means performance on PHOENIX14 and PHOENIX14T will be similar, but not exactly

---

<sup>1</sup><https://www-i6.informatik.rwth-aachen.de/~koller/RWTH-PHOENIX-2014-T/>

comparable. However, care has been taken to assure that the development and test sets of PHOENIX14 do not overlap with the new PHOENIX14T training set and also that none of the new development and test sets from PHOENIX14T overlap with the PHOENIX14 training set.

	Sign Gloss			German		
	Train	Dev	Test	Train	Dev	Test
segments	7,096	519	642	←———— <i>same</i>		
frames	827,354	55,775	64,627	←———— <i>same</i>		
vocab.	1,066	393	411	2,887	951	1,001
tot. words	67,781	3,745	4,257	99,081	6,820	7,816
tot. OOVs	-	19	22	-	57	60
singletons	337	-	-	1,077	-	-

Table 4.1: Key statistics of the RWTH-PHOENIX-Weather-2014T Dataset.

This corpus is publicly available to the research community to facilitate the future growth of SLT research. The detailed statistics of the dataset can be seen in Table 4.1. OOV stands for Out-Of-Vocabulary, *e.g.* words that occur in test, but not in training. Singletons occur only once in the training set. The corpus covers unconstrained sign language of nine different signers with a vocabulary of 1066 different signs and translations into German spoken language with a vocabulary of 2887 different words. The corpus features professional sign language interpreters and has been annotated using sign glosses by deaf specialists. The spoken German translation originates from the news speaker. It has been automatically transcribed, manually verified and normalized.

## 4.1 Evaluation Protocols

To underpin future work on PHOENIX14T and to be able to better understand the shortcomings of the proposed models, we lay down several protocols for researches to evaluate their CSLR and SLT approaches:

*Sign2Gloss* is a protocol which essentially performs CSLR. The goal of this task is to

---

predict sequences of sign glosses from continuous sign language videos.

***Sign2Text*** is the end goal of SLT, where the objective is to translate directly from continuous sign videos to spoken language sentences without utilizing any intermediary representation, such as text-based sign glosses.

***Gloss2Text*** is a text-to-text translation protocol, where the objective is to translate ground truth sign gloss sequences to German spoken language sentences. The results of these experiments act as a virtual upper bound for the available Neural Machine Translation (NMT) technology. This assumption is based on the fact that perfect sign language recognition/understanding is simulated by using the ground truth gloss annotation. However, as mentioned earlier, one needs to bear in mind that gloss representations are imprecise. As glosses are textual representations of multi-channel temporal signals, they represent an information bottleneck for any translation system. This means that under ideal conditions, a *Sign2Text* system could and should outperform *Gloss2Text*. However, more sophisticated network architectures and data are needed to achieve this and hence such a goal remains a longer term objective beyond the scope of this thesis.

***Sign2Gloss2Text*** is the current state-of-the-art in SLT. This approach utilizes CSLR models to extract gloss sequences from sign language videos. The predicted glosses are then used to solve the translation task as a text-to-text problem by training a *Gloss2Text* network.

***Sign2Gloss*→*Gloss2Text*** is similar to *Sign2Gloss2Text* and also uses CSLR models to extract gloss sequences. However, instead of training text-to-text translation networks from scratch, *Sign2Gloss*→*Gloss2Text* models use the best performing *Gloss2Text* network, which has been trained with ground truth gloss annotations. The motivation behind this protocol is to evaluate the feasibility of separating the training of *Sign2Gloss* and *Gloss2Text*, enabling us to utilize auxiliary data sources to train individual modules.

***Sign2(Gloss+Text)*** is a multi-task learning protocol similar to *Sign2Gloss2Text*. However, unlike *Sign2Gloss2Text*, which utilizes a two stage pipeline, i.e. CSLR and then text-to-text translation, *Sign2(Gloss+Text)* models are jointly trained to perform CSLR and SLT, simultaneously. The motivation behind this protocol is to evaluate the effects

of utilizing a joint training regime over training each module individually.

## 4.2 Performance Metrics

We utilize various metrics to measure the performance of the proposed approaches. Following the evaluation protocols set by PHOENIX14 [65], we use Word Error Rate (WER) to calculate the accuracy of our CSLR models. To measure our translation performance we utilize Bilingual Evaluation Understudy (BLEU) [132] and Recall-Oriented Understudy for Gisting Evaluation (ROUGE) [110] scores, which are commonly used metrics for machine translation.

### 4.2.1 Word Error Rate (WER)

WER, also known as the length normalized edit distance, is one of the most common metrics used in speech recognition [4] and machine translation [4]. Based on the Levenshtein alignment between the ground truth and predicted sign gloss sequences, WER is calculated as:

$$\text{WER} = \frac{|\mathcal{G}^d| + |\mathcal{G}^i| + |\mathcal{G}^s|}{|\mathcal{G}^*|} \quad (4.1)$$

where  $|\mathcal{G}^d|$ ,  $|\mathcal{G}^i|$  and  $|\mathcal{G}^s|$  represent the required numbers of deletion, insertion and substitution operations to transform the predicted sign glosses into the ground truth sequence  $\mathcal{G}^*$  with a cardinality of  $|\mathcal{G}^*|$ , respectively.

### 4.2.2 Bilingual Evaluation Understudy (BLEU)

The BLEU score was proposed to substitute skilled human judges for rapid evaluation of machine translation systems. To measure the BLEU score of a generated sentence  $\tilde{\mathcal{S}}$  given the reference sentence  $\mathcal{S}$ , we first extract  $n$ -grams,  $\tilde{\mathcal{S}}^n$  and  $\mathcal{S}^n$ , from both of the sentences, respectively. Commonly used in the fields of computational linguistics and natural language processing,  $n$ -grams are contiguous sequences of  $n$  units, *i.e.*

characters, words or tokens, given a text sample. A sample  $n$ -gram extraction can be seen in Table 4.2.

Type	Unit	Input	$\mathcal{S}^{n=1}$ (unigrams)	$\mathcal{S}^{n=2}$ (bigrams)	$\mathcal{S}^{n=3}$ (trigrams)
Sentence	Word	'to be or'	'to', 'be', 'or'	'to be', 'be or'	'to be or'
Word	Character	'CAT'	'C', 'A', 'T'	'CA', 'AT'	'CAT'

Table 4.2: Example  $n$ -gram extractions with the lengths of one, two and three.

Using the extracted  $n$ -grams, we calculate the precision,  $p_n$ , of a predicted sentence as:

$$p_n = \frac{|\mathcal{S}^n \cap \tilde{\mathcal{S}}^n|}{|\tilde{\mathcal{S}}^n|} \quad (4.2)$$

where  $|\tilde{\mathcal{S}}^n|$  and  $|\mathcal{S}^n \cap \tilde{\mathcal{S}}^n|$  represent the cardinality of  $\tilde{\mathcal{S}}^n$  and number of matching items between  $\tilde{\mathcal{S}}^n$  and  $\mathcal{S}^n$ , respectively. We then calculate the BLEU- $N$  score of a prediction-reference sentence pair as:

$$\text{BLEU-}N = \text{BP} \cdot \exp\left(\frac{1}{N} \sum_{n=1}^N \log p_n\right) \quad (4.3)$$

where  $N$  is the maximum length of  $n$ -grams that is being considered and BP is a brevity penalty computed as:

$$\text{BP} = \begin{cases} 1 & \text{if } |\tilde{\mathcal{S}}| > |\mathcal{S}| \\ e^{(1-|\mathcal{S}|/|\tilde{\mathcal{S}}|)} & \text{else} \end{cases} \quad (4.4)$$

Here  $|\tilde{\mathcal{S}}|$  and  $|\mathcal{S}|$  correspond to the number words in predicted sentence,  $\tilde{\mathcal{S}}$ , and reference sentence,  $\mathcal{S}$ , respectively.

### 4.2.3 Recall-Oriented Understudy for Gisting Evaluation (ROUGE)

The ROUGE score was originally proposed to automatically determine the quality of computer generated summaries by comparing them to other ideal, human created summaries. The evaluation metrics come with a number of variants, namely, ROUGE- $N$

( $n$ -gram based co-occurrence statistics), ROUGE-S (skip-bigram-based co-occurrence statistics), ROUGE-L (Longest Common Subsequence (LCS)-based statistics), and ROUGE-W (weighted LCS-based statistics that favours consecutive LCSes).

In our experiments, we utilized the ROUGE-L F1 score which measures the longest matching sequence of words. One of the advantages of using LCS-based statistics is that they do not penalize non-consecutive matches, while still considering in-sequence matches that reflect sentence level word order into consideration. We start calculating the ROUGE-L F1 score by measuring the LCS-based precision and recall as:

$$p_{lcs} = \frac{|\text{LCS}(\mathcal{S}, \tilde{\mathcal{S}})|}{|\tilde{\mathcal{S}}|} \quad (4.5)$$

$$r_{lcs} = \frac{|\text{LCS}(\mathcal{S}, \tilde{\mathcal{S}})|}{|\mathcal{S}|} \quad (4.6)$$

where the LCS function outputs the longest common subsequences between the reference sentence,  $\mathcal{S}$ , and the predicted sentence  $\tilde{\mathcal{S}}$ . We then use the precision and recall values to calculate the ROUGE-L F1 score as:

$$\text{ROUGE-L F1} = \frac{2 * p_{lcs} * r_{lcs}}{p_{lcs} + r_{lcs}} \quad (4.7)$$



## Chapter 5

# Neural Sign Language Translation

Despite common misconceptions, sign languages have their own specific linguistic rules [168] and do not translate to/from spoken languages as word to sign or vice versa. While predicting the sign language glosses provides the meaning and the order of signs in the video, the spoken language equivalent (which is what is actually desired) has both a different length and ordering. Therefore, the numerous advances in Sign Language Recognition (SLR) [43], discussed in Chapter 2, and the progress made towards Continuous Sign Language Recognition (CSLR) [97, 102], do not allow us to provide meaningful interpretations of what a signer is trying to convey.

Contrary to SLR, we propose to approach Sign Language Translation (SLT) as a Neural Machine Translation (NMT) task, and coin the term *Neural Sign Language Translation*. We employ Recurrent Neural Network (RNN) based sequence-to-sequence learning models, which were state-of-the-art in the time of this research, in combination with Convolutional Neural Networks (CNNs) to learn: 1) The spatio-temporal representation of the signs, 2) the relation between these signs (in other words the language model) and 3) how these signs map to the spoken or written language. To achieve this we introduce new vision methods, which mirror the tokenization and embedding steps of standard NMT. Using the RWTH-PHOENIX-Weather-2014T Dataset (PHOENIX14T), we realize the first end-to-end continuous sign language video to spoken language text translation approach. We also conduct a wide range of experiments using the evaluation protocols laid down in Section 4.1 to underpin future research in the field of SLT.

The rest of this chapter is structured as follows: We first introduce the proposed *Neural Sign Language Translation* approach in Section 5.1. We then describe our experimental setup and report quantitative results in Section 5.2. We share qualitative examples in Section 5.3 to give the readers further insight on the performance of our approach. We finally conclude this chapter in Section 5.4 by discussing our findings and possible future work.

## 5.1 Methodology

Translating sign videos to spoken language is a sequence-to-sequence learning problem by nature. Our objective is to learn the conditional probability  $p(\mathcal{S}|\mathcal{V})$  of generating a spoken language sentence  $\mathcal{S} = (w_1, w_2, \dots, w_U)$  with  $U$  number of words given a sign video  $\mathcal{V} = (I_1, I_2, \dots, I_T)$  with  $T$  number of frames. This is not a straight forward task as the number of frames in a sign video is much higher than the number of words in its spoken language translation (*i.e.*  $T \gg U$ ). Furthermore, the alignment between sign and spoken language sequences are usually unknown and non-monotonic. In addition, unlike other translation tasks that work on text, our source sequences are videos. This renders the straightforward use of classic RNN-based sequence modeling architectures, which work on text embeddings, difficult. Instead, we propose combining CNNs with *attention-based encoder-decoders* to model the conditional probability  $p(\mathcal{S}|\mathcal{V})$ . We experiment with training our approach in an end-to-end manner to jointly learn the alignment and the translation of sign language videos to spoken language sentences. An overview of our approach can be seen in Figure 5.1. In the remainder of this section, we will describe each component of our architecture in detail.

### 5.1.1 Spatial and Word Embeddings

NMT methods start with tokenization of source and target sequences and then projecting them to a continuous space using word embeddings [118]. The main idea behind using word embeddings is to transform the sparse one-hot vector representations, where each word is equidistant from each other, into a denser form, where words with similar

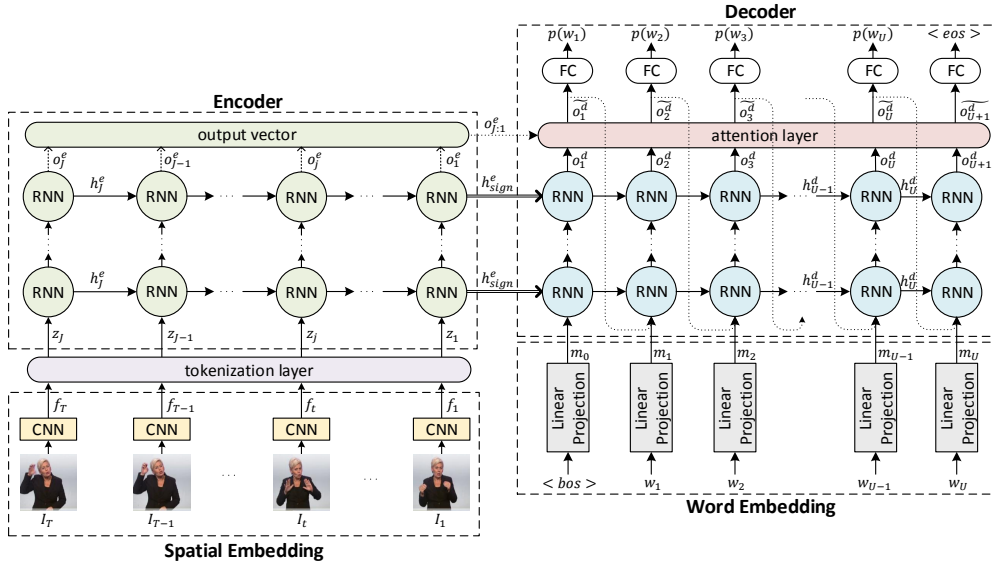


Figure 5.1: An overview of our Sign Language Translation approach that generates spoken language translations of sign language videos.

meanings are closer. These embeddings are either learned from scratch or pretrained on larger datasets and fine-tuned during training. However, contrary to text, signs are visual. Therefore, in addition to using word embeddings for our target sequences (spoken language sentences), we need to learn spatial embeddings to represent sign videos. To achieve this we utilize  $2D$  CNNs. Given a sign video  $\mathcal{V}$ , our CNN learns to extract non-linear frame level spatial representations as:

$$f_t = \text{SpatialEmbedding}(I_t) \quad (5.1)$$

where  $f_t$  corresponds to the feature vector produced by propagating a video frame  $I_t$  through our CNN.

For word embedding, we use a Fully Connected (FC) layer that learns a linear projection from one-hot vectors of spoken language words to a denser space as:

$$m_u = \text{WordEmbedding}(w_u) \quad (5.2)$$

where  $m_u$  is the embedded representation of the spoken word  $w_u$ .

### 5.1.2 Tokenization Layer

In NMT, the input and output sequences can be tokenized at many different levels of complexity: characters, words,  $n$ -grams or phrases. Low level tokenization schemes, such as the character level, allow smaller vocabularies to be used, but increase the complexity of the sequence modeling problem, and require long term relationships to be maintained. High level tokenization makes the recognition problem far more difficult due to vastly increased vocabularies, but the language modeling generally only needs to consider a small number of neighboring tokens.

As there has been no previous research on SLT directly from sign videos, it is not clear what tokenization schemes are most appropriate for this problem. This is exacerbated by the fact that, unlike NMT research, there is no simple equivalence between the tokenizations of the input sign video and the output text. The framework developed in this chapter is generic and can use various tokenization schemes on the spatial embeddings sequence  $f_{1:T}$  as:

$$z_{1:J} = \text{Tokenization}(f_{1:T}) \quad (5.3)$$

In this chapter we explore both “frame level” and “gloss level” input tokenization, with the latter exploiting a RNN-Hidden Markov Model (HMM) based forced alignment approach [102]. As in most modern NMT research, the output tokenization is at the word level, but it could be an interesting avenue for the future to investigate subword unit tokenization [106].

### 5.1.3 Attention-based Recurrent Encoder-Decoder Networks

To be able to generate the target sentence  $\mathcal{S}$  from tokenized embeddings  $z_{1:J}$  of a sign video  $\mathcal{V}$ , we need to learn a mapping function  $\mathcal{B}(z_{1:J}) \rightarrow \mathcal{S}$  which will maximize the conditional probability  $p(\mathcal{S}|\mathcal{V})$ . We propose modelling  $\mathcal{B}$  using an *attention-based recurrent encoder-decoder network*, which is composed of two specialized deep RNNs. By using these RNNs we break down the task into two phases. In the *encoding* phase, a

sign videos' features are projected into a latent space in the form of a fixed size vector, later to be used in the *decoding* phase for generating spoken sentences.

During the *encoding* phase, the *encoder* network, reads in the feature vectors one by one. Given a sequence of tokenized representations  $z_{1:J}$ , we first reverse its order on the temporal axis, as suggested by [172], to shorten the long term dependencies between the beginning of the sign video and spoken language sentence. We then feed the reversed sequence  $z_{J:1}$  to the encoder which models the changes in tokens and compresses their cumulative representation in its hidden states while generating an output for each input as:

$$o_j^e, h_j^e = \text{Encoder}(z_j, h_{j+1}^e) \quad (5.4)$$

where  $o_j^e$  and  $h_j^e$  are the output and the hidden state produced by the recurrent units of the encoder given the token representation  $z_j$  and the hidden state from the previous step<sup>1</sup>  $h_{j+1}^e$ . The initial hidden state of the encoder,  $h_{J+1}^e$ , is set to zero and the final encoder state,  $h_1^e$ , corresponds to the latent embedding of the whole sequence,  $h_{sign}^e$ , which is passed to the decoder.

The *decoding* phase starts by initializing the hidden states of the *decoder* network with the latent vector  $h_{sign}^e$ . In the classic encoder-decoder architecture [172], this latent representation is the only information source of the decoding phase. By taking its previous hidden state,  $h_u^d$ , and the word embedding,  $m_u$ , of the previously predicted word  $w_u$  as inputs, the decoder generates an output  $o_{u+1}^d$ , which represents the embedding of the next word in the sequence,  $w_{u+1}$ , and updates its hidden state,  $h_{u+1}^d$ :

$$o_{u+1}^d, h_{u+1}^d = \text{Decoder}(m_u, h_u^d). \quad (5.5)$$

Here  $h_0^d = h_{sign}^e$  is the spatio-temporal representation of sign language video learned by the encoder and  $m_0$  is the embedding of the special token  $w_0 = \langle \text{bos} \rangle$  indicating the beginning of a sentence. Outputs of the decoder are then fed to a FC layer which produces probabilities over the vocabulary for each word as :

<sup>1</sup>Note that  $j + 1$  is the previous step of  $j$  due to reversed input sequence  $z_{J:1}$ .

$$p(w_u) = \text{FullyConnected}(o_u^d) \quad (5.6)$$

This procedure continues until the decoder predicts another special token,  $\langle \text{eos} \rangle$ , which indicates the end of a sentence. By generating sentences word by word, the decoder decomposes the conditional probability  $p(\mathcal{S}|\mathcal{V})$  into ordered conditional probabilities:

$$p(\mathcal{S}|\mathcal{V}) = \prod_{u=1}^U p(w_u | w_{0:u-1}, h_{sign}^e) \quad (5.7)$$

which are then used to calculate the errors by applying Cross Entropy Loss [52] for each word. For end-to-end experiments, these errors are back propagated through the encoder-decoder network to the CNN and word embeddings, thus updating all of the network parameters.

### Encoder and Decoder Recurrent Units

Various types of recurrent units have been used for neural machine translation. The first encoder-decoder network proposed by Kalchbrenner and Blunsom [91], which utilized RNN decoders with colloquial recurrent units. Later approaches employed shallow [172, 113] and deep architectures [199] of Long Short-Term Memory (LSTM) units [82] and Gated Recurrent Units (GRUs) [38].

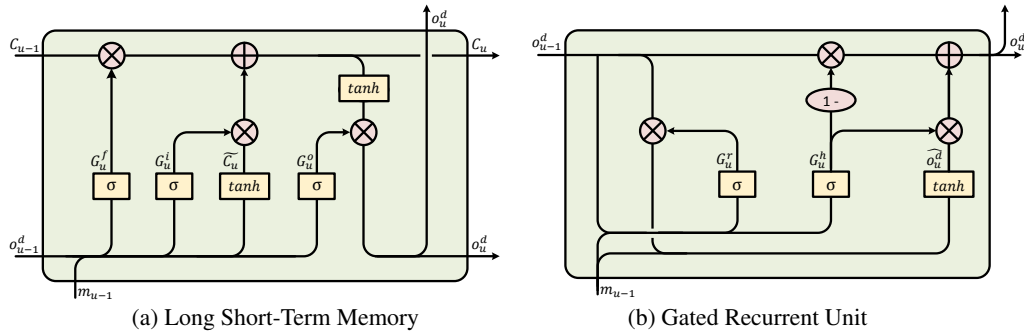


Figure 5.2: Visualizations of Long Short-Term Memory [82] unit and Gated Recurrent Unit [38].

LSTM units and GRUs were proposed to address the vanishing and exploding gradient issues of colloquial RNNs. To achieve this, both units utilize gating mechanisms to update hidden states of the recurrent cells, thus maintaining long term information flow by avoiding nonlinear operations [126].

LSTM units, visualized in Figure 5.2a, are built using three gates, namely the forget gate,  $G_u^f$ , input gate,  $G_u^i$ , and output gate,  $G_u^o$ . They also utilize cell state,  $C_u$ , which is responsible for storing long-term dependencies and is updated only by using linear operations.

Given an input,  $m_{u-1}$ , and the previous cell output,  $o_{u-1}^d$ , the LSTM first decides which information it is going to eliminate from the previous cell state,  $C_{u-1}$ , by using the forget gate,  $G_u^f$ , which is calculated as:

$$G_u^f = \sigma(W_f[o_{u-1}^d, m_{u-1}] + b_f) \quad (5.8)$$

After calculating the forget gate, the next operation is to update the cell state,  $C_u$ . The first step is to calculate a new candidate cell state,  $\tilde{C}_u$ , and the input gate,  $G_u^i$ , as:

$$\tilde{C}_u = \tanh(W_c[o_{u-1}^d, m_{u-1}] + b_c) \quad (5.9)$$

$$G_u^i = \sigma(W_i[o_{u-1}^d, m_{u-1}] + b_i) \quad (5.10)$$

Then, the new cell state  $C_u$  is calculated using the forget gate,  $G_u^f$ , input gate,  $G_u^i$ , and the candidate cell state  $\tilde{C}_u$  as:

$$C_u = G_u^f .* C_{u-1} + G_u^i .* \tilde{C}_u \quad (5.11)$$

where  $(.*)$  is an element-wise multiplication. Here the forget gate,  $G_u^f$ , acts as a soft forget mask over the information flow of the cell state while the input gate,  $G_u^i$ , regulates the amount of new information we are going to add to the cell state,  $C_u$ , from the candidate cell state  $\tilde{C}_u$ .

The last operation in an LSTM cell is to generate an output based on the new cell state  $C_u$ . This is done by utilizing the output gate,  $G_u^o$ , which is calculated as:

$$G_u^o = \sigma(W_o[o_{u-1}^d, m_{u-1}] + b_o) \quad (5.12)$$

Finally the output gate,  $G_u^o$ , is applied to the cell state,  $C_u$ , to calculate the new output  $o_u^d$  as:

$$o_u^d = G_u^o \cdot \tanh(C_u) \quad (5.13)$$

In contrast to the LSTM units, GRUs (which are visualized in Figure 5.2b) combine the forget and input gates into a single update gate,  $G_u^r$ , and use a modified output gate  $G_u^h$ . They also don't utilize the cell state thus the only source of information between cells are the cell outputs. This makes GRUs faster to train and less prone to over-fitting due to the reduced number of parameters, while achieving competitive performance against LSTM units. The modified equations of the GRUs are as follows:

$$G_u^r = \sigma(W_r[o_{u-1}^d, m_{u-1}] + b_r) \quad (5.14)$$

$$G_u^h = \sigma(W_h[o_{u-1}^d, m_{u-1}] + b_h) \quad (5.15)$$

$$\hat{o}_u^d = \tanh(W[o_{u-1}^d, m_{u-1}] + b_c) \quad (5.16)$$

$$o_u^d = (1 - G_u^h) \cdot o_{u-1}^d + G_u^h \cdot \hat{o}_u^d \quad (5.17)$$

### Attention Mechanisms

A major drawback of using a classic recurrent encoder-decoder architecture is the information bottleneck caused by representing a whole sign language video with a fixed sized vector. Furthermore, due to the large number of frames, our networks suffer from long term dependencies and vanishing gradients. To overcome these issues, we utilize attention mechanisms to provide additional information to the decoding phase. By using attention mechanisms, our networks are able to learn where to focus while generating



each word. We employ two of the most prominent attention approaches proposed by Bahdanau *et al.* [11] and Luong *et al.* [113], which are visualized in Figure 5.3 and were state-of-the-art at the time of this research.

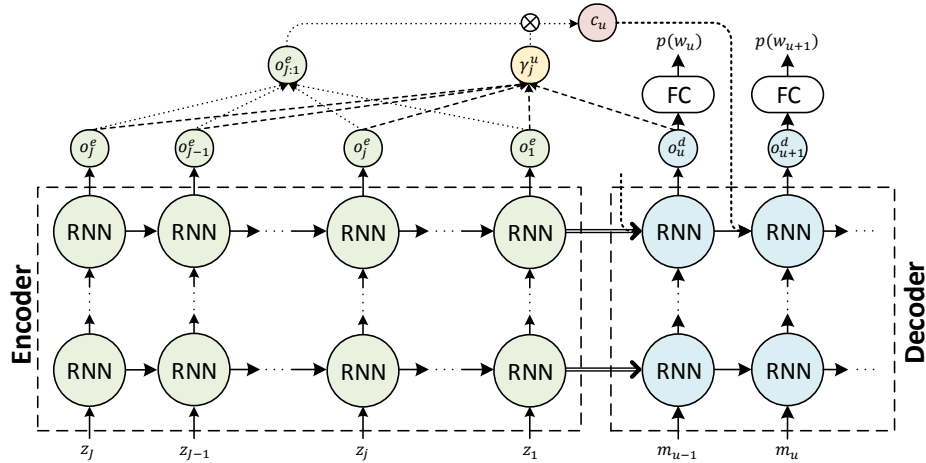
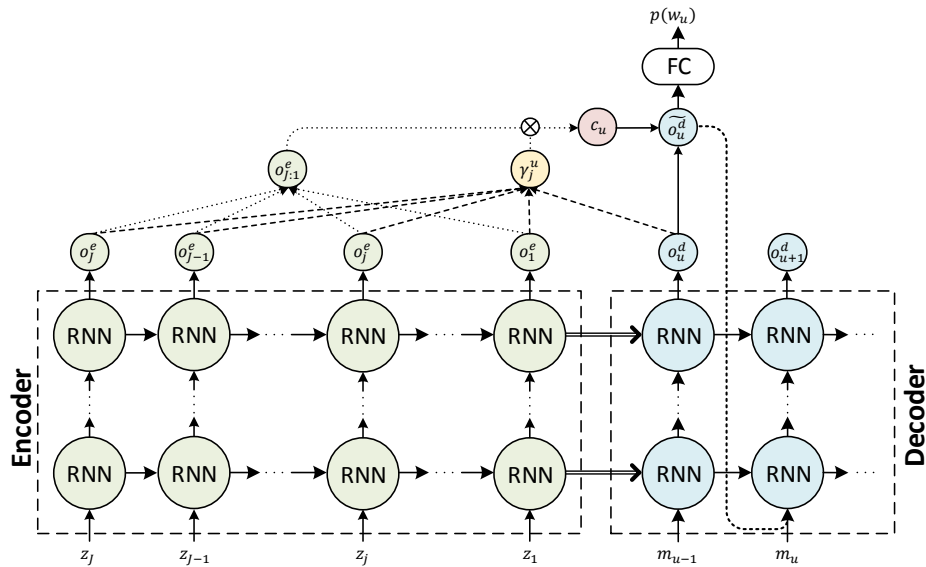
(a) Bahdanau *et al.* [11](b) Luong *et al.* [113]

Figure 5.3: Visualizations of the attention mechanisms used in this chapter.

The idea behind attention mechanisms is to create a weighted summary of the source sequence to aid the decoding phase. This summary is commonly known as the context vector and it will be notated as  $c_u$  in this chapter. For each decoding step  $u$ , a new context vector  $c_u$  is calculated by taking a weighted sum of encoder outputs  $o_{j:1}^e$  as:

$$c_u = \sum_{j=1}^J \gamma_j^u o_j^e \quad (5.18)$$

where  $\gamma_j^u$  represent the attention weights, which can be interpreted as the relevance of an encoder input  $z_j$  to generating the word  $w_u$ . These weights are calculated via a scoring function of the decoder output  $o_u^d$  against each encoder output  $o_j^e$  as:

$$\gamma_j^u = \frac{\exp(\text{score}(o_u^d, o_j^e))}{\sum_{j'=1}^J \exp(\text{score}(o_u^d, o_{j'}^e))} \quad (5.19)$$

where the scoring function depends on the attention mechanism that is being used. In this chapter we examine two scoring functions. The first one is a multiplication based approach proposed by Luong *et al.* [113], which is commonly known as the dot-product attention, and the second is a concatenation based function proposed by Bahdanau *et al.* [11]. These functions are as follows:

$$\text{score}(o_u^d, o_j^e) = \begin{cases} (o_u^d)^\top W o_j^e & \text{[Multiplication]} \\ V^\top \tanh(W[o_u^d; o_j^e]) & \text{[Concatenation]} \end{cases} \quad (5.20)$$

where  $W$  and  $V$  are learned parameters. The context vector  $c_u$  is then combined with the decoder output  $o_u^d$  to calculate an updated decoder output as:

$$\tilde{o}_u^d = \tanh(W_c[c_u; o_u^d]) \quad (5.21)$$

Finally, we feed the updated decoder output,  $\tilde{o}_u^d$ , to a FC layer, replacing the old decoder output  $o_u^d$  in Equation 5.6, to model the ordered conditional probability in Equation 5.7. Furthermore, following Luong *et al.*'s architecture [113],  $\tilde{o}_u^d$  is fed to the next decoding step  $u + 1$  thus changing Equation 5.5 to:

$$o_{u+1}^d, h_{u+1}^d = \text{Decoder}(m_u, h_u^d, \tilde{o}_u^d) \quad (5.22)$$

Besides the use of different score functions noted in Equation 5.20, there are several architectural differences between attention mechanisms proposed by Bahdanau *et al.*

(See Figure 5.3a) and Luong *et al.* (See Figure 5.3b). While Luong *et al.* use the output of the current step,  $o_u^d$  to create the context vector,  $c_u$  for the word  $w_u$ , Bahdanau *et al.* use the output of the previous step  $o_{u-1}^d$ .

Another difference is how the context vector is incorporated into the decoder network. Luong *et al.* concatenates the decoder output  $o_u^d$  with the context vector  $c_u$  to generate an updated output  $\tilde{o}_u^d$ , as noted in Equation 5.21. Bahdanau *et al.* on the other hand concatenate the context vector  $c_u$  with the decoder networks top layer’s hidden state from the previous step  $h_u^d$  and feed it to the same units in next time step.

The final distinction of Luong *et al.*’s architecture is that they feed the updated output  $\tilde{o}_u^d$  to the next step of the decoder by concatenating it to the word embedding  $m_u$ .

## 5.2 Quantitative Experiments

We evaluate the proposed Neural Sign Language Translation approach on the recently released PHOENIX14T (See Chapter 4) dataset. To underpin future research and to set baselines for SLT, we conduct several sets of experiments using some of the protocols laid down in Section 4.1. We categorize our experiments under three groups:

1. Gloss2Text, in which we simulate having a perfect SLR system as an intermediate tokenization.
2. Sign2Text which covers the end-to-end pipeline, translating directly from frame level sign language video into spoken language.
3. Sign2Gloss2Text which uses an SLR system [102] as tokenization layer to add intermediate supervision.

All of our encoder-decoder networks were built using four stacked layers of *residual* recurrent units with separate parameters. Each recurrent layer contains 1000 hidden units, thus yielding 8000 units in total for each encoder-decoder network. In our Sign2Text experiments we use AlexNet [105] without its final FC layer (FC8) as our Spatial Embedding layer and initialize it using weights that were trained on ImageNet [53].

For our Sign2Gloss2Text experiments we use the CNN-RNN-HMM network proposed by Koller *et al.* [102] as our spatial embedding and tokenization layers, which was the state-of-the-art CSLR model at the time of this research. It achieves a gloss recognition performance of 25.7% and 26.6% Word Error Rate (WER) on the development and test sets of the PHOENIX14T, respectively. All remaining parts of our networks are initialized using Xavier [69] initialization. We use Adam [94] optimization with a learning rate of  $10^{-5}$  and its default parameters ( $\beta_1 = 0.9$ ,  $\beta_2 = 0.999$ ,  $\epsilon = 10^{-8}$ ). We also use gradient clipping with a threshold of five and dropout connections with a drop probability of 0.2.

All of our networks are trained until the training perplexity has converged, which took  $\sim 30$  epochs on average. We evaluate our models on the development and test sets every half-epoch, and report results for each setup using the model that performed the best on the development set. In the decoding phase we generate spoken language sentences using a beam search with a beam width of three, which we empirically show to be the optimal beam size for our experiments.

To measure our translation performance we utilize Bilingual Evaluation Understudy (BLEU) [132] and Recall-Oriented Understudy for Gisting Evaluation (ROUGE) [110] scores, which are described in detail in Section 4.2.

### 5.2.1 Gloss2Text: Simulating Perfect Recognition

In our first set of experiments we simulate having an idealized SLR system which performs perfect SLR as an intermediate tokenizer. To do so, NMT networks are trained to generate spoken language translations from ground truth sign glosses. We refer to this set of experiments as Gloss2Text.

There are two main objectives of the Gloss2Text experiments. The first objective is to create an artificial upper bound for end-to-end SLT. This upper bound is considered as artificial, due to the fact that sign gloss annotation is prone to human error and exactly how sign glosses should be annotated is still an open linguistic research question. We believe that given enough training data and more advanced network architectures, it is possible for an end-to-end SLT system to learn more informative intermediate

representations than sign glosses, which capture sub-unit level representations of the signs. However, until we have more publicly available sign language resources and more advanced machine translation methods, this artificial upper bound should serve the SLT research community as a good comparison point for evaluating automatic sign video to spoken language translation systems.

The second objective of the Gloss2Text experiments is to examine different encoder-decoder network architectures and hyper-parameters, and evaluate their effects on sign to spoken language translation performance. As training Sign2Text networks is an order of magnitude slower than Gloss2Text (10 days vs. one day, respectively), we use the best performing setup from our Gloss2Text experiments while training our Sign2Text networks.

Note that we should expect the translation performance’s upper bound to be significantly lower than 100%. As in all natural language problems, there are many ways to say the same thing, and thus many equally valid translations. Unfortunately, this is impossible to quantify perfectly using any existing automatic evaluation measure, given that our dataset only provides a single reference translation.

### Recurrent Units: GRU vs LSTM

To choose which recurrent unit to use, we trained two Gloss2Text networks using LSTM units and GRUs. Both networks were trained using a batch size of 128 and the Luong attention mechanism as described in Section 5.1.

Unit Type:	DEV SET					TEST SET				
	ROUGE	BLEU-1	BLEU-2	BLEU-3	BLEU-4	ROUGE	BLEU-1	BLEU-2	BLEU-3	BLEU-4
LSTM	41.69	41.54	27.90	20.66	16.40	41.92	41.22	28.03	20.77	16.58
<b>GRU</b>	<b>43.85</b>	<b>43.71</b>	<b>30.49</b>	<b>23.15</b>	<b>18.78</b>	<b>43.73</b>	<b>43.43</b>	<b>30.73</b>	<b>23.36</b>	<b>18.75</b>

Table 5.1: Gloss2Text: Effects of using different recurrent units on translation performance.

As can be seen in Table 5.1, GRUs outperformed LSTM units in both BLEU and ROUGE scores. We believe this is due to over-fitting caused by the additional parameters in LSTM units and the limited number of training sequences. As discussed in detail

in Section 5.1.3, compared to LSTM units, GRUs operate using fewer gates (two vs. three) and hence have fewer parameters (three vs four weights per unit) which makes them faster to train and less prone to over-fitting with smaller datasets. We therefore use GRUs for the rest of our experiments.

### Attention Mechanisms: Luong vs. Bahdanau

Next we evaluated the effects of different attention mechanisms for the Gloss2Text translation task. We used Bahdanau [11] and Luong [113] attention, which was described in detail in Section 5.1. We also trained a network which did not use any attention mechanisms as a baseline. All of our networks were trained using GRUs and a batch size of 128.

Attention:	DEV SET					TEST SET				
	ROUGE	BLEU-1	BLEU-2	BLEU-3	BLEU-4	ROUGE	BLEU-1	BLEU-2	BLEU-3	BLEU-4
None	40.32	40.45	27.19	20.28	16.29	40.71	40.66	27.48	20.40	16.34
Bahdanau	42.93	42.93	29.71	22.43	17.99	42.61	42.76	29.55	22.00	17.40
<b>Luong</b>	<b>43.85</b>	<b>43.71</b>	<b>30.49</b>	<b>23.15</b>	<b>18.78</b>	<b>43.73</b>	<b>43.43</b>	<b>30.73</b>	<b>23.36</b>	<b>18.75</b>

Table 5.2: Gloss2Text: Attention Mechanism Experiments.

Our first observation from this experiment was that having an attention mechanism improved the translation performance drastically as shown in Table 5.2. When attention mechanisms are compared, Luong attention outperformed Bahdanau attention and generalized better on the test set.

We believe this is due to the architectural differences of how attention is implemented by Luong *et al.* and Bahdanau *et al.*, described in detail in Section 5.1.3. Luong *et al.* injects the context vector twice into the network, first in combination with the word embedding to the decoder and the second time in combination with the decoder output. Compared to Bahdanau *et al.*'s single injection from the top recurrent layer, Luong *et al.*'s approach allows further modelling of the context vector, which we believe is the reason for the performance difference. Therefore we train our remaining Gloss2Text networks using Luong attention.

### What Batch Size to use?

There have been several studies on the effects of batch sizes while using Stochastic Gradient Descent (SGD) [16, 109]. Although larger batch sizes have the advantage of providing smoother gradients, they decrease the rate of convergence. Furthermore, recent studies on the information theory behind Deep Learning (DL) suggests the noise provided by smaller batch sizes helps the networks to represent the data more efficiently [180, 158].

In addition, training and evaluation set distributions of sequence-to-sequence datasets are distinct by nature. When early stopping is employed during training, having additional noise provided by smaller batch sizes gives the optimization the opportunity to step closer to the target distribution. This suggests there is an optimal batch size given a network setup. Therefore, in our third set of experiments we evaluate the effects of the batch size on translation. We train five Gloss2Text networks using different batch sizes that are 128, 64, 32, 16 and 1. All of our networks were trained using GRUs and Luong attention.

Batch Size:	DEV SET					TEST SET				
	ROUGE	BLEU-1	BLEU-2	BLEU-3	BLEU-4	ROUGE	BLEU-1	BLEU-2	BLEU-3	BLEU-4
128	43.85	43.71	30.49	23.15	18.78	43.73	43.43	30.73	23.36	18.75
64	43.78	43.52	30.56	23.36	18.95	44.36	44.33	31.34	23.74	19.06
32	44.63	<b>44.67</b>	31.44	24.08	19.58	44.52	<b>44.51</b>	31.29	23.76	19.14
16	44.87	44.10	31.16	23.89	19.52	44.37	43.96	31.11	23.66	19.01
<b>1</b>	<b>46.02</b>	44.40	<b>31.83</b>	<b>24.61</b>	<b>20.16</b>	<b>45.45</b>	44.13	<b>31.47</b>	<b>23.89</b>	<b>19.26</b>

Table 5.3: Gloss2Text: Batch Size Experiments.

One interesting observation from this experiment was that, the networks trained using smaller batch sizes converged faster but to a higher training perplexity than one. We believe this is due to high variance between gradients. To deal with this we decrease the learning rate to  $10^{-6}$  when the training perplexity plateaus, and continue training for 100,000 iterations. Results show that having a smaller batch size helps the translation performance. As reported in Table 5.3, the Gloss2Text network with batch size one outperformed networks that were trained using larger batch sizes. Considering these results, the remainder of our experiments use a batch size of one.

### Effects of Beam Width

The most straight forward decoding approach for an Encoder-Decoder networks is to use a greedy search, in which the word with highest probability is considered the prediction and fed to the next time step of the decoder. However, this greedy approach is prone to errors, given that the predictions can have a low confidence. To address this, we use a simple left-to-right Beam Search during the decoding phase, in which a number of candidate sequences, also known as beam width, are stored and propagated through the decoder. However, larger beam widths do not necessarily mean better translation performance and increases decoding duration and memory requirements. Therefore, to find the optimal value, we use our best performing Gloss2Text network to perform a parameter search over possible beam widths and report development and test set translation performances in the form of a BLEU-4 score.

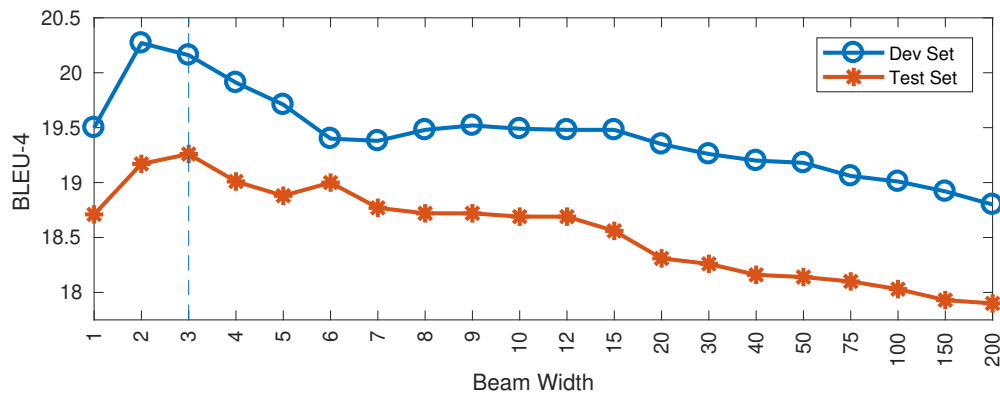


Figure 5.4: Effects of Beam Width on Gloss2Text performance.

As shown in Figure 5.4, a beam width of two or three was optimal for our Gloss2Text network. Although a beam width of two yielded the highest translation performance on the development set, beam width three generalized better to the test set. In addition, as beam width increased, the BLEU-4 scores plateau and then start to decline. Taking these results into consideration, we continue using a beam width of three for the rest of our experiments.



### 5.2.2 Sign2Text: From Sign Video To Spoken Text

In our second set of experiment we evaluate our Sign2Text networks which learn to generate spoken language from sign videos without any intermediate representation in an end-to-end manner. In this setup our tokenization layer is an Identity function, feeding the spatial embeddings directly to the encoder-decoder network. Using the hyper-parameters from our Gloss2Text experiments, we train three Sign2Text networks with different attention choices.

Attention:	DEV SET					TEST SET				
	ROUGE	BLEU-1	BLEU-2	BLEU-3	BLEU-4	ROUGE	BLEU-1	BLEU-2	BLEU-3	BLEU-4
Gloss2Text (Best)	46.02	44.40	31.83	24.61	20.16	45.45	44.13	31.47	23.89	19.26
None	31.00	28.10	16.81	11.82	9.12	29.70	27.10	15.61	10.82	8.35
Bahdanau	31.80	<b>31.87</b>	<b>19.11</b>	13.16	9.94	<b>31.80</b>	<b>32.24</b>	<b>19.03</b>	<b>12.83</b>	<b>9.58</b>
Luong	<b>32.6</b>	31.58	18.98	<b>13.22</b>	<b>10.00</b>	30.70	29.86	17.52	11.96	9.00

Table 5.4: Sign2Text: Attention Mechanism Experiments.

As with the Gloss2Text task, utilizing attention mechanisms increases the translation performance of our Sign2Text networks (See Table 5.4). However, when compared against Gloss2Text, the translation performance is lower. We believe this might be due to several reasons. As the number of frames in a sign video is much higher than the number of its gloss level representations, our Sign2Text networks suffer from long term dependencies and vanishing gradients. In addition, the dataset we are using might be too small to allow our Sign2Text network to generalize considering the number of parameters (CNN + Encoder Decoder + Attention). Furthermore, expecting our networks to recognize visual sign language and translate them to spoken language with only the translation supervision might be too much to ask from them. Therefore, in our next set of experiments, which we call Sign2Gloss2Text, we introduce the gloss level supervision to aid the task of full translation.

### 5.2.3 Sign2Gloss2Text: Gloss Level Supervision

In our final experiment we propose using glosses as an intermediate representation while going from sign videos to spoken language translations. To achieve this, we use the

CNN-RNN-HMM hybrid proposed in [102] as our spatial embedding and tokenization layers. We evaluate two setups. In the first setup: Sign2Gloss→Gloss2Text, we use our best performing Gloss2Text network without any retraining to generate sentences from the estimated gloss token embeddings. In the second setup: Sign2Gloss2Text, we train a network from scratch to learn to translate from the predicted gloss.

Approach	DEV SET					TEST SET				
	ROUGE	BLEU-1	BLEU-2	BLEU-3	BLEU-4	ROUGE	BLEU-1	BLEU-2	BLEU-3	BLEU-4
Gloss2Text	46.02	44.40	31.83	24.61	20.16	45.45	44.13	31.47	23.89	19.26
Sign2Text	31.80	31.87	19.11	13.16	9.94	31.80	32.24	19.03	12.83	9.58
Sign2Gloss→Gloss2Text	43.76	41.08	29.10	22.16	17.86	43.45	41.54	29.52	22.24	17.79
<b>Sign2Gloss2Text</b>	<b>44.14</b>	<b>42.88</b>	<b>30.30</b>	<b>23.02</b>	<b>18.40</b>	<b>43.80</b>	<b>43.29</b>	<b>30.39</b>	<b>22.82</b>	<b>18.13</b>

Table 5.5: Effects of different tokenization schemes for sign to text translation.

The Sign2Gloss→Gloss2Text network performs surprisingly well considering there was no additional training. This shows us that our Gloss2Text network has already learned some robustness to noisy inputs, despite being trained on perfect glosses, this may be due to the dropout regularization employed during training. However, our second approach Sign2Gloss2Text surpasses these results and obtains scores close to the idealized performance of the Gloss2Text network. This is likely because the translation system is able to correct the failure modes in the tokenizer. As can be seen in Table 5.5, compared to the Sign2Text network Sign2Gloss2Text was able to surpass its performance by a large margin, indicating the importance of intermediary expert gloss level supervision to simplify the training process.

### 5.3 Qualitative Examples

One of the most obvious ways of qualifying translation is to examine the resultant translations. To give a better understanding to the reader, in Table 5.6 we share translation samples generated from our Gloss2Text, Sign2Text and Sign2Gloss2Text networks accompanied by the ground truth German and word to word English translations.

We can see that the most common error made is the mistranslation of dates, places and numbers. Although this does not effect the overall structure of the translated sentence, it tells us the embedding learned for these infrequent words is insufficient.

---

## 5.4 Closing Remarks

In this chapter, we introduced the challenging task of Sign Language Translation and proposed the first end-to-end solution. In contrast to previous research, we took a machine translation perspective; treating sign language as a fully independent language and proposing SLT rather than SLR. To achieve NMT from sign videos, we employed CNN based spatial embedding, various tokenization methods including state-of-the-art RNN-HMM hybrids [102] and attention-based encoder-decoder networks, to jointly learn to align, recognize and translate sign videos to spoken text.

To evaluate our approach we use the recently released PHOENIX14T, the first continuous sign language translation dataset, which is publicly available. We conducted extensive experiments, making a number of recommendations to underpin future research.

One of the limiting factors in our experiments was the Graphics Processing Unit (GPU) technology at the time. Due to the length of sign sequences, which can go up to 300 frames in PHOENIX14T, we were only able to fit one sequence at a time on a NVIDIA Titan X Maxwell GPU with 12GB video memory. This limitation forced us to employ smaller batch sizes and legacy CNNs, namely AlexNet [105]. Although the increased intra-iteration noise caused by having a smaller batch size acts as a regularizer, it is not optimal for training large networks as it is computationally inefficient, i.e. training can take more than 10 days. One way to address this issue is to pretrain CNNs on similar sign language tasks and use the features extracted by them as the input of our SLT networks. This approach will not only drastically reduce the training times but will also enable us to exploit more sophisticated CNN architectures, which would not have fitted in the graphics card memory while training.

Another way to increase the batch size and to enable the use of more advanced CNN architectures, such as ResNets [79], is to use key frame extraction methods on the sign videos before feeding the frames to the networks. In sign sequences there are many redundant and blurry frames which carry little information besides indicating the speed of the performed sign. We can eliminate these frames by localizing stationary poses of the signers. In addition, we can employ motion descriptors such as Optical Flow

[83] to address the information loss caused by frame elimination. We believe such an approach would drastically reduce the number of frames and improve the translation performance as it would inherently shorten the long term dependencies within videos.

One of the key findings of our experiments was that using gloss information as an intermediate step to spoken language translation improved our performance drastically. We believe there are three reasons for this. Firstly, modelling gloss representations are much easier than modelling spatial embeddings, as the gloss vocabulary is limited while there are infinitely many possible input images. Secondly, gloss level representations are user independent, as the translation system does not need to handle different identities embedded in the spatial representations. Thirdly, gloss level representations are much more concise. By which we mean we can represent the same sign sequence with fewer tokens using glosses rather than frames, *e.g.* it is possible to represent a five second video with 125 frames with less than 10 glosses, which makes the job of the translation system significantly easier. However, utilizing this two step approach imposes an information bottleneck, as glosses are imprecise text-based representations of visual sign languages. Therefore, advancements are required to incorporate sign gloss supervision, while not limiting the translation performance.

To address the limitations of the work presented in this chapter, namely poor performance of Sign2Text models and the information bottleneck imposed by using text-based intermediate gloss representations, we propose a multi-task formulation of the SLT task in Chapter 6. We utilize pretrained CNNs to extract spatial representations of video frames. We then use gloss supervision as an auxiliary loss function and train our joint recognition and translation models in an end-to-end manner.





 <p>Ground Truth: und nun die wettervorhersage für morgen samstag den zweiten april . ( and now the weatherforecast for tomorrow saturday the second april . )</p> <p>Gloss2Text: und nun die wettervorhersage für morgen samstag den elften april . ( and now the weatherforecast for tomorrow saturday the eleventh april . )</p> <p>Sign2Text: und nun die wettervorhersage für morgen freitag den sechszwanzigsten märz . ( and now the weatherforecast for tomorrow friday the twentiesixth march . )</p> <p>Sign2Gloss2Text: und nun die wettervorhersage für morgen samstag den siebzehnten april . ( and now the weatherforecast for tomorrow saturday the seventeenth april . )</p>
 <p>Ground Truth: die neue woche beginnt wechselhaft und kühler . ( the new week starts unpredictable and cooler . )</p> <p>Gloss2Text: die neue woche beginnt wechselhaft und wieder kühler . ( the new week starts unpredictable and again cooler . )</p> <p>Sign2Text: am montag überall wechselhaft und kühler . ( on monday everywhere unpredictable and cooler . )</p> <p>Sign2Gloss2Text: die neue woche beginnt wechselhaft und wechselhaft . ( the new week starts unpredictable and unpredictable . )</p>
 <p>Ground Truth: im süden und südwesten gebietsweise regen sonst recht freundlich . ( in the south and southwest locally rain otherwise quite friendly . )</p> <p>Gloss2Text: in der südwesthälfte regnet es zeitweise sonst ist es recht freundlich . ( in the southwestpart it rains temporarily otherwise it is quite friendly . )</p> <p>Sign2Text: von der südhälfte beginnt es vielerorts . ( from the southpart it starts in many places . )</p> <p>Sign2Gloss2Text: am freundlichsten wird es im süden . ( the friendliest it will be in the south . )</p>
 <p>Ground Truth: am sonntag ab und an regenschauer teilweise auch gewitter . ( on sunday time to time rainshower partly also thunderstorm . )</p> <p>Gloss2Text: am sonntag teilweise kräftige schauer und gewitter . ( on sunday partly heavy shower and thunderstorm . )</p> <p>Sign2Text: am sonntag sonne und wolken und gewitter . ( on sunday sun and clouds and thunderstorm . )</p> <p>Sign2Gloss2Text: am sonntag breiten sich teilweise kräftige schauer und gewitter . ( on sunday spreads partly heavy shower and thunderstorm . )</p>

Table 5.6: Samples where models failed to correctly translate the target sentence.



## Chapter 6

# Sign Language Transformers: Joint End-to-end Sign Language Recognition and Translation

One of the main findings in Chapter 5 was that using gloss based intermediate representations improved the Sign Language Translation (SLT) performance drastically when compared to an end-to-end *Sign2Text* approach. The resulting *Sign2Gloss2Text* model first recognized glosses from continuous sign videos using a state-of-the-art Continuous Sign Language Recognition (CSLR) method [102], which worked as a tokenization layer. The recognized sign glosses were then passed to a text-to-text attention-based Neural Machine Translation (NMT) network [113] to generate spoken language sentences.

We hypothesize that there are two main reasons why *Sign2Gloss2Text* performs better than *Sign2Text* (18.13 vs 9.58 BLEU-4 scores). Firstly, the number of sign glosses is much lower than the number of frames in the videos they represent. By using gloss representations instead of the spatial embeddings extracted from the video frames, *Sign2Gloss2Text* avoids the long-term dependency issues, which *Sign2Text* suffers from. However, this problem can be addressed by reducing the number of frames using techniques proposed in the legacy key-frame based Sign Language Recognition (SLR)

literature [202]. Furthermore, optical-flow based action recognition methods could be utilized [210] to retain the motion information.

We think the second and more critical reason is the lack of direct guidance for understanding sign sentences in *Sign2Text* training. Given the complexity of the task, it might be too difficult for current *Neural Sign Language Translation* architectures to comprehend sign without any explicit intermediate supervision. In this chapter, we propose a novel *Sign Language Transformer* approach, which addresses this issue while avoiding the need for a two-step pipeline, where translation is solely dependent on recognition accuracy. This is achieved by jointly learning sign language recognition and translation from spatial-representations of sign language videos in an end-to-end manner. Exploiting the encoder-decoder based architecture of transformer networks [187], we propose a multi-task formalization of the joint continuous sign language recognition and translation problem.

To help our translation networks with sign language understanding and to achieve CSLR, we introduce a Sign Language Recognition Transformer (SLRT), an encoder transformer model trained using a Connectionist Temporal Classification (CTC) loss [3], to predict sign gloss sequences. SLRT takes spatial embeddings extracted from sign videos and learns spatio-temporal representations. These representations are then fed to the Sign Language Translation Transformer (SLTT), an autoregressive transformer decoder model, which is trained to predict one word at a time to generate the corresponding spoken language sentence.

We evaluate the recognition and translation performances of our approaches on the challenging RWTH-PHOENIX-Weather-2014T Dataset (PHOENIX14T), which is described in detail in Chapter 4. We report state-of-the-art sign language recognition and translation results achieved by our Sign Language Transformers. Our translation networks outperform both sign video to spoken language and gloss to spoken language translation models, in some cases more than doubling the performance (9.58 vs. 21.80 BLEU-4 Score). We also share new baseline translation results using transformer networks for several other text-to-text sign language translation tasks.

The rest of this chapter is organized as follows: In Section 6.1, we present Sign Lan-



guage Transformers, a novel joint sign language recognition and translation approach which can be trained in an end-to-end manner. We introduce our experimental setup and report quantitative results of the Sign Language Transformers in Section 6.2. We then share translation examples generated by our network to give the reader further qualitative insight of how our approach performs in Section 6.3. Finally, we conclude the chapter in Section 6.4 by discussing our findings and possible future work.

## 6.1 Methodology

In this section we introduce Sign Language Transformers which jointly learn to recognize and translate sign video sequences into sign glosses and spoken language sentences in an end-to-end manner. Our objective is to learn the conditional probabilities  $p(\mathcal{G}|\mathcal{V})$  and  $p(\mathcal{S}|\mathcal{V})$  of generating a sign gloss sequence  $\mathcal{G} = (g_1, \dots, g_N)$  with  $N$  glosses and a spoken language sentence  $\mathcal{S} = (w_1, \dots, w_U)$  with  $U$  words given a sign video  $\mathcal{V} = (I_1, \dots, I_T)$  with  $T$  frames.

Modelling these conditional probabilities is a sequence-to-sequence task, and poses several challenges. In both cases, the number of tokens in the source domain is much larger than the corresponding target sequence lengths (*i.e.*  $T \gg N$  and  $T \gg U$ ). Furthermore, the mapping between sign language videos,  $\mathcal{V}$ , and spoken language sentences,  $\mathcal{S}$ , is non-monotonic, as both languages have different vocabularies, grammatical rules and orderings.

Previous sequence-to-sequence based literature on SLT can be categorized into two groups: The first group break down the problem in two stages. They consider CSLR as an initial process and then try to solve the problem as a text-to-text translation task [31, 24]. For example, in Chapter 5, we utilized a CSLR method [102] to obtain sign glosses, and then used an attention-based text-to-text NMT model [113] to learn the sign gloss to spoken language sentence translation,  $p(\mathcal{S}|\mathcal{G})$  [24].

However, in doing so, this approach introduces an information bottleneck in the mid-level gloss representation. This limits the network’s ability to understand sign language as the translation model can only be as good as the sign gloss annotations it was trained

from. There is also an inherent loss of information as a sign gloss is an incomplete representation intended only for linguistic annotation and study, and it therefore neglects many crucial details and information present in the original sign language video.

The second group of methods focus on translation from the sign video representations to spoken language with no intermediate representation [24, 95]. These approaches attempt to learn  $p(\mathcal{S}|\mathcal{V})$  directly. Given enough data and a sufficiently sophisticated network architecture, such models could theoretically realize end-to-end SLT with no need for human-interpretable information that acts as a bottleneck. However, due to the lack of direct supervision guiding sign language understanding, such methods have significantly lower performance than their counterparts on the currently available datasets [24].

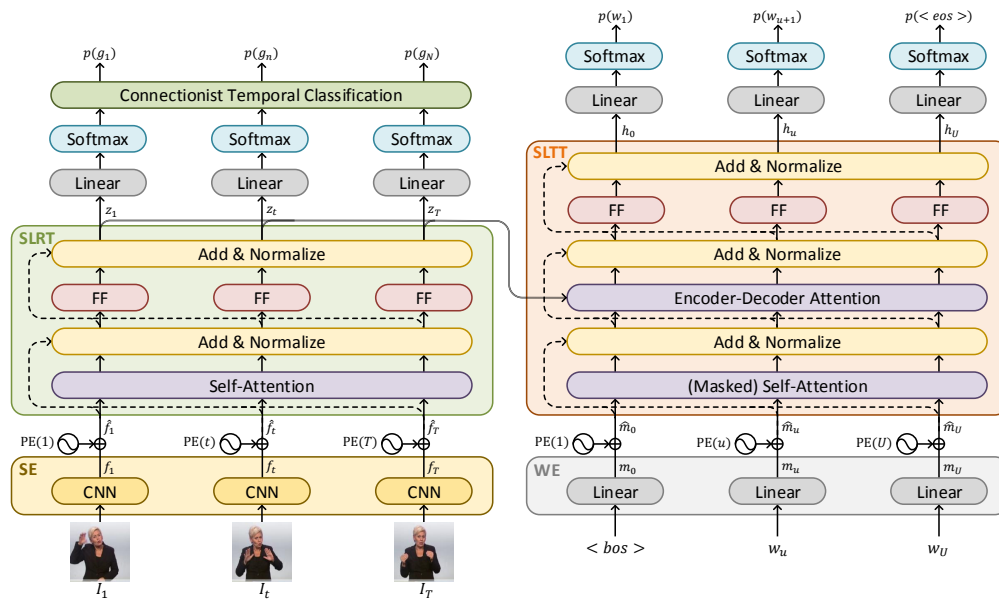


Figure 6.1: A detailed overview of a single layered Sign Language Transformer.

(SE: Spatial Embedding, WE: Word Embedding, PE: Positional Encoding, FF: Feed Forward)

To address this, we propose to jointly learn  $p(\mathcal{G}|\mathcal{V})$  and  $p(\mathcal{S}|\mathcal{V})$ , in an end-to-end manner. We build upon transformer networks [187] to create a unified model, which we call Sign Language Transformers. We train our networks to generate spoken language sentences from sign language video representations. During training, we inject intermediate

gloss supervision in the form of a CTC loss into the Sign Language Recognition Transformer (SLRT) encoder. This helps our networks learn more meaningful spatio-temporal representations of the sign without limiting the information passed to the decoder. We employ an autoregressive Sign Language Translation Transformer (SLTT) decoder which predicts one word at a time to generate the spoken language sentence translation. And overview of our approach can be seen in Figure 6.1.

### 6.1.1 Spatial and Word Embeddings

Following the classic NMT pipeline, we start by embedding our source and target tokens, namely sign language video frames and spoken language words. As word embedding we use a linear layer, which is initialized from scratch during training, to project a one-hot-vector representation of the words into a denser space. To embed video frames, we use the Spatial Embedding approach [24], and propagate each image through Convolutional Neural Networks (CNNs). We formulate these operations as:

$$\begin{aligned} m_u &= \text{WordEmbedding}(w_u) \\ f_t &= \text{SpatialEmbedding}(I_t) \end{aligned} \tag{6.1}$$

where  $m_u$  is the embedded representation of the spoken language word  $w_u$  and  $f_t$  corresponds to the non-linear frame level spatial representation obtained from a CNN.

Unlike other sequence-to-sequence models [172, 68], transformer networks do not employ recurrence or convolutions, thus lacking the positional information within sequences. To address this issue we follow the positional encoding method proposed in [187] and add temporal ordering information to our embedded representations as:

$$\begin{aligned} \hat{f}_t &= f_t + \text{PositionalEncoding}(t) \\ \hat{m}_u &= m_u + \text{PositionalEncoding}(u) \end{aligned} \tag{6.2}$$

where `PositionalEncoding` is a predefined function which produces a unique vector in the form of a phase shifted sine wave for each time step. The used `PositionalEncoding` function can be formalized as:

$$\begin{aligned} \text{PositionalEncoding}(ts, 2i) &= \sin(ts/10000^{2i/d_k}) \\ \text{PositionalEncoding}(ts, 2i + 1) &= \cos(ts/10000^{2i/d_k}) \end{aligned} \quad (6.4)$$

where  $ts$  is the time step that is being positionally encoded,  $d_k$  is the hidden size of the transformer network, and  $i \in \{0, \dots, d_k/2\}$  is used to index the feature vector. Each dimension of the positional encoding vector corresponds to a sinusoid, whose wavelengths form a geometric progression from  $2\pi$  to  $10000 \cdot 2\pi$  [187].

### 6.1.2 Sign Language Recognition Transformers

The aim of SLRT is to recognize glosses from continuous sign language videos while learning meaningful spatio-temporal representations for the end goal of sign language translation. Using the positionally encoded spatial embeddings,  $\hat{f}_{1:T}$ , we train a transformer encoder model [187].

The inputs to SLRT are first modelled by a Self-Attention layer which learns the contextual relationship between the frame representations of a video. Outputs of the self-attention are then passed through a non-linear point-wise feed forward layer. All the operations are followed by residual connections and normalization to help training. We formulate this encoding process as:

$$z_t = \text{SLRT}(\hat{f}_t | \hat{f}_{1:T}) \quad (6.5)$$

where  $z_t$  denotes the spatio-temporal representation of the frame  $I_t$ , which is generated by SLRT at time step  $t$ , given the spatial representations of all of the video frames,  $\hat{f}_{1:T}$ .

We inject intermediate supervision to help our networks understand sign language and to guide them to learn meaningful representations which would help the main task of translation. We train the SLRT to model  $p(\mathcal{G}|\mathcal{V})$  and predict sign glosses.

Due to the spatio-temporal nature of the signs, glosses have a one-to-many mapping to video frames but share the same ordering. One way to train the SLRT would be using a cross-entropy loss [70] with frame level annotations. However, sign gloss

annotations with such precision are rare. An alternative form of weaker supervision is to use a sequence-to-sequence learning loss functions, such as CTC [72]. First proposed by Graves *et al.* for phoneme recognition from speech [72], CTC has achieved state-of-the-art continuous recognition performance in a range of fields including audio-visual speech recognition [3], lip reading [10], hand shape recognition [23] and CSLR [211, 142].

Given spatio-temporal representations,  $z_{1:T}$ , we obtain frame level gloss probabilities,  $p(g_t|\mathcal{V})$ , using a linear projection layer followed by a softmax activation. We then use CTC to compute  $p(\mathcal{G}|\mathcal{V})$  by marginalizing over all possible  $\mathcal{V}$  to  $\mathcal{G}$  alignments as:

$$p(\mathcal{G}|\mathcal{V}) = \sum_{\pi \in \mathcal{B}} p(\pi|\mathcal{V}) \quad (6.6)$$

where  $\pi$  is a path and  $\mathcal{B}$  are the set of all viable paths that correspond to  $\mathcal{G}$ . We then use the  $p(\mathcal{G}|\mathcal{V})$  to calculate the CSLR loss as:

$$\mathcal{L}_R = 1 - p(\mathcal{G}^*|\mathcal{V}) \quad (6.7)$$

where  $\mathcal{G}^*$  is the ground truth gloss sequence.

### 6.1.3 Sign Language Translation Transformers

The end goal of our approach is to generate spoken language sentences from sign video representations. We propose training an autoregressive transformer decoder model, named SLTT, which exploits the spatio-temporal representations learned by the SLRT.

We start by prefixing the target spoken language sentence  $\mathcal{S}$  with the special beginning of sentence token,  $\langle \text{bos} \rangle$ . We then extract the positionally encoded word embeddings. These embeddings are passed to a masked self-attention layer. Although the main idea behind self-attention is the same as in SLRT, the SLTT utilizes a mask over the self-attention layer inputs. This ensures that each token may only use its predecessors while extracting contextual information. This masking operation is necessary, as at inference

time the SLTT won't have access to the output tokens which would follow the token currently being decoded.

Representations extracted from both SLRT and SLTT self-attention layers are combined and given to an encoder-decoder attention module which learns the mapping between source and target sequences. Outputs of the encoder-decoder attention are then passed through a non-linear point-wise feed forward layer. Similar to SLRT, all the operations are followed by residual connections and normalization. We formulate this decoding process as:

$$h_{u+1} = \text{SLTT}(\hat{m}_u | \hat{m}_{1:u-1}, z_{1:T}). \quad (6.8)$$

SLTT learns to generate one word at a time until it produces the special end of sentence token,  $\langle \text{eos} \rangle$ . It is trained by decomposing the sequence level conditional probability  $p(\mathcal{S}|\mathcal{V})$  into ordered conditional probabilities

$$p(\mathcal{S}|\mathcal{V}) = \prod_{u=1}^U p(w_u | h_u) \quad (6.9)$$

which are used to calculate the cross-entropy loss for each word as:

$$\mathcal{L}_T = 1 - \prod_{u=1}^U \sum_{d=1}^D p(\hat{w}_u^d) p(w_u^d | h_u) \quad (6.10)$$

where  $p(\hat{w}_u^d)$  represents the ground truth probability of word  $w^d$  at decoding step  $u$  and  $D$  is the target language vocabulary size.

We train our networks by minimizing the joint loss term  $\mathcal{L}$ , which is a weighted sum of the recognition loss  $\mathcal{L}_R$  and the translation loss  $\mathcal{L}_T$  as:

$$\mathcal{L} = \lambda_R \mathcal{L}_R + \lambda_T \mathcal{L}_T \quad (6.11)$$

where  $\lambda_R$  and  $\lambda_T$  are hyper parameters which decide the importance of each loss function during training and are evaluated in Section 6.2.

---

## 6.2 Quantitative Experiments

In this section we share our sign language recognition and translation experimental setups and report quantitative results on PHOENIX14T, which is described in detail in Chapter 4. We first go over the implementation details and provide a reminder of the evaluation metrics used to measure the performance.

We start our experiments by applying transformer networks to the text-to-text based SLT tasks, namely *Gloss2Text*, *Sign2Gloss2Text*, *Sign2Gloss*→*Gloss2Text* and report improved performance over using Recurrent Neural Network (RNN) based models. We share our *Sign2Gloss* experiments, in which we explore the effects of different types of spatial embeddings and network structures on the performance of CSLR. We then train *Sign2Text* and *Sign2(Gloss+Text)* models using the best performing *Sign2Gloss* configuration and investigate the effect of different recognition loss weights on the joint recognition and translation performance. Finally, we compare our best performing models against other approaches and report state-of-the-art results.

### 6.2.1 Implementation and Evaluation Details

**Framework:** We used a modified version of JoeyNMT [104] to implement our Sign Language Transformers<sup>1</sup>. All components of our network were built using the PyTorch framework [134], except the CTC beam search decoding, for which we utilized the TensorFlow implementation [1].

**Network Details:** Our transformers are built using 512 hidden units and 8 heads in each layer. We use Xavier initialization [69] and train all of our networks from scratch. We also utilize dropout with 0.1 drop rate on transformer layers and word embeddings to mitigate over-fitting.

**Performance Metrics:** We employed the metrics introduced in Section 4.2. We use Word Error Rate (WER) for assessing our recognition models, as it is the prevalent metric for evaluating CSLR performance [97]. To measure the translation performance of our networks, we utilized BLEU [132] score, which is the most common metric for

---

<sup>1</sup><https://github.com/neccam/slt>

machine translation. While calculating the BLEU scores we utilize n-grams ranging from 1 to 4 to give the reader a better understanding of the translation performance on different segment levels.

**Training:** We used the Adam [94] optimizer to train our networks using a batch size of 32 with a learning rate of  $10^{-3}$  ( $\beta_1=0.9$ ,  $\beta_2=0.998$ ) and a weight decay of  $10^{-3}$ . We utilize plateau learning rate scheduling which tracks the development set performance, using WER and BLEU-4 scores as evaluation metrics for recognition and translation, respectively. We evaluate our network every 100 iterations. If the development score does not decrease for 8 evaluation steps, we decrease the learning rate by a factor of 0.7. This continues until the learning rate drops below a minimum value, which is set as  $10^{-6}$  in our experiments. We trained 3 networks for all of our experiments, with different random seeds, which are common between different experiments. We then report the best performing model for each network setup.

**Decoding:** Previous work has shown the importance of decoding algorithms and how they affect both the recognition and the translation performance [23, 24]. During the training and validation steps we employ a greedy search to decode both gloss sequences and spoken language sentences. At inference time, we utilize beam search decoding with widths ranging from 0 to 10. We also implement a length penalty [199] with  $\alpha$  values ranging from 0 to 2. We find the best performing combination of beam width and  $\alpha$  on the development set and use these values for the test set evaluation.

## 6.2.2 Text-to-Text Sign Language Translation

In Chapter 5, we used RNN-based attention mechanisms [113, 11] to tackle text-to-text SLT tasks on the PHOENIX14T dataset. However, Transformers [187] have become the new baseline for many natural language understanding and machine translation tasks. To evaluate the performance gain achieved over the RNN-based attention architectures, in our first set of experiments, we adapt the transformer backbone of our technique for text-to-text SLT protocols.

As can be seen in Table 6.1, utilizing transformers for text-to-text sign language translation improves the performance across all tasks, reaching an impressive 25.35/24.54 BLEU-



---

4 score on the development and test sets. We believe this performance gain is due to the more sophisticated attention architectures, namely self-attention modules, which learn the contextual information within both source and target sequences.

### 6.2.3 *Sign2Gloss*

To tackle the *Sign2Gloss* task, we utilize our SLRT network. Any CNN architecture can be used as spatial embedding layers to learn the sign language video frame representation while training SLRT in an end-to-end manner. However, due to hardware limitations (graphics card memory) we utilize pretrained CNNs as our spatial embeddings. We extract frame level representations from sign videos and train our sign language transformers to learn CSLR and SLT jointly in an end-to-end manner.

In our first set of experiments, we investigate which CNN we should be using to represent our sign videos. We utilize state-of-the-art EfficientNets [178], namely B0, B4 and B7 variants, which were trained on ImageNet [53]. We also use an Inception [175] network which was pretrained for learning sign language recognition in a CNN+LSTM+HMM setup [96]. In this set of experiments we employed a two layered transformer encoder model.

Table 6.2 shows that as the spatial embedding layer becomes more advanced, *i.e.* B0 vs B7, the recognition performance increases. However, our networks benefit more when we used pretrained features, as these networks had seen sign videos before and had the opportunity to learn kernels which can embed more meaningful representations in the latent space. We then tried utilizing Batch Normalization [87] followed by a ReLU [122] to normalize our inputs and allow our networks to learn more abstract non-linear representations. This improved our results drastically, giving us a boost of nearly 7% and 6% of absolute WER reduction on the development and test sets, respectively. Considering these findings, the rest of our experiments used the batch normalized pretrained CNN features of [96] followed by ReLU.

Next, we investigated the effects of having different numbers of transformer layers. Although having a larger number of layers would allow our networks to learn more

Text-to-Text Tasks (RNNs vs Transformers)	WER	DEV				WER	TEST			
		BLEU-1	BLEU-2	BLEU-3	BLEU-4		BLEU-1	BLEU-2	BLEU-3	BLEU-4
Gloss2Text [24] (Chapter 5)	-	44.40	31.83	24.61	20.16	-	44.13	31.47	23.89	19.26
<b>Our Gloss2Text</b>	-	<b>50.69</b>	<b>38.16</b>	<b>30.53</b>	<b>25.35</b>	-	<b>48.90</b>	<b>36.88</b>	<b>29.45</b>	<b>24.54</b>
Sign2Gloss2Text [24] (Chapter 5)	-	42.88	30.30	23.02	18.40	-	43.29	30.39	22.82	18.13
<b>Our Sign2Gloss2Text</b>	-	<b>47.73</b>	<b>34.82</b>	<b>27.11</b>	<b>22.11</b>	-	<b>48.47</b>	<b>35.35</b>	<b>27.57</b>	<b>22.45</b>
Sign2Gloss→Gloss2Text [24] (Chapter 5)	-	41.08	29.10	22.16	17.86	-	41.54	29.52	22.24	17.79
<b>Our Sign2Gloss→Gloss2Text</b>	-	<b>47.84</b>	<b>34.65</b>	<b>26.88</b>	<b>21.84</b>	-	<b>47.74</b>	<b>34.37</b>	<b>26.55</b>	<b>21.59</b>
Video-to-Text Tasks	WER	BLEU-1	BLEU-2	BLEU-3	BLEU-4	WER	BLEU-1	BLEU-2	BLEU-3	BLEU-4
CNN+LSTM+HMM [96]	24.50	-	-	-	-	26.50	-	-	-	-
<b>Our Sign2Gloss</b>	<b>24.88</b>	-	-	-	-	<b>24.59</b>	-	-	-	-
Sign2Text [24] (Chapter 5)	-	31.87	19.11	13.16	9.94	-	32.24	19.03	12.83	9.58
<b>Our Sign2Text</b>	-	<b>45.54</b>	<b>32.60</b>	<b>25.30</b>	<b>20.69</b>	-	<b>45.34</b>	<b>32.31</b>	<b>24.83</b>	<b>20.17</b>
<b>Our Best Recog. Sign2(Gloss+Text)</b>	<b>24.61</b>	<b>46.56</b>	<b>34.03</b>	<b>26.83</b>	<b>22.12</b>	<b>24.49</b>	<b>47.20</b>	<b>34.46</b>	<b>26.75</b>	<b>21.80</b>
<b>Our Best Trans. Sign2(Gloss+Text)</b>	<b>24.98</b>	<b>47.26</b>	<b>34.40</b>	<b>27.05</b>	<b>22.38</b>	<b>26.16</b>	<b>46.61</b>	<b>33.73</b>	<b>26.19</b>	<b>21.32</b>

Table 6.1: (Top) New baseline results for text-to-text tasks on Phoenix2014T [24] using transformer networks and (Bottom) Our best performing Sign Language Transformers compared against the state-of-the-art.

Spatial Embedding	DEV		TEST	
	del / ins	WER	del / ins	WER
EfficientNet-B0	47.22 / 1.59	57.06	46.09 / 1.75	56.29
EfficientNet-B4	40.73 / 2.45	51.26	38.34 / 2.80	50.09
EfficientNet-B7	39.29 / 2.84	50.18	37.05 / 2.76	47.96
Pretrained CNN	21.51 / 6.10	33.90	20.29 / 5.35	33.39
<b>+ BN &amp; ReLU</b>	<b>13.54 / 5.74</b>	<b>26.70</b>	<b>13.85 / 6.43</b>	<b>27.62</b>

Table 6.2: Impact of the Spatial Embedding Layer variants.

abstract representations, it also makes them prone to over-fitting. To this end, we built our SLRT networks using one to six layers and evaluate their CSLR performance.

# Layers	DEV		TEST	
	del/ins	WER	del/ins	WER
1	11.72 / 9.02	28.08	11.20 / 10.57	29.90
2	13.54 / 5.74	26.70	13.85 / 6.43	27.62
<b>3</b>	<b>11.68 / 6.48</b>	<b>24.88</b>	<b>11.16 / 6.09</b>	<b>24.59</b>
4	12.55 / 5.87	24.97	13.48 / 6.02	26.87
5	11.94 / 6.12	25.23	11.81 / 6.12	25.51
6	15.01 / 6.11	27.46	14.30 / 6.28	27.78

Table 6.3: Impact of different numbers of layers

Our recognition performance initially improves with additional layers (See Table 6.3). However, as we continue adding more layers, our networks started to over-fit on the training data, causing performance degradation. In the light of this, for the rest of our experiments, we constructed our sign language transformers using three layers.

#### 6.2.4 *Sign2Text* and *Sign2(Gloss+Text)*

In our next set of experiments we examine the performance gain achieved by unifying the recognition and translation tasks into a single model. As a baseline, we trained a *Sign2Text* network by setting our recognition loss weight  $\lambda_R$  to zero. We then jointly

train our sign language transformers, for recognition and translation, with various weightings between the losses.

Loss Weights		DEV		TEST	
$\lambda_R$	$\lambda_T$	WER	BLEU-4	WER	BLEU-4
1.0	0.0	24.88	-	24.59	-
0.0	1.0	-	20.69	-	20.17
1.0	1.0	35.13	21.73	33.75	21.22
2.5	1.0	26.99	22.11	27.55	21.37
5.0	1.0	<b>24.61</b>	22.12	<b>24.49</b>	<b>21.80</b>
10.0	1.0	24.98	<b>22.38</b>	26.16	21.32
20.0	1.0	25.87	20.90	25.73	20.93

Table 6.4: Training Sign Language Transformers to jointly learn recognition and translation with different weight on recognition loss.

As can be seen in Table 6.4, jointly learning recognition and translation with equal weighting ( $\lambda_R=\lambda_T=1.0$ ) improves the translation performance, while degrading the recognition performance compared to task specific networks. We believe this is due to the scale differences of the CTC and word-level cross entropy losses. Increasing the recognition loss weight improved both the recognition and the translation performance, demonstrating the value of sharing training between these related tasks.

Compared to previously published methods, our Sign Language Transformers surpass both their recognition and translation performance (See Table 6.1). We report a decrease of 2% WER over [96] on the test set in both *Sign2Gloss* and *Sign2(Gloss+Text)* setups. More impressively, both our *Sign2Text* and *Sign2(Gloss+Text)* networks doubled the previous state-of-the-art translation results (9.58 vs. 20.17 and 21.32 BLEU-4, respectively). Furthermore, our best performing translation *Sign2(Gloss+Text)* outperforms our previous text-to-text based *Gloss2Text* translation performance (19.26 vs 21.32 BLEU-4), which was previously proposed as a pseudo upper bound on performance in Chapter 5 [24]. This supports our claim that given more sophisticated network architectures, one would and should achieve better performance translating directly from video representations rather than doing text-to-text translation through an information limited gloss representation.

---

## 6.3 Qualitative Examples

In this section we report our qualitative results. We share the spoken language translations generated by our best performing *Sign2(Gloss+Text)* model given sign video representations (See Table 6.5 and Table 6.6). As the annotations in the PHOENIX14T dataset are in German, we share both the produced sentences and their translations in English.

Overall, the quality of the translations is good, and even where the exact wording differs, it conveys the same information. The most difficult translations seem to be named entities like locations which occur in limited contexts in the training data. Specific numbers are also challenging as there is no grammatical context to distinguish one from another. Despite this, the sentences produced follow standard grammar with surprisingly few exceptions.

## 6.4 Closing Remarks

Sign language recognition and understanding is an essential part of the sign language translation task. Previous translation approaches have relied heavily on recognition as the initial step in their pipeline. In this chapter we proposed Sign Language Transformers, a novel transformer based architecture to jointly learn sign language recognition and translation in an end-to-end manner. We utilized CTC loss to inject gloss level supervision into the transformer encoder, training it to do sign language recognition while learning meaningful representations for the end goal of sign language translation, without having an explicit gloss representation as an information bottleneck.

We evaluated our approach on the challenging PHOENIX14T dataset and report state-of-the-art sign language recognition and translation results, in some cases doubling the performance of previous translation approaches. Our first set of experiments have shown that using features which were pretrained on sign data outperformed using generic ImageNet based spatial representations. Furthermore, we have shown that jointly training for recognition and translation tasks improved the performance across both. More importantly, we have surpassed the text-to-text translation results, which






	<p>Reference: im süden schwacher wind . ( in the south gentle wind . )</p> <p>Ours: der wind weht im süden schwach . ( the wind blows gentle in the south . )</p>
	<p>Reference: ähnliches wetter dann auch am donnerstag . ( similar weather then also on thursday . )</p> <p>Ours: ähnliches wetter auch am donnerstag . ( similar weather also on thursday . )</p>
	<p>Reference: ganz ähnliche temperaturen wie heute zwischen sechs und elf grad . ( quite similar temperatures as today between six and eleven degrees . )</p> <p>Ours: ähnlich wie heute nacht das sechs bis elf grad . ( similar as today at night that six to eleven degrees . )</p>
	<p>Reference: heute nacht neunzehn bis fünfzehn grad im südosten bis zwölf grad . ( tonight nineteen till fifteen degrees in the southeast till twelve degrees . )</p> <p>Ours: heute nacht werte zwischen neun und fünfzehn grad im südosten bis zwölf grad . ( tonight values between nine and fifteen degrees in the southeast till twelve degrees . )</p>
	<p>Reference: am sonntag im norden und in der mitte schauer dabei ist es im norden stürmisch . ( on sunday in the north and center shower while it is stormy in the north . )</p> <p>Ours: am sonntag im norden und in der mitte niederschläge im norden ist es weiter stürmisch . ( on sunday in the north and center rainfall in the north it is continuously stormy . )</p>

Table 6.5: Generated spoken language translations by our models - I.

was set as a virtual upper-bound, by directly translating spoken language sentences from video representations.

As with any other emerging field, SLT is in need of more data. Although incorporating sign glosses seems essential for future translation systems, it might not be feasible for the longer term as annotating sign glosses is a laborious task. Ideally, we would like





 <p>Reference: im süden und südwesten gebietsweise regen sonst recht freundlich . ( in the south and southwest partly rain otherwise quite friendly . )</p> <p>Ours: im südwesten regnet es zum teil kräftig . ( in the southwest partly heavy rain . )</p>
 <p>Reference: in der nacht sinken die temperaturen auf vierzehn bis sieben grad . ( at night the temperatures lower till fourteen to seven degrees . )</p> <p>Ours: heute nacht werte zwischen sieben und sieben grad . ( tonight values between seven and seven degrees . )</p>
 <p>Reference: heute nacht ist es meist stark bewölkt örtlich regnet oder nieselt es etwas . ( tonight it is mostly cloudy locally rain or drizzle . )</p> <p>Ours: heute nacht ist es verbreitet wolkenverhangen gebietsweise regnet es kräftig . ( tonight it is widespread covered with clouds partly strong rain . )</p>
 <p>Reference: an der saar heute nacht milde sechzehn an der elbe teilweise nur acht grad . ( at the saar river tonight mild sixteen at the elbe river partly only eight degrees . )</p> <p>Ours: im rhein und südwesten macht sich morgen nur knapp über null grad . ( in the rhine river and south west becomes just above zero degrees . )</p>

Table 6.6: Generated spoken language translations by our models - II.

be able to utilize the signed content from broadcasts, which is plentiful. However, it is not be feasible to annotate the large amounts of data that is produced daily. In addition, even if such annotations were obtained for one sign language, it is unlikely to transfer to other sign languages.

To generalize to other sign languages and to make use of readily available broadcast data, we believe future research should focus more on the building blocks of the sign, also known as subunits, which are generally common between sign languages. From a spoken language point of view, subunits can be seen as the phonetic alphabet of the signs. If we consider a system which aims to translate texts from images, it would not

start with a word classifier but with Optical Character Recognition (OCR) [119]. It would then combine the recognized characters to build up words and sentences. Such an OCR system would also be useful for recognizing other languages with similar alphabets. Considering this, focusing on subunits and building systems that can robustly recognize the manual and non-manual features of signs seems the logical next step for future research.

In an attempt to move towards a phonetic representation, to reduce our dependence on sign gloss annotations and to incorporate manual and non-manual features into SLT pipelines, we propose a novel multi-channel transformer architecture for multi-articulatory SLT in Chapter 7. We use hand, mouthing and upper body pose representations to model sign languages in a holistic manner by learning the inter and intra-channel contextual relationships. We also introduce a channel anchoring loss to help our models preserve channel specific information while also regulating translation loss against overfitting. More importantly, we show it is possible to achieve on par translation performance with models which require labor intensive gloss annotations.



## **Chapter 7**

# **Multi-Channel Transformers for Multi-Articulatory Sign Language Translation**

To date, the literature that underpins vision based sign language research has predominantly focused on using the manual features to realize sign language recognition and translation [133, 101, 24], thus ignoring the rich and essential information contained in the non-manual features. This focus on the manual features is partially responsible for the common misconception that sign language recognition and translation problems are special sub-tasks of the gesture recognition field [139]. Sign language is as rich and complex as any spoken language. However, the multi-channel nature adds additional complexity as channels are not synchronised.

In contrast to much of the existing literature, in this chapter we approach sign language translation by incorporating both manual and non-manual features for the Sign Language Translation (SLT) task. To achieve this, we use multiple channels which correspond to a subset of the articulatory subunits of sign, namely hand shape, upper body pose and mouthings. We explore several approaches to combine the information present in these channels using both early and late fusion building upon the transformer architecture developed in Chapter 6. Based on these findings we then introduce a novel deep

learning architecture, named *Multi-channel Transformers*. This approach incorporates both inter and intra-channel contextual relationships to learn meaningful spatio-temporal representations of asynchronous multi-channel source signals. Although this approach was designed specifically for SLT, we believe it can also be used to tackle other multi-channel sequence to sequence learning tasks, such as audio-visual speech recognition [3].

We evaluate our approach on the challenging RWTH-PHOENIX-Weather-2014T Dataset (PHOENIX14T) which provides both sign gloss annotations and spoken language translations. Previous approaches (*e.g.* [24] of Chapter 5 and [27] of Chapter 6) heavily relied upon the sign gloss annotations of PHOENIX14T, which are labor intensive to obtain. We aim to remove this dependency on gloss annotation, by utilizing channel specific features obtained from related tasks, such as human pose estimation approaches [28, 81] to represent the upper body pose channel or lip reading features [37, 10] to represent mouthings [100, 99]. Removing the dependency on manual annotation allows our approach to scaled beyond what is possible with previous techniques, potentially allowing the use of huge collections of un-annotated data. We empirically show that by integrating multiple articulator channels into our multi-channel transformer, it is possible to achieve state-of-the-art SLT performance which is *on par* with models trained using additional gloss annotation.

The remainder of this chapter is structured as follows: In Section 7.1, we present the proposed multi-channel transformer architecture. We introduce our experimental setup and implementation details in Section 7.2. We then report extensive quantitative results in Section 7.3 and share translation samples in Section 7.4. We finally conclude this chapter in Section 7.5 by discussing our findings and possible future work.

## 7.1 Methodology

In this chapter, we extend the transformer network architecture and adapt it to the task of multi-articulatory SLT. We introduce two novel layers, namely the *Multi-Channel Encoder* and the *Multi-Channel Decoder*. An overview of our approach

applied to the task of SLT can be seen in Figure 7.1. The multi-channel encoder aims to learn the inter and intra-channel contextual relationships of sign articulator channels. We utilize *Channel-wise Self Attention* modules to attend to each sign articulator individually. Outputs of these modules are then passed to a *Multi-Channel Encoder Attention* module, which enriches the channel representation by modeling them in the context of other sign articulators. When available, we utilize anchoring losses to help networks maintain channel specific information using weak annotations obtained from relevant classifiers. The spatio-temporal channel representations are then passed to the *Multi-Channel Decoder* layer, which employs a *Multi-Channel Decoder Attention* module that combines multiple source channel representations to produce spoken language sentences in an auto-regressive manner. Detailed visualizations of the proposed modules can be seen in Figure 7.2.

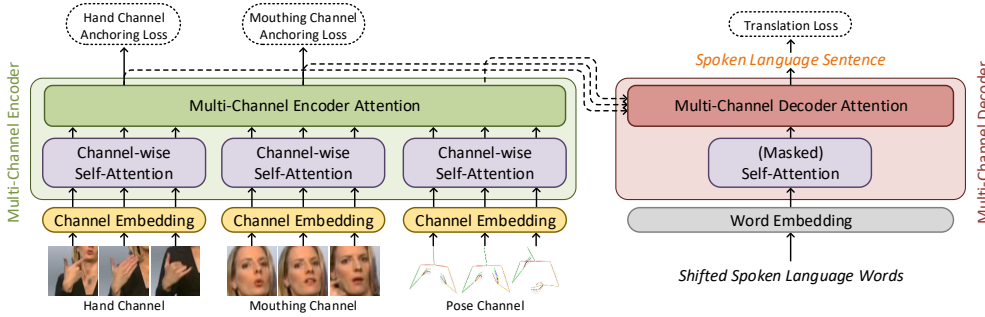


Figure 7.1: An overview of the proposed Multi-channel Transformer [26] approach.

Given source sequences  $\mathcal{X} = (X_1, \dots, X_N)$ , where  $X_i$  is the  $i^{\text{th}}$  source channel with a cardinality of  $T_i$ , our objective is to learn the conditional probability  $p(\mathcal{Y}|\mathcal{X})$ , where  $\mathcal{Y} = (y_1, \dots, y_U)$  is the target sequence with  $U$  tokens. In the application domain of SLT, source channels correspond to representations of the manual and non-manual features of the sign, such as hand shapes and mouthings, while the target sequence is the spoken/written language sentence. To keep the formulation simple, and to focus the readers attention on the differentiating factors of the proposed architecture, we omit the layer normalization and residual connections, which are the same as the original transformer network [187].

In the rest of this section, we introduce each component of our *Multi-Channel Trans-*

*former* and describe them in detail.

### 7.1.1 Channel and Word Embeddings

As with other machine translation tasks, we start by projecting both the source channel features and the one-hot word vectors into a denser embedding where similar inputs lie close to one-another. To achieve this we use linear layers. We employ normalization and activation layers to change the scale of the embedded channel features and give additional non-linear representational capability to the model. The transformer networks do not have an implicit structure to model the position of a token within the sequence. To overcome this, we employ positional encoding [187] to add temporal ordering to the embedded representations. The embedding process for an input feature  $x_{i,t}$  coming from the  $i^{\text{th}}$  channel at time  $t$  can be formalized as:

$$\hat{x}_{i,t} = \text{Activ}(\text{Norm}(x_{i,t}W_i^{\text{ce}} + b_i^{\text{ce}})) + \text{PosEnc}(t) \quad (7.1)$$

where  $W_i^{\text{ce}}$  and  $b_i^{\text{ce}}$  are channel specific learnt parameters of the linear projection layers. Similarly, the word embedding is as follows:

$$\hat{y}_u = y_u W^{\text{we}} + b^{\text{we}} + \text{PosEnc}(u) \quad (7.2)$$

where  $W^{\text{we}}$  and  $b^{\text{we}}$  are the weights of a linear layer which are either learned from scratch or pretrained on a large corpus [14, 90].

### 7.1.2 Multi-Channel Encoder Layer

**Channel-wise Self Attention (cs):** Each multi-channel encoder layer starts by learning the contextual relationships within a single channel by utilizing individual self-attention layers. As per the original transformer implementation, we use the scaled dot product scoring function in the attention mechanisms. Given embedded source channel representations,  $\hat{X}_i$ , we obtain Queries, Keys and Values for the channel  $i$  as<sup>1</sup>:

<sup>1</sup>Note that we use a vectorized formulation in our equations. All softmax and bias addition operations are done row-wise.

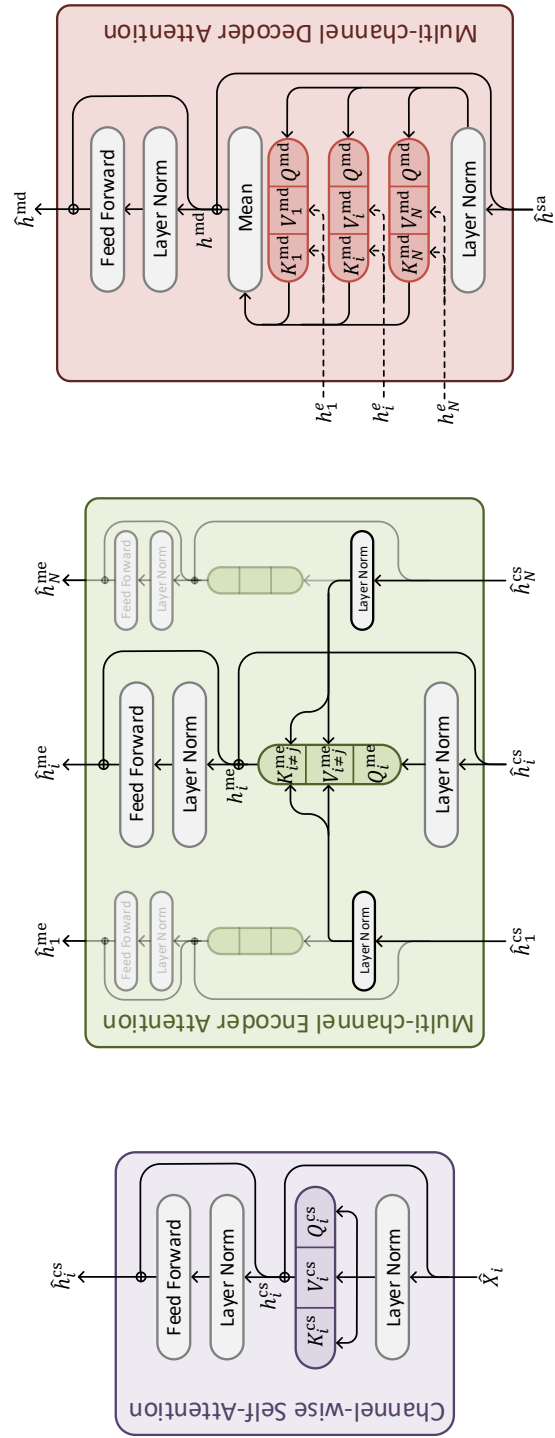


Figure 7.2: A detailed overview of the proposed multi-channel attention modules.

$$\begin{aligned}
Q_i^{\text{cs}} &= \hat{X}_i W_i^{\text{cs,q}} + b_i^{\text{cs,q}} \\
K_i^{\text{cs}} &= \hat{X}_i W_i^{\text{cs,k}} + b_i^{\text{cs,k}} \\
V_i^{\text{cs}} &= \hat{X}_i W_i^{\text{cs,v}} + b_i^{\text{cs,v}}
\end{aligned} \tag{7.3}$$

which are then passed to the channel-wise self attention function to have their intra channel contextual relationship modeled as:

$$h_i^{\text{cs}} = \text{softmax} \left( \frac{Q_i^{\text{cs}} (K_i^{\text{cs}})^T}{\sqrt{d_m}} \right) V_i^{\text{cs}} \tag{7.4}$$

where  $h_i^{\text{cs}}$  is the spatio-temporal representation of the  $i^{\text{th}}$  source channel and  $d_m$  is the hidden size of the model. We also utilize individual feed forward layers as described in [187] for each channel as:

$$\text{FF}(x) = \max \left( 0, x W_1^{\text{ff}} + b_1 \right) W_2^{\text{ff}} + b_2 \tag{7.5}$$

By feeding the contextually modeled channel representations through feed forward layers, we obtain the final outputs of the channel-wise attention layer of our multi-channel encoder layer as:

$$\hat{h}_i^{\text{cs}} = \text{FF}_i^{\text{cs}}(h_i^{\text{cs}}) \tag{7.6}$$

**Multi-channel Encoder Attention (me):** We now introduce the multi-channel encoder attention, which learns the contextual relationship between the self-attended channel representations. As we are using dot product attention, we start by obtaining  $Q$ ,  $K$  and  $V$  for each source as:

$$\begin{aligned}
Q_i^{\text{me}} &= \hat{h}_i^{\text{cs}} W_i^{\text{me,q}} + b_i^{\text{me,q}} \\
K_i^{\text{me}} &= \hat{h}_i^{\text{cs}} W_i^{\text{me,k}} + b_i^{\text{me,k}} \\
V_i^{\text{me}} &= \hat{h}_i^{\text{cs}} W_i^{\text{me,v}} + b_i^{\text{me,v}}
\end{aligned} \tag{7.7}$$

These values are then passed to the multi-channel attention layers where the Queries of each channel are used to estimate the scores over the concatenated Keys of the other channels. These scores are then used to calculate the channel-fused representations by taking a weighted sum over the other channels' concatenated Values. More formally, multi-channel attention can be defined as:

$$h_i^{\text{me}} = \text{softmax} \left( \frac{Q_i^{\text{me}} \left( [\forall K_j^{\text{me}} \text{ where } j \neq i] \right)^T}{\sqrt{d_m}} \right) [\forall V_j^{\text{me}} \text{ where } j \neq i] \quad (7.8)$$

We would like to note that, the concatenation operation ( $[ \ ]$ ) is performed over the time axis, thus making our approach applicable to tasks where the source channels have different numbers of tokens. We then pass multi-channel attention outputs to individual feed forward layers to obtain the final outputs of the multi-channel encoder layer as:

$$\hat{h}_i^{\text{me}} = \text{FF}_i^{\text{me}}(h_i^{\text{me}}) \quad (7.9)$$

Several multi-channel encoder layers can be stacked to form the encoder network with the aim of learning more complex multi-channel contextual representations,  $h^e = (h_1^e, \dots, h_N^e)$ , where  $h_i^e$  is the output corresponding to the  $i^{\text{th}}$  source channel.

### 7.1.3 Multi-Channel Decoder Layer

Transformer networks utilize a masked self attention and an encoder-decoder attention in each decoder layer (See Figure 3.3). The subsequent masking on self-attention is essential, as the target tokens' successors will not be available at inference time. In our approach, we also employ the masked self-attention to model the contextual relationship between target tokens' and its predecessors. However, we replace encoder-decoder attention with multi-channel decoder attention, which is modified to work with multiple source channel representations. Given the word embeddings  $\hat{\mathcal{Y}}$  of a sentence  $\mathcal{Y}$ , we first obtain the masked self-attention (sa) outputs  $\hat{h}^{\text{sa}}$  using the generic approach [187], which are then in turn passed to our multi-channel decoder attention.

**Multi-channel Decoder Attention (md):** In generic transformers, encoder-decoder attention Queries are obtained from the decoder self-attention estimates,  $\hat{h}^{\text{sa}}$ , while Keys and Values are calculated from the final encoder layer outputs,  $h_e$ . In order to incorporate information coming from multiple channels using transformer models, we propose the multi-channel decoder attention module. We first obtain the  $Q$ ,  $K$  and  $V$  as:

$$\begin{aligned} Q^{\text{md}} &= \hat{h}^{\text{sa}} W^{\text{md},q} + b^{\text{md},q} \\ K_i^{\text{md}} &= h_i^e W_i^{\text{md},k} + b_i^{\text{md},k} \\ V_i^{\text{md}} &= h_i^e W_i^{\text{md},v} + b_i^{\text{md},v} \end{aligned} \quad (7.10)$$

Note that each source channel  $i$  has their own learned Key and Value matrices,  $W_i^{\text{md},k}$  and  $W_i^{\text{md},v}$  respectively.

These are then passed to the multi-channel decoder attention module where Queries of each target token are scored against all channel Keys. Channel scores are then used to calculate the weighted average of their respective Values. Individual channel outputs are averaged to obtain the final output of the multi-channel decoder attention module. This process can be formalized as:

$$h^{\text{md}} = \frac{1}{N} \sum_{i=1}^N \left( \text{softmax} \left( \frac{Q^{\text{md}} (K_i^{\text{md}})^T}{\sqrt{d_m}} \right) V_i^{\text{md}} \right) \quad (7.11)$$

The attention module outputs are then passed through a feed forward layer to obtain the final representations of the multi-channel decoder layer as:

$$\hat{h}^{\text{md}} = \text{FF}^{\text{md}}(h^{\text{md}}) \quad (7.12)$$

Like the multi-channel encoder layer, multiple decoder layers can be stacked to improve the representation capabilities of the decoder network. The output of the stacked decoder is denoted as  $h^d = (h_1, \dots, h_U)$  which is used to condition target token generation.



### 7.1.4 Loss Functions

We propose training multi-channel transformers using two types of loss functions, namely Translation Loss, which is commonly used in machine translation, and Channel Anchoring Loss, which aims to preserve channel specific information during encoding.

**Translation Loss:** Although different loss functions have been used to train translation models, such as a mixture-of-softmaxes [203], token level cross-entropy loss is the most common approach to learn network parameters. Given a source-target pair, the translation loss,  $\mathcal{L}_T$ , is calculated as the accumulation of the error at each decoding step  $u$ , which is estimated using a classification loss over the target vocabulary as:

$$\mathcal{L}_T = 1 - \prod_{u=1}^U \sum_{g=1}^G p(y_u^g) p(\hat{y}_u^g) \quad (7.13)$$

where  $p(y_u^g)$  and  $p(\hat{y}_u^g)$  represent the ground truth and the generation probabilities of the target  $y^g$  at decoding step  $u$ , respectively, and  $G$  is the target language vocabulary size. In our networks, the probability of generating target token  $y_u$  at the decoding step  $u$  is conditioned on the hidden state of the decoder network  $h_u^d$  at the corresponding time step,  $p(\hat{y}_u) = p(\hat{y}_u | h_u^d)$ . Softmaxed linear projection of  $h_u^d$  is used to model the probability of producing tokens over the whole target vocabulary as:

$$p(\hat{y}_u | h_u^d) = \text{softmax}(h_u^d W^o + b^o) \quad (7.14)$$

where  $W^o$  and  $b^o$  are trainable parameters of a linear layer.

**Channel Anchoring Loss:** For source channels, where we have access to a relevant classifier, we use an anchoring loss to preserve channel specific information. Predictions of these classifiers are used as ground truth to calculate token level cross entropy losses in the same manner as the translation loss. Given the classifier outputs corresponding to the  $i^{th}$  channel,  $C_i = (c_{i,1}, \dots, c_{i,T_i})$ , and the hidden state of the encoder,  $h^e$ , we first calculate the prediction probabilities over the target channel classes as:

$$p(\hat{c}_{i,t} | h^e) = \text{softmax}(h^e W_i^o + b_i^o) \quad (7.15)$$

where  $p(\hat{c}_{i,t}|h^e)$  represent the prediction probabilities over the  $i^{th}$  channel’s classifier vocabulary, while  $W_i^o$  and  $b_i^o$  are weights and biases of the linear layer used for the  $i^{th}$  channel, respectively. We then use a modified version of the Equation 7.13 to calculate the  $i^{th}$  channel’s anchoring loss,  $\mathcal{L}_{A,i}$ , as:

$$\mathcal{L}_{A,i} = 1 - \prod_{t=1}^T \sum_{g=1}^{G_i} p(c_{i,t}^g) p(\hat{c}_{i,t}^g | h_t^e) \quad (7.16)$$

where  $p(c_{i,t}^g)$  and  $p(\hat{c}_{i,t}^g)$  represent the classifier output and the predicted probabilities of the class  $c_i^d$  at the encoders  $t^{th}$  step, respectively, while  $G_i$  is the number of target classes the classifier corresponding to channel  $i$  predicted. For example, as our hand channel, we used a hand shape classifier’s convolutional layer as our source channel and the same classifier’s outputs as our anchoring loss to preserve the hand shape information during training as well as to regularize the translation loss.

**Total Loss:** We use a weighed combination of Translation loss,  $\mathcal{L}_T$ , and Anchoring losses,  $\mathcal{L}_A = (\mathcal{L}_{A,1}, \dots, \mathcal{L}_{A,N})$ , during training as:

$$\mathcal{L} = \lambda_T \mathcal{L}_T + \lambda_A \sum_{i=1}^N \mathcal{L}_{A,i} \quad (7.17)$$

where  $\lambda_T$  and  $\lambda_A$  decide the importance of each loss function during training.

## 7.2 Implementation and Evaluation Details

**Dataset:** We evaluate our model on the challenging PHOENIX14T [24] dataset, which is currently the only publicly available large vocabulary continuous SLT dataset aimed at vision based sign language research. See Chapter 4 for further details of the dataset.

**Sign Channels:** We use three different articulators/channels to represent the manual and non-manual features of the sign, namely hand shapes, mouthings and upper body pose.

*Hand Shape and Mouthing Channels:* Any symbolic or continuous representation can be used in the proposed multi-channel transformer architecture. In this work, we employ Convolutional Neural Network (CNN) classifiers proposed by Koller *et al.*

---

[96]. In their work, the authors utilize Inception CNN architecture [175] and train individual networks to predict hand shape, sign gloss and mouthings from full frames (*i.e.* not cropped) using weakly aligned labels. They iteratively finetune the training labels by using a HMM-based forced alignment approach [101] until the validation set Continuous Sign Language Recognition (CSLR) performance converges.

We utilize the aforementioned hand shape and mouthing recognition networks and obtain 1024 dimensional continuous feature vectors for each frame by extracting the activations from the last layer before the fully connected layer. We use the forced aligned class labels from the same networks to anchor the channel representations. Although these networks were trained on 61 and 40 hand shape and mouthing classes, respectively (including transition/silence class), the predictions only contained 52 and 36 classes. Hence, our anchoring losses are calculated over the predicted number of classes.

*Upper Body Pose Channel:* To represent the upper body pose of the signers, we start by extracting 2D skeletal pose information using the OpenPose library [28]. We only consider the upper body and both hands' joint locations provided by the OpenPose, which are represented as 2D coordinates on the image plane. We then pass the extracted 2D joint locations to the 2D-to-3D uplifting approach proposed by Zelinka *et al.* [207]. Designed for sign language production, their method formalizes 2D-to-3D uplifting as an inverse-kinematics problem, and produces a viable 3D skeleton based on OpenPose predictions using a gradient descent solver. The resulting 3D joint positions of 50 upper body joints are then flattened and passed through a linear projection layer to match the transformer models hidden size. As there were no prior subunit classes for the upper body pose on PHOENIX14T, we do not utilize an anchoring loss on the pose channel.

**Training and Network Details:** Our networks are trained using the PyTorch framework [134] with a modified version<sup>2</sup> of the JoeyNMT library [104]. We use Adam [94] optimizer with a batch size of 32, a learning rate of  $10^{-3}$  ( $\beta_1 = 0.9, \beta_2 = 0.998$ ) and a weight decay of  $10^{-3}$ . We utilize Xavier [69] initialization and train all networks from scratch. We do not apply dropout and only use a single headed scaled dot-product

---

<sup>2</sup>The code, the models and the channel features will be made publicly available.

attention to reduce the number of hyper-parameters in our experiments.

We employ a modified version of plateau learning rate scheduling with validation performance on the development set as its criterion. Our networks are evaluated after every 100 iterations (roughly half an epoch). If the validation score does not improve for 8 consecutive evaluation steps, the scheduler decreases the learning rate by a factor of 0.7. This continues until either the learning rate drops below a minimum value ( $10^{-4}$  in our experiments) or there were no further improvements in the validation score from the previous learning rate step. We employ early stopping by using the model with highest validation score to report the final scores. All of our loss functions are normalized by the number of samples in each batch.

**Decoding:** During training we use a greedy search to evaluate development set translation performance. At inference, we employ beam search decoding with the beam width ranging from 0 to 10. We also employ a length penalty as proposed by [88] with  $\alpha$  values ranging from 0 to 5. We use the development set to find the best performing beam width and  $\alpha$ , and use these during test set inference for final results.

**Performance Metrics:** We use BLEU [132] and ROUGE [110] scores to measure the translation performance. Following extensive code optimisation, our networks take roughly half an hour each to train including beam search decoding. Hence, to give the reader a better understanding of the networks behaviour, we repeat each experiment 10 times and report mean and standard deviation of BLEU-4 and ROUGE scores on both development and test sets. We also report our best result for every setup based on BLEU-4 score as per the development set. BLEU-4 score is also used as the validation score for our learning scheduler and for early stopping.

### 7.3 Quantitative Experiments

In this section we propose several multi-channel SLT experimental setups and report our quantitative results. We start by sharing our experiments to identify the optimal embedding setup for using CNN features with transformer networks. We then report single channel SLT performance using the best embedding setup with different network

---

architectures, varying in number of hidden units, both to set a baseline for our multi-channel approaches and to find the optimal network size. After that, we propose two naive channel fusion approaches, namely early fusion and late fusion, to set a fusion benchmark for our novel *Multi-channel Transformer* architecture. Finally, we report the performance of the multi-channel transformer approach with and without the channel anchoring losses and compare our results against the state-of-the-art.

### 7.3.1 Channel Feature and Word Embeddings

In our first set of experiments we investigate different ways to embed CNN based channel features and one-hot word vectors. Machine translation orientated transformer implementations either use pretrained word embeddings or train a linear projection layer from scratch. Although not stated in the original paper [187], the official transformer implementation also utilizes *embedding scaling*, where the projected word representations are multiplied by a constant which is the square root of the hidden size<sup>3</sup>.

Compared to the one-hot vectors which have a constant scale between  $[0 - 1]$ , CNN features can have an arbitrary scale. To see how important the input scale is and to examine the effects of the different embedding setups, we initially trained translation networks that only used hand channel features as input. Each network has two layers, with a hidden size of 64 and 128 position-wise feed forward units.

As can be seen in the first row of Table 7.1, the translation performance degrades drastically when we apply the commonly used embedding scaling on either of the embeddings. We have experimentally found that transformer networks are extremely sensitive to input scale and this is substantiated by these results. Thus, in our next set of experiments we investigate ways to normalize inputs to control their scale. To do so, we utilize batch normalization [87] and soft-sign activation [125]. While batch normalization scales the inputs between  $[-3, 3]$  it has also been shown to improve convergence rate. On the other hand, soft-sign activation scales the inputs between  $[-1, 1]$  while also enhancing the representation capability of the embedding due to its non-linear nature.

---

<sup>3</sup>[github.com/tensorflow/models/blob/master/official/nlp/transformer/embedding\\_layer.py#L83](https://github.com/tensorflow/models/blob/master/official/nlp/transformer/embedding_layer.py#L83)

Feature Embedding			Word Embedding			Dev Set		Test Set					
Scaling	BatchNorm	SoftSign	Scaling	Batch Norm	Soft-Sign	BLEU-4	ROUGE	BLEU-4	ROUGE				
						Best	mean $\pm$ std	mean $\pm$ std	mean $\pm$ std				
$\times$	-	-	$\times$	-	-	<b>15.46</b>	<b>15.08</b> $\pm$ 0.25	<b>40.57</b>	<b>39.39</b> $\pm$ 0.55	<b>15.98</b>	<b>15.23</b> $\pm$ 0.54	<b>39.97</b>	<b>39.44</b> $\pm$ 0.80
$\checkmark$	-	-	$\times$	-	-	13.96	13.39 $\pm$ 0.36	37.51	37.07 $\pm$ 0.33	14.29	13.77 $\pm$ 0.47	37.91	37.54 $\pm$ 0.60
$\times$	-	-	$\checkmark$	-	-	14.54	14.20 $\pm$ 0.26	37.86	38.02 $\pm$ 0.70	14.80	14.58 $\pm$ 0.46	38.35	38.38 $\pm$ 0.72
$\checkmark$	-	-	$\checkmark$	-	-	13.52	12.88 $\pm$ 0.36	36.93	36.14 $\pm$ 0.66	13.99	13.29 $\pm$ 0.53	37.72	36.58 $\pm$ 0.81
$\times$	$\checkmark$	-	$\times$	$\times$	-	<b>16.35</b>	<b>15.64</b> $\pm$ 0.46	<b>41.30</b>	<b>40.41</b> $\pm$ 0.55	<b>16.09</b>	<b>15.60</b> $\pm$ 0.50	<b>40.89</b>	<b>40.15</b> $\pm$ 0.70
$\times$	$\times$	-	$\checkmark$	-	-	14.49	14.07 $\pm$ 0.25	37.96	37.86 $\pm$ 0.58	14.72	14.33 $\pm$ 0.38	37.49	37.53 $\pm$ 0.43
$\times$	$\checkmark$	-	$\checkmark$	$\checkmark$	-	15.17	14.69 $\pm$ 0.30	39.92	39.10 $\pm$ 0.56	15.50	14.99 $\pm$ 0.64	39.83	39.26 $\pm$ 0.81
$\times$	-	$\checkmark$	$\times$	-	$\times$	<b>16.40</b>	<b>15.74</b> $\pm$ 0.55	<b>41.90</b>	<b>40.73</b> $\pm$ 0.63	14.95	<b>15.63</b> $\pm$ 0.48	39.31	39.93 $\pm$ 0.57
$\times$	-	$\times$	$\times$	-	$\checkmark$	15.75	15.10 $\pm$ 0.35	39.72	39.47 $\pm$ 0.46	<b>16.33</b>	15.41 $\pm$ 0.55	40.74	39.76 $\pm$ 0.67
$\times$	-	$\checkmark$	$\times$	-	$\checkmark$	15.98	15.50 $\pm$ 0.35	41.02	40.24 $\pm$ 0.51	15.64	15.49 $\pm$ 0.48	<b>40.79</b>	<b>40.02</b> $\pm$ 0.59
$\times$	$\checkmark$	$\checkmark$	$\times$	$\times$	$\times$	<b>16.44</b>	<b>15.70</b> $\pm$ 0.41	40.79	40.45 $\pm$ 0.64	<b>16.18</b>	15.63 $\pm$ 0.65	<b>40.62</b>	40.07 $\pm$ 0.80
$\times$	$\times$	$\times$	$\times$	$\checkmark$	$\checkmark$	15.15	14.18 $\pm$ 0.42	39.30	37.78 $\pm$ 0.77	15.14	14.30 $\pm$ 0.53	38.41	37.53 $\pm$ 0.82
$\times$	$\checkmark$	$\checkmark$	$\times$	$\checkmark$	$\checkmark$	15.59	15.04 $\pm$ 0.33	39.83	39.40 $\pm$ 0.46	14.85	14.99 $\pm$ 0.51	39.26	39.30 $\pm$ 0.65
$\times$	$\checkmark$	$\checkmark$	$\times$	$\checkmark$	$\checkmark$	15.98	15.62 $\pm$ 0.21	<b>41.63</b>	<b>40.63</b> $\pm$ 0.62	15.97	<b>15.65</b> $\pm$ 0.49	40.54	<b>40.24</b> $\pm$ 0.74
$\times$	$\times$	$\checkmark$	$\times$	$\checkmark$	$\checkmark$	15.09	14.70 $\pm$ 0.26	39.36	38.95 $\pm$ 0.25	14.70	14.80 $\pm$ 0.49	39.38	38.87 $\pm$ 0.67

Table 7.1: Effects of using different embedding setups on hand channel features to spoken language translation performance.

---

Individually, both batch norm and soft-sign significantly improve the translation performance when applied to the projected CNN features (see second and third rows of Table 7.1). We then investigate their combined use on both CNN features and word embeddings. Although there were several comparable setups, we concur that the joint application of batch norm and soft sign only on the CNN features yield the most stable and balanced performance in terms of development and test set for BLEU-4 and ROUGE scores. We believe this setup’s success is due to the already scaled nature of the word embeddings and the additional stability and non-linearity introduced by applying soft-sign and batch norm to the projected CNN features. Therefore, for the rest of our experiments we utilize this embedding setup.

### 7.3.2 Single Channel Baselines

In the next set of experiments, we train single channel SLT models. The main objective of these experiments is to set translation baselines for all future multi-channel fusion models. However, we would also like to examine the relative information presented in each channel by comparing their translation performance against one another. In addition, we wish to identify the optimal network setup for each channel to guide the future experiments. Therefore, we conduct experiments with four network setups for all three articulators with sizes varying from 32x64 to 256x512 (hidden size (HS) x number of feed forward (FF) units). All networks were built using two encoder and decoder layers.

As can be seen in Table 7.2, **H**and is the best performing channel in all network setups. Furthermore, using a network setup of 128x256 outperforms all of the alternatives. We believe this is closely related to the limited number of training samples we have and the over-fitting issues that come with it. Therefore, for the rest of our experiments we use 128x256 parameters for each channel.

We further train a **G**loss single channel network to set a baseline for our multi-channel approaches to compare against. As shown in Table 7.2, using CNN features that were trained using gloss level annotations outperforms all single sign articular based models (19.52 vs. 16.44 dev BLEU-4 score). Although the 256x512 network setup

obtained the best individual development and test set translation performances, the mean performance of the 128x256 network was better, encouraging us to utilize this setup going forward.

### 7.3.3 Early and Late Fusion of Sign Channels

To set another benchmark for our multi-channel transformer, we propose two naive multi-channel fusion approaches, namely early and late fusion. In the early fusion setup, features from different channels are concatenated to create a fused representation of each frame. These representations are then used to train SLT models, as if they were features coming from a single channel. Hence, the contextual relationship is performed in an implicit manner by the transformer architecture. In our second, late fusion setup, individual SLT models are built which are then fused at the decoder output level, *i.e.*  $h^d$ , by concatenation. The fused representation is then used to generate target tokens using a linear projection layer. Compared to early fusion, this approach’s capability to learn more abstract relationships is limited as the fusion is only done by a single linear layer. We examine all four possible fusions of our three channels. Network setup is set to linearly scale with respect to the number of channels that are fused together with a factor of 128x256 per channel.

As can be seen in Table 7.3, fusion of **H**ands and **M**outh yields slightly better results than single channel translation models (excluding gloss). However, unlike late fusion, which saw improvement in all scenarios, early fusion’s performance gets worse as more features are added to the network. As this means having more parameters in our networks, we believe this is due to the natural propensity of the transformers to over-fit on small training datasets, like ours.

### 7.3.4 Multi-Channel Transformers

In this set of experiments we examine the translation performance of the proposed multi-channel transformer architecture for multi-articulatory SLT. We first start by investigating the effects of the anchoring loss. We then compare our best performing



Channel	HSxFF	Dev Set				Test Set			
		BLEU-4		ROUGE		BLEU-4		ROUGE	
		Best	mean $\pm$ std	-	mean $\pm$ std	-	mean $\pm$ std	-	mean $\pm$ std
<b>Hand</b>	32x64	14.54	14.05 $\pm$ 0.42	38.47	38.49 $\pm$ 0.45	13.88	13.80 $\pm$ 0.63	38.05	38.04 $\pm$ 0.49
<b>Hand</b>	64x128	<b>16.44</b>	15.70 $\pm$ 0.41	40.79	40.45 $\pm$ 0.64	16.18	15.63 $\pm$ 0.65	40.62	40.07 $\pm$ 0.80
<b>Hand</b>	128x256	16.32	<b>15.91</b> $\pm$ 0.43	<b>41.87</b>	<b>41.08</b> $\pm$ 0.73	<b>16.76</b>	<b>16.02</b> $\pm$ 0.88	<b>41.85</b>	<b>40.67</b> $\pm$ 0.93
<b>Hand</b>	256x512	16.06	15.41 $\pm$ 0.46	41.46	40.13 $\pm$ 0.83	15.43	15.57 $\pm$ 0.60	40.48	39.88 $\pm$ 0.71
<b>Mouthing</b>	32x64	11.70	11.24 $\pm$ 0.34	33.26	32.84 $\pm$ 0.60	10.77	11.01 $\pm$ 0.34	33.51	33.05 $\pm$ 0.34
<b>Mouthing</b>	64x128	12.91	12.55 $\pm$ 0.30	36.22	35.17 $\pm$ 0.71	12.83	12.62 $\pm$ 0.50	35.30	35.04 $\pm$ 0.72
<b>Mouthing</b>	128x256	<b>13.74</b>	<b>13.08</b> $\pm$ 0.41	<b>37.20</b>	<b>35.96</b> $\pm$ 0.79	<b>13.77</b>	<b>13.50</b> $\pm$ 0.37	<b>37.24</b>	<b>36.60</b> $\pm$ 0.72
<b>Mouthing</b>	256x512	12.86	12.40 $\pm$ 0.42	34.13	34.63 $\pm$ 0.64	13.25	12.34 $\pm$ 0.51	35.53	34.83 $\pm$ 0.47
<b>Pose</b>	32x64	9.64	8.91 $\pm$ 0.42	30.27	29.92 $\pm$ 0.67	9.64	8.55 $\pm$ 0.62	30.03	29.13 $\pm$ 0.79
<b>Pose</b>	64x128	10.64	10.28 $\pm$ 0.26	31.06	31.31 $\pm$ 0.47	9.97	9.88 $\pm$ 0.18	29.63	30.44 $\pm$ 0.60
<b>Pose</b>	128x256	<b>11.02</b>	<b>10.52</b> $\pm$ 0.31	<b>32.22</b>	<b>31.85</b> $\pm$ 0.42	<b>10.26</b>	<b>10.03</b> $\pm$ 0.46	<b>30.44</b>	<b>30.79</b> $\pm$ 0.82
<b>Pose</b>	256x512	10.06	9.50 $\pm$ 0.51	31.03	30.11 $\pm$ 0.75	9.51	8.62 $\pm$ 0.70	29.92	28.65 $\pm$ 0.99
<b>Gloss</b>	32x64	17.21	16.03 $\pm$ 0.49	42.20	41.26 $\pm$ 0.64	15.45	15.68 $\pm$ 0.43	41.21	40.71 $\pm$ 0.42
<b>Gloss</b>	64x128	18.50	18.16 $\pm$ 0.23	44.99	43.87 $\pm$ 0.68	18.14	17.89 $\pm$ 0.56	43.57	43.02 $\pm$ 0.69
<b>Gloss</b>	128x256	19.43	<b>19.14</b> $\pm$ 0.36	<b>46.10</b>	<b>45.17</b> $\pm$ 0.63	19.52	<b>19.08</b> $\pm$ 0.48	<b>45.32</b>	<b>44.52</b> $\pm$ 0.80
<b>Gloss</b>	256x512	<b>19.52</b>	18.36 $\pm$ 0.50	45.97	44.16 $\pm$ 0.79	<b>19.61</b>	18.60 $\pm$ 0.63	45.29	43.92 $\pm$ 0.98

Table 7.2: Single channel SLT baselines using different network architectures,

Fusion Channels	HSXFF	Dev Set		Test Set						
		BLEU-4	ROUGE	BLEU-4	ROUGE					
		Best	mean $\pm$ std	-	mean $\pm$ std	-	mean $\pm$ std			
-	Best Hand	128x256	16.32	15.91 $\pm$ 0.43	41.87	41.08 $\pm$ 0.73	16.76	16.02 $\pm$ 0.88	41.85	40.67 $\pm$ 0.93
-	Best Mouthing	128x256	13.74	13.08 $\pm$ 0.41	37.20	35.96 $\pm$ 0.79	13.77	13.50 $\pm$ 0.37	37.2	36.60 $\pm$ 0.72
-	Best Pose	128x256	11.02	10.52 $\pm$ 0.31	32.22	31.85 $\pm$ 0.42	10.26	10.03 $\pm$ 0.46	30.44	30.79 $\pm$ 0.82
-	Best Gloss	128x256	19.43	19.14 $\pm$ 0.36	46.10	45.17 $\pm$ 0.63	19.52	19.08 $\pm$ 0.48	45.32	44.52 $\pm$ 0.80
Early	H + M	2*(128x256)	<b>17.25</b>	<b>16.73</b> $\pm$ 0.57	<b>42.04</b>	<b>41.72</b> $\pm$ 0.61	<b>17.37</b>	<b>16.73</b> $\pm$ 0.82	<b>42.35</b>	<b>41.76</b> $\pm$ 0.80
Early	H + P	2*(128x256)	16.17	15.70 $\pm$ 0.32	40.51	40.28 $\pm$ 0.46	15.75	15.83 $\pm$ 0.30	40.54	40.34 $\pm$ 0.70
Early	M + P	2*(128x256)	13.57	12.91 $\pm$ 0.30	36.43	35.88 $\pm$ 0.40	13.23	13.02 $\pm$ 0.42	35.66	36.01 $\pm$ 0.72
Early	H + M + P	3*(128x256)	15.69	15.08 $\pm$ 0.55	39.78	39.38 $\pm$ 0.73	15.19	15.19 $\pm$ 0.49	39.99	39.43 $\pm$ 0.80
Late	H + M	2*(128x256)	<b>17.03</b>	<b>16.36</b> $\pm$ 0.48	41.69	<b>41.58</b> $\pm$ 0.79	16.81	<b>16.67</b> $\pm$ 0.49	41.69	<b>41.69</b> $\pm$ 0.54
Late	H + P	2*(128x256)	16.61	16.16 $\pm$ 0.33	41.54	41.18 $\pm$ 0.58	15.90	16.07 $\pm$ 0.76	40.81	40.50 $\pm$ 0.99
Late	M + P	2*(128x256)	14.22	13.55 $\pm$ 0.31	36.44	37.03 $\pm$ 0.64	14.11	13.65 $\pm$ 0.49	36.25	36.95 $\pm$ 0.65
Late	H + M + P	3*(128x256)	17.00	16.35 $\pm$ 0.38	<b>42.09</b>	41.29 $\pm$ 0.42	<b>16.95</b>	16.50 $\pm$ 0.47	<b>42.12</b>	41.53 $\pm$ 0.52

Table 7.3: SLT performance of early and late channel fusion approaches.

---

method against other fusion options, gloss based translation and other state-of-the-art methods. As with other fusion experiments, we examine all possible fusion combinations. In addition to using the 128x256 network setup, we also evaluate having a larger network to see if the additional anchoring losses help with over-fitting by regularizing the translation loss.

As can be seen in the first row of Table 7.4, while using the same number of parameters as the early and late fusion setups, our proposed *Multi-Channel Transformer* approach outperforms both configurations. However, doubling the network size does effect the direct application of multi-channel attention negatively. To counteract this issue and to examine the effects of the anchoring loss, we run experiments with both 128x256 and 256x512 setups. We normalize our losses on the sequence level instead of token level and we set the anchoring loss weight,  $\lambda_A$ , to 0.15 to counteract different source (video) and target (sentence) sequence lengths. Using the anchoring losses not only improves the performance of the 128x256 models but also allows the 256x512 networks to achieve similar translation performance to using gloss features. We believe this is due to two main factors. Firstly, the anchoring loss forces the encoder channels to preserve the channel specific information while being contextually modeled against other articulators. Secondly, it acts as a regularizer for the translation loss and counteracts the over-fitting previously discussed.

Furthermore, we report state-of-the art performance using both the gloss single channel model and the proposed anchored multi-channeled transformer architecture, surpassing the performance of methods such as Sign2Gloss→Gloss2Text [24], which are heavily reliant on gloss annotation.

## 7.4 Qualitative Examples

In this section we share translation examples generated by our best performing model. As the ground truth spoken language annotations of the PHOENIX14T dataset are in German, we share both the original German translations and their equivalent word-by-word translations in English. As can be seen in Table 7.5 and Table 7.6, we

Channels	Anchoring Loss	HSxFF	Dev Set		Test Set					
			BLEU-4	ROUGE	BLEU-4	ROUGE				
			Best	mean $\pm$ std	-	mean $\pm$ std	-	mean $\pm$ std		
H+M	X	2*(128x256)	17.71	16.97 $\pm$ 0.53	43.43	42.02 $\pm$ 0.86	17.72	17.19 $\pm$ 0.73	42.70	41.95 $\pm$ 0.85
H+P	X	2*(128x256)	17.20	16.36 $\pm$ 0.58	42.15	41.23 $\pm$ 0.46	16.41	16.25 $\pm$ 0.66	40.56	40.87 $\pm$ 0.67
M+P	X	2*(128x256)	14.17	13.50 $\pm$ 0.40	36.82	36.62 $\pm$ 0.52	13.43	13.93 $\pm$ 0.44	37.03	37.43 $\pm$ 0.65
H+M+P	X	3*(128x256)	17.98	16.89 $\pm$ 0.59	44.01	41.85 $\pm$ 0.93	17.15	16.85 $\pm$ 0.65	42.38	41.83 $\pm$ 0.85
H+M	X	2*(256x512)	15.95	15.46 $\pm$ 0.34	41.01	40.31 $\pm$ 0.50	15.87	15.80 $\pm$ 0.36	40.30	40.40 $\pm$ 0.70
H+P	X	2*(256x512)	15.41	14.95 $\pm$ 0.33	40.10	39.22 $\pm$ 0.53	15.91	15.14 $\pm$ 0.59	40.24	39.11 $\pm$ 0.69
M+P	X	2*(256x512)	13.39	12.60 $\pm$ 0.49	35.70	35.01 $\pm$ 0.73	13.38	12.74 $\pm$ 0.48	36.89	35.39 $\pm$ 0.79
H+M+P	X	3*(256x512)	15.87	14.97 $\pm$ 0.51	40.53	39.65 $\pm$ 0.79	16.02	15.17 $\pm$ 0.81	40.15	39.61 $\pm$ 1.12
H+M	✓	2*(128x256)	18.52	17.93 $\pm$ 0.39	44.56	43.25 $\pm$ 0.57	17.93	17.76 $\pm$ 0.49	43.21	42.91 $\pm$ 0.49
H+P	✓	2*(128x256)	17.70	16.53 $\pm$ 0.67	43.19	41.26 $\pm$ 0.99	16.93	16.41 $\pm$ 0.48	42.62	40.80 $\pm$ 0.82
M+P	✓	2*(128x256)	15.14	14.60 $\pm$ 0.32	38.57	38.45 $\pm$ 0.45	15.32	15.05 $\pm$ 0.72	38.47	38.66 $\pm$ 0.76
H+M+P	✓	3*(128x256)	18.80	17.81 $\pm$ 0.68	44.24	43.17 $\pm$ 0.81	18.30	17.75 $\pm$ 0.58	43.65	42.90 $\pm$ 0.62
H+M	✓	2*(256x512)	19.05	18.07 $\pm$ 0.44	45.04	43.76 $\pm$ 0.82	<b>19.21</b>	17.71 $\pm$ 0.72	<b>45.05</b>	43.29 $\pm$ 0.99
H+P	✓	2*(256x512)	16.80	16.29 $\pm$ 0.36	41.15	40.86 $\pm$ 0.42	16.68	16.29 $\pm$ 0.48	41.34	40.68 $\pm$ 0.76
M+P	✓	2*(256x512)	15.14	14.60 $\pm$ 0.26	39.45	38.47 $\pm$ 0.62	15.36	15.13 $\pm$ 0.44	40.06	39.09 $\pm$ 0.62
H+M+P	✓	3*(256x512)	<b>19.51</b>	<b>18.66</b> $\pm$ 0.52	<b>45.90</b>	<b>44.30</b> $\pm$ 0.92	18.51	<b>18.31</b> $\pm$ 0.57	43.57	<b>43.75</b> $\pm$ 0.63
Gloss	-	128x256	19.43	<b>19.14</b> $\pm$ 0.36	<b>46.10</b>	<b>45.17</b> $\pm$ 0.63	19.52	<b>19.08</b> $\pm$ 0.48	<b>45.32</b>	<b>44.52</b> $\pm$ 0.80
Gloss	-	256x512	<b>19.52</b>	18.36 $\pm$ 0.50	45.97	44.16 $\pm$ 0.79	<b>19.61</b>	18.60 $\pm$ 0.63	45.29	43.92 $\pm$ 0.98
Cangoz <i>et al.</i> [24]	-	-	18.40	-	44.14	-	18.13	-	43.80	-
Orbay <i>et al.</i> [129]	-	-	-	-	-	-	14.56	-	38.05	-

Table 7.4: Multi-channel Transformer based multi-articulatory SLT results.

also categorize the results into three categories, namely *Good*, *Mediocre* and *Poor* translations, to give further insight to the reader on the limitations of the current approach.

We categorize translations as *Good* or *Mediocre* when the produced sentences convey the same or similar information as the reference sentences. These examples follow the standard grammar with few exceptions. We classify translations as *Poor* when the model fails to understand and translate the conveyed information in sign videos. Most of these examples contain repetitions. In some cases, the model is not able distinguish some sign glosses from another, such as named entities like locations or numbers which occur in limited contexts in the training data. One way to address this issue might be to utilize pretrained spoken language models to improve the produced translations.

Good Translations:




 <p>Reference: und nun die wettvorhersage für morgen diensttag den ersten februar . ( and now the weather forecast for tomorrow tuesday the first of february . )</p> <p>Ours: und nun die wettvorhersage für morgen diensttag den ersten februar . ( and now the weather forecast for tomorrow tuesday the first of february . )</p>
 <p>Reference: der sorgt wieder für wolken die regen im bergland auch schnee bringen . ( it provides clouds again and the rain in the mountains also brings snow . )</p> <p>Ours: im übrigen land fällt gebietsweise regen im bergland auch schnee . ( in the rest of the country there is rain in the mountains and snow in some areas . )</p>
 <p>Reference: die neue woche beginnt wechselhaft und kühler . ( the next week starts unpredictable and colder . )</p> <p>Ours: auch am montag wechselhaft und deutlich kühler . ( also on monday unpredictable and significantly colder . )</p>

Table 7.5: Spoken language translations produced by our best Multi-Channel Transformer model - Good Translation Examples.

We also share and compare qualitative translation examples obtained from the mod-







Mediocre Translations:	
	<p>Reference: ab sonntag wird es wieder milder dabei gibt es viele wolken zeitweise fällt regen im nordwesten windig . (from sunday on it will be more mild with many clouds partly rain in the northwest windy .)</p> <p>Ours: am sonntag mehr wolken als sonne hier und da regen im westen ist es windig . (on sunday more clouds than sun from time to time rain in the west windy .)</p>
	<p>Reference: im osten und südosten auch schnee oder schneeregen . ( in the east and south-east also snow or sleet . )</p> <p>Ours: im südosten schnee oder schnee . ( snow or snow in the south-east . )</p>
	<p>Reference: westlich des rheins und im nordosten bleibt es meist trocken . ( west of the rhine and in the northeast it remains mostly dry . )</p> <p>Ours: im westen und südwesten bleibt es noch im nordosten trocken . ( in the west and southwest it remains still dry in the northeast . )</p>
Poor Translations:	
	<p>Reference: deutschland liegt morgen unter hochdruckeinfluss der die wolken weitgehend vertreibt . ( germany will be under the influence of high pressure tomorrow which will largely dispel the clouds . )</p> <p>Ours: deutschland liegt morgen über deutschland nach deutschland . ( germany is tomorrow over germany to germany . )</p>
	<p>Reference: am freitag insgesamt viele wolken die regen bringen . ( on friday overall many clouds bringing rain . )</p> <p>Ours: am freitag gibt es am freitag viele wolken . ( on friday there are on friday many clouds )</p>
	<p>Reference: dazu weht ein starker wind vor allen dingen wieder über vorpommern aus südost . ( in addition a strong wind blows before all things again over vorpommern from southeast . )</p> <p>Ours: es weht ein kräftiger nordostwind . ( a strong north-easterly wind is blowing . )</p>

Table 7.6: Spoken language translations produced by our best Multi-Channel Transformer model - Mediocre and Poor Translation Examples.

els presented in different chapters of this thesis, namely the RNN-based end-to-end Sign2Text model from Chapter 5, the transformer-based state-of-the-art multi-task learn-

---

ing approach Sign2(Gloss+Text) from Chapter 6, and the multi-channel transformers presented in this chapter. As with the quantitative results, overall, transformer-based models yield more accurate translations than RNN-based models (See Examples 1, 2 and 3 in Table 7.7). However, in the case of Example 3, all three models fail to fully translate the context of the source sign language video, suggesting that there is still further room for improvement. Another interesting behaviour of transformer based models is their capability to generate contextually similar sentences using different vocabulary. In Example 4, the multi-channel transformer model produced the German translation using “zeitweise” instead of “ab un an”, both of which translate to “time to time” in this context. However, our model is penalized for producing the word “zeitweise”, as it is not present in the reference translation. This suggests using a single reference translation limits the accuracy of the evaluation metric, hence we recommend future datasets to include multiple acceptable reference translations.

## 7.5 Closing Remarks

In contrast to most of the previous vision based sign language research, which mainly focused on manual features, in this chapter we model sign languages in a holistic manner, using both manual and non-manual features. We utilize hand, mouthing and upper body pose representations and realize the first multi-articulatory video-to-text SLT. We propose a novel multi-channel architecture, named *Multi-Channel Transformers*, for sequence-to-sequence tasks, such as SLT, where the source information is embedded over different channels. This approach allows both the inter and intra relationship between different asynchronous sign channels to be modelled within the transformer network itself. We also introduce a channel anchoring loss to help our models to preserve channel specific information while also regulating training against overfitting.

We evaluate our approach on the PHOENIX14T dataset. When compared against other fusion techniques, namely late and early, the proposed multi-channel transformer approach achieve superior translation performance. More importantly, we report on par or better translation performance compared to previous work which requires labor intensive gloss annotations.





	
<b>Example 1 - Reference:</b>	und nun die wettvorhersage für morgen samstag den zweiten april . ( and now the weatherforecast for tomorrow saturday the second april . )
Sign2Text (RNN-Based) Chapter 5	und nun die wettvorhersage für morgen freitag den sechszwanzigsten märz . ( and now the weatherforecast for tomorrow friday the twentieth march . )
Sign2(Gloss+Text) (Transformers) Chapter 6	und nun die wettvorhersage für morgen samstag den ersten april . ( and now the weatherforecast for tomorrow saturday the first april . )
Multi-Channel Transformers Chapter 7	und nun die wettvorhersage für morgen samstag den ersten märz . ( and now the weatherforecast for tomorrow saturday the first march . )
	
<b>Example 2 - Reference:</b>	die neue woche beginnt wechselhaft und kühler . ( the next week starts unpredictable and cooler . )
Sign2Text (RNN-Based) Chapter 5	am montag überall wechselhaft und kühler . ( on monday everywhere unpredictable and cooler . )
Sign2(Gloss+Text) (Transformers) Chapter 6	die neue woche beginnt wechselhaft und wieder kühler . ( the next week starts unpredictable and again cooler . )
Multi-Channel Transformers Chapter 7	auch am montag wechselhaft und deutlich kühler . ( also on monday unpredictable and significantly colder . )
	
<b>Example 3 - Reference:</b>	im süden und südwesten gebietsweise regen sonst recht freundlich . ( in the south and southwest partly rain otherwise quite friendly . )
Sign2Text (RNN-Based) Chapter 5	von der südhälfte beginnt es vielerorts . ( from the southpart it starts in many places . )
Sign2(Gloss+Text) (Transformers) Chapter 6	im südwesten regnet es zum teil kräftig . ( in the southwest partly heavy rain . )
Multi-Channel Transformers Chapter 7	im südwesten etwas regen richtung nord freundlich . ( in the southwest a little rain towards the north friendly . )
	
<b>Example 4 - Reference:</b>	am sonntag ab und an regenschauer teilweise auch gewitter . ( on sunday time to time rainshower partly also thunderstorm . )
Sign2Text (RNN-Based) Chapter 5	am sonntag sonne und wolken und gewitter . ( on sunday sun and clouds and thunderstorm . )
Sign2(Gloss+Text) (Transformers) Chapter 6	am sonntag muss in stellenweise mit regen oder gewittern gerechnet werden . ( on sunday rain or thunderstrom is to be expected in places . )
Multi-Channel Transformers Chapter 7	und am sonntag fällt noch zeitweise regen . ( and on sunday rain falls from time to time . )

Table 7.7: Spoken language translations produced by RNN-based (Chapter 5) and transformer-based (Chapter 6 and Chapter 7) approaches presented in this thesis.

Now we have broken the dependency upon gloss information, future work will be to scale learning to larger dataset, where gloss information is not available such as



---

broadcast footage. It would also be interesting to incorporate additional channels to our model, such as eye-brows, mouth gestures and facial expressions. For channels where there are no previous classifiers for anchoring losses, one can either use limited linguistic corpora with high quality annotations or learn classes in a data driven manner from large scale broadcast footage.



## Chapter 8

# Conclusions and Future Work

In this thesis we tackled the challenging Sign Language Translation (SLT) task, with a goal of producing spoken/written language interpretations of the information conveyed in continuous sign language videos. Using the RWTH-PHOENIX-Weather-2014T Dataset (PHOENIX14T), the first and currently the only publicly available continuous SLT dataset, we developed several evaluation protocols to underpin future research in the field (See Chapter 4).

In chapter 5, using PHOENIX14T, we realized the first SLT approach, *Neural Sign Language Translation* [24]. We combined Convolutional Neural Networks (CNNs) with attention-based encoder-decoder models and trained models in an end-to-end manner. Although these models were able to generate meaningful translations, their performance was drastically worse than other Neural Machine Translation (NMT) baselines (9.58 BLEU-4). We think this was due to SLT being a challenging task, which requires recognizing and understanding of spatio-temporal linguistic constructs, as well as the models' limited ability to generalize from the currently available datasets. To ease the translation problem, we proposed dividing the pipeline into two stages, namely recognition and translation. We utilized state-of-the-art Continuous Sign Language Recognition (CSLR) models to predict gloss sequences from continuous sign language videos. We then trained a translation model from these gloss predictions to spoken/written language sentences. This approach yielded much higher translation accuracy (18.13 BLEU-4), indicating the benefits of using intermediate gloss level supervision.

Translating from predicted glosses imposes an information bottleneck as they do not fully encapsulate sign languages. However, as empirically shown in Chapter 5, their use improve the translation performance. To utilize gloss supervision, without limiting the translation, we reformulated the previous 2-stage SLT approach as a multi-task problem in chapter 6. We utilized state-of-the-art transformer networks [187] and proposed *Sign Language Transformers* [27]. We jointly trained our models to perform CSLR and SLT and reported significantly improved translation performance over models trained on single tasks.

We need more parallel datasets to realize reliable large scale sign language translation systems. However, our reliance on sign glosses is one of the limiting factors for us to exploit available broadcast sign language interpretations. To address this issue, in chapter 7 we took inspiration from sub-unit based sign language recognition literature and proposed modelling SLT based on sign articulators instead of glosses. We utilized hand shape, mouthing and upper body pose representations to incorporate both manual and non-manual features of the sign to our translation networks. We proposed a novel *Multi-Channel Transformer* architecture that supports multi-channel, asynchronous, sequence-to-sequence learning. We used the proposed approach to realize the first successful approach to multi-articulatory SLT, which models the inter and intra relationship of manual and non-manual channels. Our models achieved on par or better translation performance against several single channel and fusion baselines, including networks that utilized gloss supervision. These results suggested that we may not need extensive gloss annotations to realize large scale SLT.

## 8.1 Possible Future Work

Although there has been major progress in the field, such as realizing continuous Sign Language Recognition (SLR) and SLT using weakly annotated data, there are still several challenges to overcome. One of these challenges is Signer Independent recognition and translation. In their recent paper, Koller *et al.* [102] evaluated their methods' performance both on signer dependent and independent setups and noted that their methods performed drastically worse in the signer independent scenario.

---

One approach to model signs in a user independent manner could be to focus on more explicit modeling of sign linguistics and their building blocks, commonly known as subunits [173]. To do so, computer vision researchers can collaborate with sign linguists to gain access to high-quality annotated data [78, 153]. However, there are relatively few linguistic resources at the scale required for machine learning on the fundamental building blocks and grammar of sign e.g. the combined use of different articulators, sign syntax and grammatical constructs such as the use of syntactic/topographic space and placement. One way to tackle this issue would be to utilize an iterative refinement workflow. Computer vision researchers can train systems with the available data to generate automatic annotations which would then be corrected by sign linguists to train better systems, thus effectively increasing the amount of high-quality annotations. These subunit based representations can then be combined using our proposed Multi-Channel Transformer approach to train signer-independent SLT networks.

There is also the issue of the racial bias that is being learned due to the dataset demographics. For example, a system that was trained on images of PHOENIX datasets [64, 65], even if it was trained on subunit level, would not be able to generalize to all sign language users, as the dataset only contain adult Caucasian signers. One way to overcome such biases could be to change the input domains from images to canonical body models [111]. By using such body models as a pre-processing step, we can decouple the signers' physical attributes from their body, face and hand poses. We can then use these representations to train our models in a signer independent manner.

Besides CSLR and SLT, there are various other applications of computer vision based sign language research. Such applications include, but are not limited to, automatic sign language assessment and production. The field of automatic Sign Language Assessment (SLA) is still in its infancy. Similar to the historical progress of SLR, SLA has started by focusing on isolated signs and manual features. However, with the availability of datasets, such as the SMILE Dataset [57], the field will gain more attention from both computational linguists and computer vision researchers. Following the footsteps of SLR, the next step for SLA would be to move to the continuous domain, where non-manual features are more prominent. To achieve continuous SLA, researchers will require more data. However, annotating continuous sequences with

both manual and non-manual feature is a laborious task and considerably more complex than working on isolated signs. One way to overcome this problem might be to exploit the recent developments in the field of image synthesis using adversarial training [71], and artificially increase the training data.

Another interesting line of study is sign language production, which has been mainly dominated by avatar based methods [62]. However, Stoll *et al.* [169] recently combined approaches from the fields of NMT and Generative Adversarial Networks (GANs) to generate videos of sign sequences translated from text without the use of avatars. Current state-of-the-art in the field of image synthesis is able to generate images of humans in unseen poses with promising performance [115, 159] and preliminary work [170] in sign language production has already demonstrated its utility in the generation of signs. Using these methods, researchers could generate vast amounts of data with variability from prototypical signs. Another solution could lie in utilizing the currently available avatar-based sign language production systems and combining them with recent machine learning approaches [157], which can transform simulated avatars into real life looking images.

Considering all the recent developments and open questions, computer vision based sign language recognition is a fruitful research field, attracting more interest as it progresses. Being at the conjunction of language and vision, sign language research still offers challenges to be solved which can enhance the lives of Deaf communities, and it is an exciting time to work in this field.

# Bibliography

- [1] Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, Manjunath Kudlur, Josh Levenberg, Rajat Monga, Sherry Moore, Derek G. Murray, Benoit Steiner, Paul Tucker, Vijay Vasudevan, Pete Warden, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. Tensorflow: A System for Large-scale Machine Learning. In *Proceedings of the 12th Symposium on Operating Systems Design and Implementation*, 2016.
- [2] Ossama Abdel-Hamid, Abdel-rahman Mohamed, Hui Jiang, Li Deng, Gerald Penn, and Dong Yu. Convolutional Neural Networks for Speech Recognition. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 22(10), 2014.
- [3] Triantafyllos Afouras, Joon Son Chung, Andrew Senior, Oriol Vinyals, and Andrew Senior. Deep Audio-visual Speech Recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2018.
- [4] Ahmed Ali and Steve Renals. Word Error Rate Estimation for Speech Recognition: e-WER. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (ACL)*, 2018.
- [5] Dario Amodei, Rishita Anubhai, Eric Battenberg, Carl Case, Jared Casper, Bryan Catanzaro, Jingdong Chen, Mike Chrzanowski, Adam Coates, Greg Diamos, et al. Deep Speech 2: End-to-end Speech Recognition in English and Mandarin. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2016.

- 
- [6] Stanislaw Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C Lawrence Zitnick, and Devi Parikh. VQA: Visual Question Answering. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2015.
- [7] E. Antonakos, V. Pitsikalis, I. Rodomagoulakis, and P. Maragos. Unsupervised Classification of Extreme Facial Events Using Active Appearance Models Tracking for Sign Language Videos. In *Proceedings of the IEEE International Conference on Image Processing (ICIP)*, 2012.
- [8] Oya Aran. *Vision-based Sign Language Recognition: Modeling and Recognizing Isolated Signs with Manual and Non-Manual Components*. PhD thesis, Bogazici University, 2008.
- [9] Oya Aran, Ismail Ari, Lale Akarun, Bulent Sankur, Alexandre Benoit, Alice Caplier, Pavel Campr, Ana Huerta Carrillo, and Francois-Xavier Fanard. Sign-Tutor: An Interactive System for Sign Language Tutoring. *IEEE MultiMedia*, 16(1), 2009.
- [10] Yannis M Assael, Brendan Shillingford, Shimon Whiteson, and Nando De Freitas. Lipnet: End-to-end Sentence-level Lipreading. In *GPU Technology Conference*, 2017.
- [11] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural Machine Translation by Jointly Learning to Align and Translate. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2015.
- [12] Herbert Bay, Tinne Tuytelaars, and Luc Van Gool. Surf: Speeded Up Robust Features. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2006.
- [13] Ursula Bellugi and Susan Fischer. A comparison of sign language and spoken language. *Cognition*, 1(2), 1972.
- [14] Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. Enrich-



- 
- ing Word Vectors with Subword Information. *Transactions of the Association for Computational Linguistics (ACL)*, 5, 2017.
- [15] Mark Borg and Kenneth P Camilleri. Sign Language Detection “in the Wild” with Recurrent Neural Networks. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2019.
- [16] Léon Bottou. Large-scale Machine Learning with Stochastic Gradient Descent. In *Proceedings of the International Conference on Computational Statistics (COMPSTAT)*, 2010.
- [17] Penny Boyes-Braem and Rachel Sutton-Spence. *The Hands are the Head of the Mouth: The Mouth as Articulator in Sign Languages*. Gallaudet University Press, 2001.
- [18] Diane Brentari. *Sign Language Phonology*. Cambridge University Press, 2019.
- [19] Patrick Buehler, Andrew Zisserman, and Mark Everingham. Learning Sign Language by Watching TV (using weakly aligned subtitles). In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009.
- [20] Jan Bungeroth and Hermann Ney. Statistical Sign Language Translation. In *Proceedings of the Workshop on Representation and Processing of Sign Languages at International Conference on Language Resources and Evaluation (LREC)*, 2004.
- [21] Necati Cihan Camgoz, Simon Hadfield, and Richard Bowden. Particle Filter based Probabilistic Forced Alignment for Continuous Gesture Recognition. In *Proceedings of the IEEE International Conference on Computer Vision Workshops (ICCVW)*, 2017.
- [22] Necati Cihan Camgoz, Simon Hadfield, Oscar Koller, and Richard Bowden. Using Convolutional 3D Neural Networks for User-Independent Continuous Gesture Recognition. In *Proceedings of the IEEE International Conference on Pattern Recognition Workshops (ICPRW)*, 2016.

- 
- [23] Necati Cihan Camgoz, Simon Hadfield, Oscar Koller, and Richard Bowden. SubUNets: End-to-end Hand Shape and Continuous Sign Language Recognition. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2017.
- [24] Necati Cihan Camgoz, Simon Hadfield, Oscar Koller, Hermann Ney, and Richard Bowden. Neural Sign Language Translation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.
- [25] Necati Cihan Camgoz, Ahmet Alp Kindiroglu, and Lale Akarun. Sign Language Recognition for Assisting the Deaf in Hospitals. In *Proceedings of the International Workshop on Human Behavior Understanding (HBU)*, 2016.
- [26] Necati Cihan Camgoz, Oscar Koller, Simon Hadfield, and Richard Bowden. Multi-channel Transformers for Multi-articulatory Sign Language Translation. In *16th European Conference on Computer Vision Workshops (ECCVW)*, 2020.
- [27] Necati Cihan Camgoz, Oscar Koller, Simon Hadfield, and Richard Bowden. Sign Language Transformers: Joint End-to-end Sign Language Recognition and Translation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.
- [28] Z Cao, G Martinez Hidalgo, T Simon, SE Wei, and YA Sheikh. OpenPose: Realtime Multi-Person 2D Pose Estimation using Part Affinity Fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2019.
- [29] Zhe Cao, Tomas Simon, Shih-En Wei, and Yaser Sheikh. Realtime Multi-Person 2D Pose Estimation using Part Affinity Fields. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.
- [30] George Caridakis, Stylianos Asteriadis, and Kostas Karpouzis. Non-Manual Cues in Automatic Sign Language Recognition. *Personal and ubiquitous computing*, 18(1), 2014.
- [31] Xiujuan Chai, Guang Li, Yushun Lin, Zhihao Xu, Yili Tang, Xilin Chen, and Ming Zhou. Sign Language Recognition and Translation with Kinect. In

- 
- Proceedings of the International Conference on Automatic Face and Gesture Recognition (FG)*, 2013.
- [32] Xiujuan Chai, Zhipeng Liu, Fang Yin, Zhuang Liu, and Xilin Chen. Two Streams Recurrent Neural Networks for Large-Scale Continuous Gesture Recognition. In *Proceedings of the International Conference on Pattern Recognition Workshops (ICPRW)*, 2016.
- [33] Xiujuan Chai, Hanjie Wang, and Xilin Chen. The DEVISIGN Large Vocabulary of Chinese Sign Language Database and Baseline Evaluations. *Technical Report VIPL-TR-14-SLR-001*, 2014.
- [34] James Charles, Tomas Pfister, Mark Everingham, and Andrew Zisserman. Automatic and Efficient Human Pose Estimation for Sign Language Videos. *International Journal of Computer Vision (IJCV)*, 110(1), 2014.
- [35] Neva Cherniavsky, Richard E Ladner, and Eve A Riskin. Activity Detection in Conversational Sign Language Video for Mobile Telecommunication. In *Proceedings of the International Conference on Automatic Face and Gesture Recognition (FG)*, 2008.
- [36] Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning Phrase Representations Using RNN Encoder-Decoder for Statistical Machine Translation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2014.
- [37] Joon Son Chung, Andrew Senior, Oriol Vinyals, and Andrew Zisserman. Lip Reading Sentences in the Wild. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.
- [38] Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling. In *Proceedings of the Advances in Neural Information Processing Systems Workshops (NIPSW)*, 2014.

- 
- [39] Kevin Clark, Minh-Thang Luong, Quoc V Le, and Christopher D Manning. ELECTRA: Pre-training Text Encoders as Discriminators Rather Than Generators. In *International Conference on Learning Representations*, 2020.
- [40] Helen Cooper and Richard Bowden. Sign Language Recognition using Boosted Volumetric Features. In *Proceedings of the IAPR International Conference on Machine Vision Applications (MVA)*, 2007.
- [41] Helen Cooper and Richard Bowden. Learning Signs from Subtitles: A Weakly Supervised Approach to Sign Language Recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009.
- [42] Helen Cooper and Richard Bowden. Sign Language Recognition using Linguistically Derived Sub-Units. In *Proceedings of the 4th Workshop on the Representation and Processing of Sign Languages: Corpora and Sign Language Technologies*, 2010.
- [43] Helen Cooper, Brian Holt, and Richard Bowden. Sign Language Recognition. In *Visual Analysis of Humans*. Springer, 2011.
- [44] Helen Cooper, Eng-Jon Ong, Nicolas Pugeault, and Richard Bowden. Sign Language Recognition using Sub-units. *Journal of Machine Learning Research (JMLR)*, 13, 2012.
- [45] Helen Cooper, Nicolas Pugeault, and Richard Bowden. Reading the Signs: A Video Based Sign Dictionary. In *Proceedings of the IEEE International Conference on Computer Vision Workshops (ICCVW)*, 2011.
- [46] Kearsy Cormier, Neil Fox, Bencie Woll, Andrew Zisserman, Necati Cihan Camgoz, and Richard Bowden. ExTOL: Automatic recognition of British Sign Language using the BSL Corpus. In *Proceedings of the Sign Language Translation and Avatar Technology (SLTAT)*, 2019.
- [47] Stephen Cox, Michael Lincoln, Judy Tryggvason, Melanie Nakisa, Mark Wells, Marcus Tutt, and Sanja Abbott. TESSA, a System to Aid Communication with

- 
- Deaf People. In *Proceedings of the ACM International Conference on Assistive Technologies*, 2002.
- [48] Runpeng Cui, Hu Liu, and Changshui Zhang. Recurrent Convolutional Neural Networks for Continuous Sign Language Recognition by Staged Optimization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.
- [49] Runpeng Cui, Hu Liu, and Changshui Zhang. A Deep Neural Framework for Continuous Sign Language Recognition by Iterative Training. *IEEE Transactions on Multimedia*, 2019.
- [50] Zihang Dai, Zhilin Yang, Yiming Yang, William W Cohen, Jaime Carbonell, Quoc V Le, and Ruslan Salakhutdinov. Transformer-XL: Attentive Language Models beyond a Fixed-length Context. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, 2019.
- [51] Navneet Dalal and Bill Triggs. Histograms of Oriented Gradients for Human Detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2005.
- [52] Pieter-Tjerk De Boer, Dirk P Kroese, Shie Mannor, and Reuven Y Rubinstein. A Tutorial on the Cross-Entropy Method. *Annals of Operations Research*, 134(1), 2005.
- [53] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Fei-Fei Li. ImageNet: A Large-scale Hierarchical Image Database. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009.
- [54] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics (ACL): Human Language Technologies*, 2019.
- [55] Jeffrey Donahue, Lisa Anne Hendricks, Sergio Guadarrama, Marcus Rohrbach, Subhashini Venugopalan, Kate Saenko, and Trevor Darrell. Long-term Recurrent

- 
- Convolutional Networks for Visual Recognition and Description. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.
- [56] Sarah Ebling. *Automatic Translation from German to Synthesized Swiss German Sign Language*. PhD thesis, University of Zurich, 2016.
- [57] Sarah Ebling, Necati Cihan Camgoz, Penny Boyes Braem, Katja Tissi, Sandra Sidler-Miserez, Stephanie Stoll, Simon Hadfield, Tobias Haug, Richard Bowden, Sandrine Tornay, Marzieh Razavi, and Mathew Magimai-Doss. SMILE Swiss German Sign Language Dataset. In *Proceedings of the International Conference on Language Resources and Evaluation (LREC)*, 2018.
- [58] Sarah Ebling, Volker Hegelheimer, Necati Cihan Camgoz, and Richard Bowden. Use of New Technologies in L2 Assessment. In *Handbook of Language Assessment across Modalities*. Oxford University Press, 2020.
- [59] Sarah Ebling and Matt Huenerfauth. Bridging the Gap between Sign Language Machine Translation and Sign Language Animation using Sequence Classification. In *Proceedings of the 6th Workshop on Speech and Language Processing for Assistive Technologies (SPLAT)*, 2015.
- [60] Eleni Efthimiou, Stavroula-Evita Fotinea, Thomas Hanke, John Glauert, Richard Bowden, Annelies Braffort, Christophe Collet, Petros Maragos, and Francois Lefebvre-Albaret. The Dicta-Sign Wiki: Enabling Web Communication for the Deaf. In *Proceedings of the International Conference on Computers for Handicapped Persons*, 2012.
- [61] Ralph Elliott, Helen Cooper, Eng-Jon Ong, John Glauert, Richard Bowden, and F Lefebvre-Albaret. Search-by-Example in Multilingual Sign Language Databases. In *Proceedings of the Sign Language Translation and Avatar Technologies Workshops*, 2011.
- [62] Ralph Elliott, John RW Glauert, JR Kennaway, Ian Marshall, and Eva Safar. Linguistic Modelling and Language Processing Technologies for Avatar-based Sign

- 
- Language Presentation. *Universal Access in the Information Society*, 6(4):375–391, 2008.
- [63] Biyi Fang, Jillian Co, and Mi Zhang. DeepASL: Enabling Ubiquitous and Non-Intrusive Word and Sentence-Level Sign Language Translation. In *Proceedings of the ACM Conference on Embedded Networked Sensor Systems (SenSys)*, 2017.
- [64] Jens Forster, Christoph Schmidt, Thomas Hoyoux, Oscar Koller, Uwe Zelle, Justus H Piater, and Hermann Ney. RWTH-PHOENIX-Weather: A Large Vocabulary Sign Language Recognition and Translation Corpus. In *Proceedings of the International Conference on Language Resources and Evaluation (LREC)*, 2012.
- [65] Jens Forster, Christoph Schmidt, Oscar Koller, Martin Bellgardt, and Hermann Ney. Extensions of the Sign Language Recognition and Translation Corpus RWTH-PHOENIX-Weather. In *Proceedings of the International Conference on Language Resources and Evaluation (LREC)*, 2014.
- [66] Jonathan Frankle and Michael Carbin. The Lottery Ticket Hypothesis: Finding Sparse, Trainable Neural Networks. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2019.
- [67] Kuniyiko Fukushima. Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. *Biological cybernetics*, 36(4), 1980.
- [68] Jonas Gehring, Michael Auli, David Grangier, Denis Yarats, and Yann N Dauphin. Convolutional Sequence to Sequence Learning. In *Proceedings of the ACM International Conference on Machine Learning (ICML)*, 2017.
- [69] Xavier Glorot and Yoshua Bengio. Understanding the Difficulty of Training Deep Feedforward Neural Networks. In *Proceedings of the International Conference on Artificial Intelligence and Statistics*, 2010.
- [70] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT press, 2016.

- 
- [71] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative Adversarial Nets. In *Proceedings of the Advances in Neural Information Processing Systems (NIPS)*, 2014.
- [72] Alex Graves, Santiago Fernández, Faustino Gomez, and Jürgen Schmidhuber. Connectionist Temporal Classification: Labelling Unsegmented Sequence Data with Recurrent Neural Networks. In *Proceedings of the ACM International Conference on Machine Learning (ICML)*, 2006.
- [73] Alex Graves, Marcus Liwicki, Santiago Fernández, Roman Bertolami, Horst Bunke, and Jürgen Schmidhuber. A Novel Connectionist System for Unconstrained Handwriting Recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 31(5), 2009.
- [74] Alex Graves, Abdel-rahman Mohamed, and Geoffrey Hinton. Speech Recognition with Deep Recurrent Neural Networks. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2013.
- [75] Dan Guo, Shuo Wang, Qi Tian, and Meng Wang. Dense Temporal Convolution Network for Sign Language Translation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2019.
- [76] Dan Guo, Wengang Zhou, Houqiang Li, and Meng Wang. Hierarchical LSTM for Sign Language Translation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2018.
- [77] M. Amac Guvensan, A. Oguz Kansiz, Necati Cihan Camgoz, H. Irem Turkmen, A. Gokhan Yavuz, and M. Elif Karsligil. An Energy-Efficient Multi-Tier Architecture for Fall Detection on Smartphones. *Sensors*, 17(7), 2017.
- [78] Thomas Hanke, Lutz König, Sven Wagner, and Silke Matthes. DGS Corpus & Dicta-Sign: The Hamburg Studio Setup. In *Proceedings of the Representation and Processing of Sign Languages: Corpora and Sign Language Technologies*, 2010.



- 
- [79] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep Residual Learning for Image Recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.
- [80] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Identity Mappings in Deep Residual Networks. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2016.
- [81] Gines Hidalgo, Yaadhav Raaj, Haroon Idrees, Donglai Xiang, Hanbyul Joo, Tomas Simon, and Yaser Sheikh. Single-Network Whole-Body Pose Estimation. In *IEEE International Conference on Computer Vision (ICCV)*, 2019.
- [82] Sepp Hochreiter and Jürgen Schmidhuber. Long Short-Term Memory. *Neural Computation*, 9(8), 1997.
- [83] Berthold KP Horn and Brian G Schunck. Determining Optical Flow. *Artificial Intelligence*, 17(1-3), 1981.
- [84] De-An Huang, Li Fei-Fei, and Juan Carlos Niebles. Connectionist Temporal Modeling for Weakly Supervised Action Labeling. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2016.
- [85] Gao Huang, Zhuang Liu, Laurens van der Maaten, and Kilian Q Weinberger. Densely Connected Convolutional Networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.
- [86] Jie Huang, Wengang Zhou, Qilin Zhang, Houqiang Li, and Weiping Li. Video-based Sign Language Recognition without Temporal Segmentation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2018.
- [87] Sergey Ioffe and Christian Szegedy. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2015.
- [88] Melvin Johnson, Mike Schuster, Quoc V Le, Maxim Krikun, Yonghui Wu, Zhifeng Chen, Nikhil Thorat, Fernanda Viégas, Martin Wattenberg, Greg Corrado, et al. Google’s Multilingual Neural Machine Translation System: Enabling

- 
- Zero-Shot Translation. *Transactions of the Association for Computational Linguistics*, 5, 2017.
- [89] Hanbyul Joo, Tomas Simon, and Yaser Sheikh. Total Capture: A 3D Deformation Model for Tracking Faces, Hands, and Bodies. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.
- [90] Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. Bag of Tricks for Efficient Text Classification. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics (ACL)*, 2017.
- [91] Nal Kalchbrenner and Phil Blunsom. Recurrent Continuous Translation Models. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2013.
- [92] Andrej Karpathy, George Toderici, Sanketh Shetty, Thomas Leung, Rahul Sukthankar, and Li Fei-Fei. Large-scale Video Classification with Convolutional Neural Networks. In *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014.
- [93] Shujjat Khan, Donald G Bailey, and Gourab Sen Gupta. Pause detection in continuous sign language. *International Journal of Computer Applications in Technology*, 50, 2014.
- [94] Diederik P. Kingma and Jimmy Ba. Adam: A Method for Stochastic Optimization. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2014.
- [95] Sang-Ki Ko, Chang Jo Kim, Hyedong Jung, and Choongsang Cho. Neural Sign Language Translation based on Human Keypoint Estimation. *Applied Sciences*, 9(13), 2019.
- [96] Oscar Koller, Necati Cihan Camgoz, Richard Bowden, and Hermann Ney. Weakly Supervised Learning with Multi-Stream CNN-LSTM-HMMs to Dis-

- 
- cover Sequential Parallelism in Sign Language Videos. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2019.
- [97] Oscar Koller, Jens Forster, and Hermann Ney. Continuous Sign Language Recognition: Towards Large Vocabulary Statistical Recognition Systems Handling Multiple Signers. *Computer Vision and Image Understanding (CVIU)*, 141, 2015.
- [98] Oscar Koller, Hermann Ney, and Richard Bowden. May the Force be with You: Force-Aligned SignWriting for Automatic Subunit Annotation of Corpora. In *Proceedings of the International Conference on Automatic Face and Gesture Recognition (FG)*, 2013.
- [99] Oscar Koller, Hermann Ney, and Richard Bowden. Read My Lips: Continuous Signer Independent Weakly Supervised Viseme Recognition. In *European Conference on Computer Vision (ECCV)*, 2014.
- [100] Oscar Koller, Hermann Ney, and Richard Bowden. Deep Learning of Mouth Shapes for Sign Language. In *Proceedings of the IEEE International Conference on Computer Vision Workshops (ICCVW)*, 2015.
- [101] Oscar Koller, Hermann Ney, and Richard Bowden. Deep Hand: How to Train a CNN on 1 Million Hand Images When Your Data Is Continuous and Weakly Labelled. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.
- [102] Oscar Koller, Sepehr Zargaran, and Hermann Ney. Re-Sign: Re-Aligned End-to-End Sequence Modelling with Deep Recurrent CNN-HMMs. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.
- [103] Oscar Koller, Sepehr Zargaran, Hermann Ney, and Richard Bowden. Deep Sign: Hybrid CNN-HMM for Continuous Sign Language Recognition. In *Proceedings of the British Machine Vision Conference (BMVC)*, 2016.

- 
- [104] Julia Kreutzer, Joost Bastings, and Stefan Riezler. Joey NMT: A minimalist NMT toolkit for novices. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP): System Demonstrations*, 2019.
- [105] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. ImageNet Classification with Deep Convolutional Neural Networks. In *Proceedings of the Advances in Neural Information Processing Systems (NIPS)*, 2012.
- [106] Taku Kudo and John Richardson. SentencePiece: A Simple and Language Independent Subword Tokenizer and Detokenizer for Neural Text Processing. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2018.
- [107] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based Learning Applied to Document Recognition. *IEEE*, 86(11), 1998.
- [108] Yann LeCun, D Touresky, G Hinton, and T Sejnowski. A Theoretical Framework for Back-Propagation. In *Connectionist Models Summer School*, volume 1, 1988.
- [109] Mu Li, Tong Zhang, Yuqiang Chen, and Alexander J Smola. Efficient Mini-batch Training for Stochastic Optimization. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2014.
- [110] Chin-Yew Lin. ROUGE: A Package for Automatic Evaluation of Summaries. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics, Text Summarization Branches Out Workshop*, 2004.
- [111] Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J Black. Smpl: A skinned multi-person linear model. *ACM Transactions on Graphics (TOG)*, 34(6), 2015.
- [112] David G Lowe. Distinctive Image Features from Scale-Invariant Keypoints. *International Journal of Computer Vision (IJCV)*, 60(2), 2004.
- [113] Minh-Thang Luong, Hieu Pham, and Christopher D Manning. Effective Approaches to Attention-based Neural Machine Translation. In *Proceedings of the*

- 
- Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2015.
- [114] Marcos Luzardo, Matti Karppa, Jorma Laaksonen, and Tommi Jantunen. Head Pose Estimation for Sign Language Video. *Image Analysis*, 2013.
- [115] Liqian Ma, Xu Jia, Qianru Sun, Bernt Schiele, Tinne Tuytelaars, and Luc Van Gool. Pose Guided Person Image Generation. In *Proceedings of the Advances in Neural Information Processing Systems (NIPS)*, 2017.
- [116] Evie Malaia, Joshua D Borneman, and Ronnie B Wilbur. Information Transfer Capacity of Articulators in American Sign Language. *Language and Speech*, 61(1), 2018.
- [117] Charles Malleson, John Collomosse, and Adrian Hilton. Real-time multi-person motion capture from multi-view video and imus. *International Journal of Computer Vision (IJCV)*, 2019.
- [118] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient Estimation of Word Representations in Vector Space. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2013.
- [119] Shunji Mori, Hirobumi Nishida, and Hiromitsu Yamada. *Optical Character Recognition*. John Wiley & Sons, Inc., 1999.
- [120] Sara Morrissey. *Data-driven Machine Translation for Sign Languages*. PhD thesis, Dublin City University, 2008.
- [121] Sara Morrissey, Harold Somers, Robert Smith, Shane Gilchrist, and Sandipan Dandapat. Building a Sign Language Corpus for Use in Machine Translation. In *Proceedings of the Representation and Processing of Sign Languages: Corpora and Sign Language Technologies*, 2010.
- [122] Vinod Nair and Geoffrey E Hinton. Rectified Linear Units Improve Restricted Boltzmann Machines. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2010.

- 
- [123] Graham Neubig. Neural Machine Translation and Sequence-to-Sequence Models: A Tutorial. *arXiv:1703.01619*, 2017.
- [124] Natalia Neverova, Christian Wolf, Graham W Taylor, and Florian Nebout. Multi-scale Deep Learning for Gesture Detection and Localization. In *Proceedings of the European Conference on Computer Vision Workshops (ECCVW)*, 2014.
- [125] Chigozie Nwankpa, Winifred Ijomah, Anthony Gachagan, and Stephen Marshall. Activation Functions: Comparison of Trends in Practice and Research for Deep Learning. *arXiv:1811.03378*, 2018.
- [126] Christopher Olah. Understanding LSTM Networks, 2015.
- [127] Eng-Jon Ong, Helen Cooper, Nicolas Pugeault, and Richard Bowden. Sign Language Recognition using Sequential Pattern Trees. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.
- [128] Eng-Jon Ong, Oscar Koller, Nicolas Pugeault, and Richard Bowden. Sign Spotting using Hierarchical Sequential Patterns with Temporal Intervals. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014.
- [129] Alptekin Orbay and Lale Akarun. Neural Sign Language Translation by Learning Tokenization. *arXiv:2002.00479*, 2020.
- [130] Oğulcan Özdemir, Necati Cihan Camgöz, and Lale Akarun. Isolated Sign Language Recognition using Improved Dense Trajectories. In *Proceedings of the Signal Processing and Communication Application Conference (SIU)*, 2016.
- [131] Oğulcan Ozdemir, Ahmet Alp Kindiroglu, Necati Cihan Camgoz, and Lale Akarun. BosphorusSign22k Sign Language Recognition Dataset. In *9th Workshop on the Representation and Processing of Sign Languages*, 2020.
- [132] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. BLEU: A Method for Automatic Evaluation of Machine Translation. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, 2002.

- 
- [133] Becky Sue Parton. Sign Language Recognition and Translation: A Multidisciplinary Approach From the Field of Artificial Intelligence. *The Journal of Deaf Studies and Deaf Education*, 11(1), 2005.
- [134] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic Differentiation in PyTorch. In *Proceedings of the Advances in Neural Information Processing Systems Workshops (NIPSW)*, 2017.
- [135] Georgios Pavlakos, Vasileios Choutas, Nima Ghorbani, Timo Bolkart, Ahmed A. A. Osman, Dimitrios Tzionas, and Michael J. Black. Expressive body capture: 3d hands, face, and body from a single image. In *Proceedings IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2019.
- [136] Roland Pfau and Josep Quer. Nonmanuals: Their Grammatical and Prosodic Roles. In *Sign Languages*. Cambridge University Press, 2010.
- [137] Tomas Pfister, James Charles, Mark Everingham, and Andrew Zisserman. Automatic and Efficient Long Term Arm and Hand Tracking for Continuous Sign Language TV Broadcasts. In *Proceedings of the British Machine Vision Conference (BMVC)*, 2012.
- [138] Tomas Pfister, James Charles, and Andrew Zisserman. Large-scale Learning of Sign Language by Watching TV (Using Co-occurrences). In *Proceedings of the British Machine Vision Conference (BMVC)*, 2013.
- [139] Lionel Pigou, Sander Dieleman, Pieter-Jan Kindermans, and Benjamin Schrauwen. Sign Language Recognition using Convolutional Neural Networks. In *European Conference on Computer Vision Workshops (ECCVW)*, 2014.
- [140] Lionel Pigou, Aäron Van Den Oord, Sander Dieleman, Mieke Van Herreweghe, and Joni Dambre. Beyond Temporal Pooling: Recurrence and Temporal Convolutions for Gesture Recognition in Video. *International Journal of Computer Vision (IJCV)*, 126(2), 2018.

- 
- [141] Ronald Poppe. A Survey on Vision-based Human Action Recognition. *Image and Vision Computing*, 28(6), 2010.
- [142] Junfu Pu, Wengang Zhou, and Houqiang Li. Iterative Alignment Network for Continuous Sign Language Recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.
- [143] Siddharth S Rautaray and Anupam Agrawal. Vision-based Hand Gesture Recognition for Human Computer Interaction: a Survey. *Artificial Intelligence Review*, 43(1), 2015.
- [144] Leo Sampaio Ferraz Ribeiro, Tu Bui, John Collomosse, and Moacir Ponti. Sketchformer: Transformer-based representation for sketched structure. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.
- [145] Tony Robinson. An Application of Recurrent Nets to Phone Probability Estimation. *IEEE Transactions on Neural Networks (TNN)*, 5(3), 1994.
- [146] Javier Romero, Dimitrios Tzionas, and Michael J. Black. Embodied Hands: Modeling and Capturing Hands and Bodies Together. *ACM Transactions on Graphics, (Proc. SIGGRAPH Asia)*, 36(6), 2017.
- [147] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-Net: Convolutional Networks for Biomedical Image Segmentation. In *Proceedings of the International Conference on Medical Image Computing and Computer-Assisted Intervention*, 2015.
- [148] Simon Ruffieux, Denis Lalanne, Elena Mugellini, and Omar Abou Khaled. A Survey of Datasets for Human Gesture Recognition. In *Proceedings of the International Conference on Human-Computer Interaction (HCI)*, 2014.
- [149] Wendy Sandler. Sign Language and Modularity. *Lingua*, 89(4), 1993.
- [150] Pinar Santemiz, Oya Aran, Murat Saraclar, and Lale Akarun. Automatic Sign Segmentation from Continuous Signing via Multiple Sequence Alignment. In



- 
- Proceedings of the IEEE International Conference on Computer Vision Workshops (ICCVW)*, 2009.
- [151] Benjamin Saunders, Necati Cihan Camgoz, and Richard Bowden. Adversarial Training for Multi-Channel Sign Language Production. In *Proceedings of the British Machine Vision Conference (BMVC)*, 2020.
- [152] Benjamin Saunders, Necati Cihan Camgoz, and Richard Bowden. Progressive Transformers for End-to-End Sign Language Production. In *16th European Conference on Computer Vision (ECCV)*, 2020.
- [153] Adam Schembri, Jordan Fenlon, Ramas Rentelis, Sally Reynolds, and Kearsy Cormier. Building the British Sign Language Corpus. *Language Documentation & Conservation (LD&C)*, 7, 2013.
- [154] Christoph Schmidt, Oscar Koller, Hermann Ney, Thomas Hoyoux, and Justus Piater. Using Viseme Recognition to Improve a Sign Language Translation System. In *Proceedings of the International Workshop on Spoken Language Translation*, 2013.
- [155] Frank M Shipman, Satyakiran Duggina, Caio DD Monteiro, and Ricardo Gutierrez-Osuna. Speed-Accuracy Tradeoffs for Detecting Sign Language Content in Video Sharing Sites. In *Proceedings of the International ACM SIGACCESS Conference on Computers and Accessibility (ASSETS)*, 2017.
- [156] Jamie Shotton, Ross Girshick, Andrew Fitzgibbon, Toby Sharp, Mat Cook, Mark Finocchio, Richard Moore, Pushmeet Kohli, Antonio Criminisi, Alex Kipman, et al. Efficient Human Pose Estimation from Single Depth Images. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 35(12), 2012.
- [157] Ashish Shrivastava, Tomas Pfister, Oncel Tuzel, Joshua Susskind, Wenda Wang, and Russell Webb. Learning from Simulated and Unsupervised Images through Adversarial Training. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

- 
- [158] Ravid Shwartz-Ziv and Naftali Tishby. Opening the Black Box of Deep Neural Networks via Information. *arXiv:1703.00810*, 2017.
- [159] Aliaksandr Siarohin, Enver Sangineto, Stéphane Lathuilière, and Nicu Sebe. Deformable GANs for Pose-based Human Image Generation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.
- [160] Tomas Simon, Hanbyul Joo, Iain Matthews, and Yaser Sheikh. Hand Keypoint Detection in Single Images using Multiview Bootstrapping. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.
- [161] Karen Simonyan and Andrew Zisserman. Two-stream Convolutional Networks for Action Recognition in Videos. In *Proceedings of the Advances in Neural Information Processing Systems (NIPS)*, 2014.
- [162] Karen Simonyan and Andrew Zisserman. Very Deep Convolutional Networks for Large-scale Image Recognition. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2015.
- [163] Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah. UCF101: A Dataset of 101 Human Actions Classes from Videos in the Wild. *arXiv:1212.0402*, 2012.
- [164] Thad Starner and Alex Pentland. Real-time American Sign Language Recognition from Video using Hidden Markov Models. *Motion-Based Recognition*, 1997.
- [165] Thad Starner, Joshua Weaver, and Alex Pentland. Real-time American Sign Language Recognition using Desk and Wearable Computer Based Video. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 20(12), 1998.
- [166] Daniel Stein, Christoph Schmidt, and Hermann Ney. Sign Language Machine Translation Overkill. In *International Workshop on Spoken Language Translation*, 2010.

- 
- [167] Daniel Stein, Christoph Schmidt, and Hermann Ney. Analysis, Preparation, and Optimization of Statistical Sign Language Machine Translation. *Machine Translation*, 26(4), 2012.
- [168] William C Stokoe. Sign Language Structure. *Annual Review of Anthropology*, 9(1), 1980.
- [169] Stephanie Stoll, Necati Cihan Camgoz, Simon Hadfield, and Richard Bowden. Sign Language Production using Neural Machine Translation and Generative Adversarial Networks. In *Proceedings of the British Machine Vision Conference (BMVC)*, 2018.
- [170] Stephanie Stoll, Necati Cihan Camgoz, Simon Hadfield, and Richard Bowden. Text2Sign: Towards Sign Language Production using Neural Machine Translation and Generative Adversarial Networks. *International Journal of Computer Vision (IJCV)*, 2020.
- [171] Chen Sun, Austin Myers, Carl Vondrick, Kevin Murphy, and Cordelia Schmid. VideoBERT: A Joint Model for Video and Language Representation Learning. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2019.
- [172] Ilya Sutskever, Oriol Vinyals, and Quoc V Le. Sequence to Sequence Learning with Neural Networks. In *Proceedings of the Advances in Neural Information Processing Systems (NIPS)*, 2014.
- [173] Rachel Sutton-Spence and Bencie Woll. *The Linguistics of British Sign Language: An Introduction*. Cambridge University Press, 1999.
- [174] Muhammed Mirac Suzgun, Hilal Ozdemir, Necati Cihan Camgoz, Ahmet Kindiroglu, Dogac Basaran, Cengiz Togay, and Lale Akarun. Hospisign: An Interactive Sign Language Platform for Hearing Impaired. In *Proceedings of the International Conference on Computer Graphics, Animation and Gaming Technologies (Eurasia Graphics)*, 2015.

- [175] Christian Szegedy, Sergey Ioffe, Vincent Vanhoucke, and Alexander A Alemi. Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning. In *Proceedings of the AAAI Conference on Artificial Intelligence.*, 2017.
- [176] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going Deeper with Convolutions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.
- [177] Shinichi Tamura and Shingo Kawasaki. Recognition of Sign Language Motion Images. *Pattern Recognition*, 21(4), 1988.
- [178] Mingxing Tan and Quoc V Le. EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2019.
- [179] Stavros Theodorakis, Vassilis Pitsikalis, and Petros Maragos. Dynamic–static unsupervised sequentiality, statistical subunits and lexicon for sign language recognition. *Image and Vision Computing*, 32(8), 2014.
- [180] Naftali Tishby and Noga Zaslavsky. Deep Learning and the Information Bottleneck Principle. In *Proceedings of the IEEE Information Theory Workshop (ITW)*, 2015.
- [181] Sandrine Tornay, Marzieh Razavi, Necati Cihan Camgoz, Richard Bowden, and Mathew Magimai-Doss. HMM-based Approaches to Model Multichannel Information in Sign Language Inspired from Articulatory Features-based Speech Processing. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2019.
- [182] Du Tran, Lubomir Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. Learning Spatiotemporal Features with 3D Convolutional Networks. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2015.

- 
- [183] Matthew Trumble, Andrew Gilbert, Charles Malleson, Adrian Hilton, and John Collomosse. Total capture: 3d human pose estimation fusing video and inertial sensors. In *Proceedings of 28th British Machine Vision Conference (BMVC)*, 2017.
- [184] Yao-Hung Hubert Tsai, Shaojie Bai, Paul Pu Liang, J. Zico Kolter, Louis-Philippe Morency, and Ruslan Salakhutdinov. Multimodal Transformer for Unaligned Multimodal Language Sequences. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, 2019.
- [185] Clayton Valli and Ceil Lucas. *Linguistics of American Sign Language: An Introduction*. Gallaudet University Press, 2000.
- [186] Jan P van Hemert. Automatic Segmentation of Speech. *IEEE Transactions on Signal Processing*, 39(4), 1991.
- [187] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is All You Need. In *Proceedings of the Advances in Neural Information Processing Systems (NIPS)*, 2017.
- [188] Subhashini Venugopalan, Marcus Rohrbach, Jeffrey Donahue, Raymond Mooney, Trevor Darrell, and Kate Saenko. Sequence to Sequence-Video to Text. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.
- [189] Christian Vogler and Siome Goldenstein. Facial Movement Analysis in ASL. *Universal Access in the Information Society*, 6(4), 2008.
- [190] Matthew Vowels, Necati Cihan Camgoz, and Richard Bowden. Gated Variational AutoEncoders: Incorporating Weak Supervision to Encourage Disentanglement. In *Proceedings of the IEEE International Conference on Automatic Face and Gesture Recognition (FG)*, 2020.
- [191] Matthew Vowels, Necati Cihan Camgoz, and Richard Bowden. NestedVAE:

- 
- Isolating Common Factors via Weak Supervision. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.
- [192] Jun Wan, Stan Z Li, Yibing Zhao, Shuai Zhou, Isabelle Guyon, and Sergio Escalera. ChaLearn Looking at People RGB-D Isolated and Continuous Datasets for Gesture Recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2016.
- [193] Hanjie Wang, Xiujuan Chai, Xiaopeng Hong, Guoying Zhao, and Xilin Chen. Isolated Sign Language Recognition with Grassmann Covariance Matrices. *ACM Transactions on Accessible Computing*, 8(4), 2016.
- [194] Shuo Wang, Dan Guo, Wen-gang Zhou, Zheng-Jun Zha, and Meng Wang. Connectionist Temporal Fusion for Sign Language Translation. In *Proceedings of the ACM International Conference on Multimedia*, 2018.
- [195] Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7794–7803, 2018.
- [196] Shih-En Wei, Varun Ramakrishna, Takeo Kanade, and Yaser Sheikh. Convolutional Pose Machines. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.
- [197] Ronnie B. Wilbur. Phonological and Prosodic Layering of Nonmanuals in American Sign Language. *The Signs of Language Revisited: An Anthology to Honor Ursula Bellugi and Edward Klima*, 2000.
- [198] Terry Windeatt, Cemre Zor, and Necati Cihan Camgoz. Approximation of Ensemble Boundary using Spectral Coefficients. *IEEE Transactions on Neural Networks and Learning Systems (TNNLS)*, 30(4), 2019.
- [199] Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V. Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, Jeff Klingner, Apurva Shah, Melvin Johnson, Xiaobing Liu, Łukasz Kaiser, Stephan Gouws, Yoshikiyo Kato, Taku Kudo, Hideto Kazawa, Keith Stevens,

- 
- George Kurian, Nishant Patil, Wei Wang, Cliff Young, Jason Smith, Jason Riesa, Alex Rudnick, Oriol Vinyals, Greg Corrado, Macduff Hughes, and Jeffrey Dean. Google's Neural Machine Translation System: Bridging the Gap Between Human and Machine Translation. *arXiv:1609.08144*, 2016.
- [200] Donglai Xiang, Hanbyul Joo, and Yaser Sheikh. Monocular Total Capture: Posing Face, Body, and Hands in the Wild. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.
- [201] Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhudinov, Rich Zemel, and Yoshua Bengio. Show, Attend and Tell: Neural Image Caption Generation with Visual Attention. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2015.
- [202] Ruiduo Yang and Sudeep Sarkar. Detecting Coarticulation in Sign Language using Conditional Random Fields. In *International Conference on Pattern Recognition (ICPR)*, 2006.
- [203] Zhilin Yang, Zihang Dai, Ruslan Salakhutdinov, and William W Cohen. Breaking the Softmax Bottleneck: A High-Rank RNN Language Model. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2018.
- [204] Fang Yin, Xiujuan Chai, and Xilin Chen. Iterative Reference Driven Metric Learning for Signer Independent Isolated Sign Language Recognition. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2016.
- [205] Norimasa Yoshida. *Automatic Utterance Segmentation in Spontaneous Speech*. PhD thesis, Massachusetts Institute of Technology, 2002.
- [206] Zahoor Zafrulla, Helene Brashear, Thad Starner, Harley Hamilton, and Peter Presti. American Sign Language Recognition with the Kinect. In *Proceedings of the ACM International Conference on Multimodal Interfaces (ICMI)*, 2011.
- [207] Jan Zelinka, Jakub Kanis, and Petr Salajka. NN-Based Czech Sign Language Synthesis. In *International Conference on Speech and Computer*, 2019.

- [208] Zhengyan Zhang, Xu Han, Zhiyuan Liu, Xin Jiang, Maosong Sun, and Qun Liu. ERNIE: Enhanced Language Representation with Informative Entities. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, 2019.
- [209] Zhengyou Zhang. Microsoft Kinect Sensor and Its Effect. *IEEE Multimedia*, 2012.
- [210] Jiaojiao Zhao and Cees G. M. Snoek. Dance With Flow: Two-In-One Stream Action Detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.
- [211] Hao Zhou, Wengang Zhou, and Houqiang Li. Dynamic Pseudo Label Decoding for Continuous Sign Language Recognition. In *Proceedings of the IEEE International Conference on Multimedia and Expo (ICME)*, 2019.
- [212] Hao Zhou, Wengang Zhou, Yun Zhou, and Houqiang Li. Spatial-Temporal Multi-Cue Network for Continuous Sign Language Recognition. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2020.