Data Mining and Modelling for Sign Language

Boris Mocialov

Submitted for the degree of

Doctor of Philosophy

EPSRC Centre for Doctoral Training

in Robotics and Autonomous Systems

Heriot-Watt University and The University of Edinburgh

January, 2021

Abstract

Sign languages have received significantly less attention than spoken languages in the research areas of corpus analysis, machine translation, recognition, synthesis and social signal processing, amongst others. This is mainly due to signers being in a clear minority and there being a strong prior belief that sign languages are simply arbitrary gestures. To date, this manifests in the insufficiency of sign language resources available for computational modelling and analysis, with no agreed standards and relatively stagnated advancements compared to spoken language interaction research. Fortunately, the machine learning community has developed methods, such as transfer learning, for dealing with sparse resources, while data mining techniques, such as clustering can provide insights into the data. The work described here utilises such transfer learning techniques to apply neural language model to signed utterances and to compare sign language phonemes, which allows for clustering of similar signs, leading to automated annotation of sign language resources. This thesis promotes the idea that sign language research in computing should rely less on hand-annotated data thus opening up the prospect of using readily available online data (e.g. signed song videos) through the computational modelling and automated annotation techniques presented in this thesis.

To Zanna and Tonny for always being patient and understanding.

Acknowledgements

I thank Heriot-Watt University for providing access to the facilities that were needed for this research and the CDT in Robotics and Autonomous Systems for the organised events that allowed me to expand my knowledge beyond the technical expertise. The limitless guidance from both Professor Helen Hastie and Professor Graham Turner has paved the way towards what this thesis represents. Periodic help from Dr Patricia Vargas and Dr Katrin Lohan had major contributions to this research and its directions. Major credit deserve the gatekeepers of various sign language resources, who directly or indirectly have given hints and provided ideas for the use of those resources. Last but not least expressing my appreciation to the fellow students who gave constructive feedback.

ACADEMIC REGISTRY

Research Thesis Submission

Name:	Boris Mocialov		
School:	Engineering and Physical Sciences		
Version: *(i.e. First, Resubmission, Final)*	First	Degree Sought:	PhD

Declaration

In accordance with the appropriate regulations I hereby submit my thesis and I declare that:

1) the thesis embodies the results of my own work and has been composed by myself
2) where appropriate, I have made acknowledgement of the work of others and have made reference to work carried out in collaboration with other persons
3) the thesis is the correct version of the thesis for submission and is the same version as any electronic versions submitted*.
4) my thesis for the award referred to, deposited in the Heriot-Watt University Library, should be made available for loan or photocopying and be available via the Institutional Repository, subject to such conditions as the Librarian may require
5) I understand that as a student of the University I am required to abide by the Regulations of the University and to conform to its discipline.
6) I confirm that the thesis has been verified against plagiarism via an approved plagiarism detection application e.g. Turnitin.

* *Please note that it is the responsibility of the candidate to ensure that the correct version of the thesis is submitted.*

Signature of Candidate:		Date:	31/08/2020

Submission

Submitted By *(name in capitals)*:	BORIS MOCIALOV
Signature of Individual Submitting:	
Date Submitted:	31/08/2020

For Completion in the Student Service Centre (SSC)

Received in the SSC by *(name in capitals)*:			
Method of Submission *(Handed in to SSC; posted through internal/external mail)*:			
*E-thesis Submitted (**mandatory for final theses)***			
Signature:		Date:	

Please note this form should be bound into the submitted thesis.

Contents

List of Figures

List of Tables

Glossary

Gloss	Context-dependent translation of a sign (e.g. 'HOME' in BSL).
ID-Gloss	Context-independent label (e.g. 'HOUSE' for the gloss 'HOME').
Lemma	The basic form of a word (equivalent of a headword in a dictionary)
Movement epenthesis	Transitional hand movement between the two consecutive signs.
Phoneme	Perceptually distinct combination of phonological parameters.
Phonological parameter	Minimal sub-lexical component in sign language, which can either be a handshape, hand orientation, hand location, hand movement, or non-manual gestures.
Sign	One or more phonemes representing a meaningful unit.

Abbreviations

ASL	The American Sign Language
BSL	The British Sign Language
DTS	The Danish Sign Language
Libras	The Brazilian Sign Language
NGT	Sign Language of the Netherlands

Libraries Used

PyTorch Scikit-learn Tensorflow	Machine learning modelling frameworks
OpenPose	Pose estimation library

Chapter 1

Introduction

Mostly, the first thought that comes to mind when thinking about natural languages is about spoken languages due to the majority of people using spoken languages and, hence, these languages are the ones that have received the most attention in computing over the years. However, a significant proportion of the population use *sign languages* for communication within the deaf community. Signers compose meaningful sequences of signs that enable communication by combining *phonological parameters* such as: hand shape, hand and finger orientation, hand location relative to the body, hand movement, and non-manual features such as facial expressions. Traditional research on sign language in linguistics [27, 26, 197] breaks down signs into their constituent parts and uses data mining techniques to study patterns in sign languages, often introducing a large overhead on annotation of signing videos. In computing, similar to natural language processing (NLP) applied to speech and text, sign languages have been modelled for the last few decades with the aim of understanding, synthesising, and translating unconstrained signed utterances. With the rise of machine learning techniques, annotated sign language resources are in even greater demand to support such models and reap the benefits of the latest developments in deep machine learning methods. However, due to the smaller number of sign language users, such resources are scarce. Nevertheless, sign language research is advancing rapidly with linguists being interested in the use and the evolution of language within and outwith communities, while the research on sign language in computing is mostly motivated by the potential social impact and often targets the communication gap between deaf and non-signing [78, 130, 90]. Contrary to such noble motivations coming from the Computer Science community, there has been growing resistance from the Deaf community [242].

While linguists working with sign language often take the anthropologist approach to studying languages [134, 182], researchers in computing often see sign languages as an interesting application for the general-purpose models [14, 149]. As a consequence, solutions proposed by the computing community often lack the knowledge about the needs of the Deaf community [31]. Therefore, certain products do not find their way into everyday use [242], despite large projects in sign language recognition, translation, and sign generation [173, 172, 271]. The reason for this is the magnitude of the dimensions associated with the problem of sign language understanding. During signing, the information is delivered through the context, signer's hands, body pose, facial expressions, and the space around the body. Individually, these dimensions are defined by corresponding handshapes, orientations, signing location in relation to the body parts, and gestures. Together, this diverse mix points to a meaning of what the signer is communicating. Modelling every individual dimension solves the problem partially as the correlation of the dimensions and their frequency during signing plays an important role. Therefore, this thesis focuses on sign language resources and techniques to better understand the complexity of the language and how this complexity could be incorporated into modelling. This thesis explores how both fields (data mining in linguistics and modelling in Computer Science) could benefit from mutual research and how the potential of sign language data, readily available on the internet, could be exploited in order to overcome sparsity of sign language resources.

1.1 Motivation

This research sets out to show that data-driven research in computing for sign language can be approached from a novel angle. Traditional research begins with a collection of limited vocabulary signs in highly-controlled settings for the specific purposes of that research. This results in sparsity and insufficiency of sign language resources for research in both Linguistics and Computer Science. The approaches proposed here explore noisy resources, apply data mining to find significant co-dependencies, and reuse pre-trained models in order to overcome the sparsity of the available sign language data. Such novel data mining approaches open up opportunities, such as finding similar phonemes in different signing videos and comparing sign languages, relying on the data rather than the linguistic perspective, which is expensive and resource intensive. Furthermore, data mining gives an opportunity to analyse continuous signing videos and cluster similar phonemes, signs, and even phrases, saving annotation time and even annotating automatically. Therefore, the overall motivation for this thesis is to be able to harvest as much value from the little publicly available heterogeneous sign language data, with a view to support computational modelling of sign language.

This work can have impact on both Linguistics and Computer Science research communities. Linguists, on one hand, can benefit from automated pattern recognition and automated suggestions during the annotation of raw sign language videos. Computer scientists working on modelling sign language data, on the other hand, can benefit from this research by utilising the automated generation and transcription of sign language data that can be used for sign language translation purposes, which is one of the most popular topics in sign language research in computing today [94, 174, 52, 233, 218]. Therefore, data mining can provide the tools needed for automated analysis of complex data, which in turn can provide insights into supervised as well as unsupervised modelling.

1.2 Sign Language Preliminaries

A number of sign languages and multiple corpora will be explored in this thesis, which can be grouped into French and British Sign Language families [30], however, such groupings are largely speculative since it is challenging to trace the origins and the influences a sign language might have had throughout its history [270].

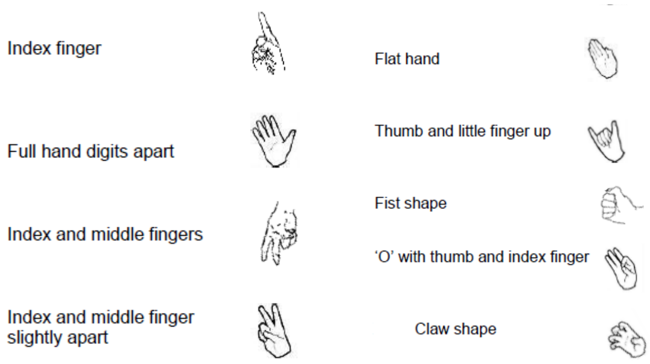

Index finger
Full hand digits apart
Index and middle fingers
Index and middle finger slightly apart
Flat hand
Thumb and little finger up
Fist shape
'O' with thumb and index finger
Claw shape

Figure 1.1: BSL handshape phonological parameter defines 9 handshapes that serve as basis for creation of signs in BSL (Source: deafsupport.org.uk).

By way of an introduction to sign languages, this section takes a closer look at BSL and assumes that similar general rules apply to most sign languages in that all sign languages use a set of phonological parameters with varying handshapes [11], as can be seen in Figure 1.1 for BSL. Other than the handshape, Stokoe et al. identified phonological parameters such as hand movement, hand location relative to the body, hand and extended finger orientation, and non-manual gestures such as facial expressions, all used simultaneously during signing [235]. Combined phonological parameters (excluding motion) correspond to a single frame in a video footage and can be used to sign letters. Adding motion phonological parameter creates a phoneme that may represent a whole sign or a part of a sign. Originally, the term ‘chereme’ was used to refer to the combined location, handshape, and the movement phonological parameters as means of distinguishing signed languages from spoken languages, however, it has later been shown that the phonological terminology of the spoken and signed language is organisationally and functionally equivalent at the sub-lexical level [163]. Nevertheless, these original terms are being used to date in both computing and linguistic studies [41, 209]. Trying to find a parallel between signed and spoken languages, one can equate sign phonological parameters with spoken phonetics, phonemes in sign language with spoken phonemes, and signs with words. The syntax of any natural language specifies rules that guide the meaningful construction of utterances [240]. If the syntactic rules are not obeyed, the meaning may change or even get lost. To create a new sign with a specific meaning, a signer uses rules to combine the phonological parameters together, while individual signs are combined by adding a transition between the signs, which is called *epenthesis* and does not carry any linguistic meaning.

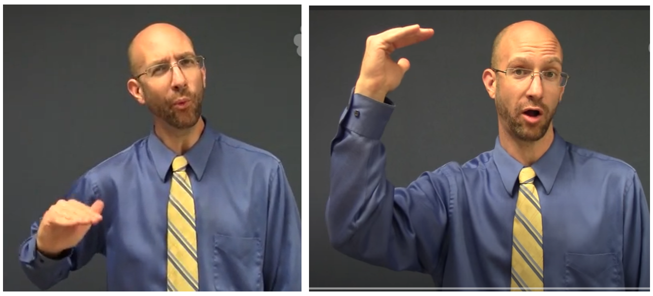
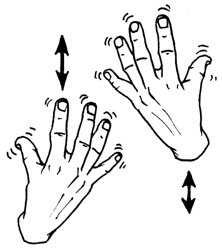
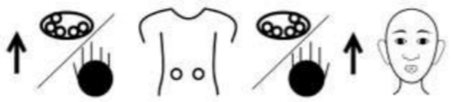
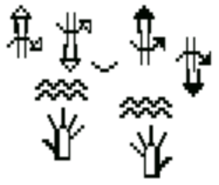
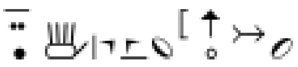


Figure 1.2: BSL for ‘SHORT’ on the left and ‘TALL’ on the right that can be applicable to describe the height of a person or a building combined together with a sign for a person or a building. Non-manual gestures, such as facial expressions are also important in understanding a sign Source: YouTube.

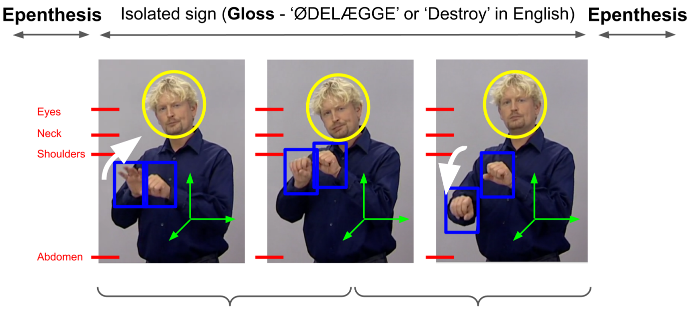
The meaning of a sign can be further modified using changes in phonological parameters that would resemble the real-life observation as, for example, showing relative height of buildings as in Figure 1.2. Similar to vocabularies in spoken languages, BSL has a *fixed* lexicon (similar to the words in a dictionary in spoken languages) and a *productive* lexicon of signs that can be constructed ‘on the fly’ that is potentially infinite and can sometimes make its way into the fixed lexicon [34]. Fixed and productive lexicons of BSL make use of the rules of the language to construct new signs and new meanings as required for communicative purposes. Moreover, co-articulation coupled with the movement epenthesis affects the production of signs as hands move from one sign to another with the next sign being influenced by the hand configuration, location, or orientation of the previous sign whilst still remaining conceptually distinguished [223]. Consequently, the existence of co-articulation, productive lexicon, and non-exhaustive combinations of phonological parameters shows how fluid sign languages are and how challenging they are to model.

Table 1.1: Sign ‘FIRE’ that is similar in German and American sign languages is illustrated in the first row (source: www.lifeprint.com) written using SiLOrB (source: www.bleegiimuusclark.com), SignWriting (source: www.signwriting.org), and HamNoSys (source: www.sign-lang.uni-hamburg.de) notation systems. The set of symbols is different in each notation, some of which may appear iconic (e.g. hands in SiLOrB, SignWriting, and HamNoSys, face and torso in SiLOrB).

<table>
<tr><td colspan="2"></td></tr>
<tr><td>SiLOrB</td><td></td></tr>
<tr><td>SignWriting</td><td></td></tr>
<tr><td>HamNoSys</td><td></td></tr>
</table>

Signed languages have a number of writing conventions that have been proposed over the time, some of which include Hamburg Sign Language Notation System (*HamNoSys*) [112], SignWriting [239], and SiLOrB [53]. Examples of the sign writing systems can be seen in Table 1.1 where hands, body parts, and motion can be observed. These are the writing systems for the phonological parameters that can be applied to any sign language. Beside the notation systems for phonological parameters, sign language can be written using *glosses* or *ID-Glosses*. While a gloss is the translation of the meaning of a sign in contrast ID-gloss is a word that is consistently used throughout a corpus to label a sign, regardless of the context in which that sign is used. Therefore, for glosses, different labels are possible for the same sign while this is not the case for the ID-Glosses. The benefit of using ID-Gloss over gloss is that the ID-Gloss provides information about the sign phonological parameters (sign form). Usually, ID-Gloss requires a reference lexical database, from which labels are pulled. During the annotation, continuous signing is segmented into isolated signs and a label for each sign is picked from this database. The use of the ID-Glosses ensures not only consistent use of labels across the corpus, but also consistency when the corpus has multiple annotators or the same annotator annotating the corpus at different intervals. In essence, both gloss (see Figure 1.3) and ID-gloss (see Figure 2.2) are translations of individual isolated signs into their written language equivalent. For example, a gloss ‘HOME’ could be ID-Glossed as ‘HOUSE’ since both signs are signed the same in BSL and ‘HOUSE’ is the *lemma*. *Free translation* is the translation of an entire utterance into spoken language wording (note word order may vary) [125, 124] (see Figure 2.3). Glosses, ID-Glosses, and free translations are often used in corpus annotation, whereas phonological parameter and epentheses annotations are rare.

It is challenging to speak grammatical English and sign grammatical BSL at the same time [12], because the syntax of the two languages is mostly different. In particular, the common word order of English language is: subject, verb, object. BSL, on the other hand, uses: time frame, topic, comment sign order [240]. For example, a signed BSL utterance (glossed) ‘YESTERDAY (time frame) WEATHER (topic) AWFUL (comment)’ could be translated (free translation) as ‘The weather (topic) was awful (comment) yesterday (time frame)’ [68].



Phonological Parameters

		Phoneme 1		Phoneme 2	
	Colour	Start	End	Start	End
Handshape Right	Blue	5-hand			s-hand
Handshape Left		s-hand			s-hand
Orientation fingers right	Green	front up			front
Orientation fingers left		front up			
Orientation palm right		down			
Orientation palm left		down			
Location	Red	hand edge next to the thumb			
Movement	White			down front	
Non-Manual Features	Yellow				

Figure 1.3: All sign language concepts discussed in this thesis (epenthesis, gloss, phoneme, and phonological parameters in bold) shown on an example of an isolated sign ('DESTROY') with accompanying phonological annotation from the Ordbog over Dansk Tegnsprog dataset (see Table 2.1). Phonemes are inferred later using method that will be discussed in Chapter 5. Phonological annotation in Ordbog over Dansk Tegnsprog dataset specifies phonological parameters throughout the video without explicitly stating when a change in phonological parameter takes place in case if a sign contains more than one phoneme. Note, ID-Gloss and gloss in this case are the same as both refer to the same lemma.

Figure 1.3 demonstrates all the concepts that will be used throughout the thesis (epenthesis, gloss, ID-Gloss (in caption), phoneme, and phonological parameters) on one example of an isolated sign from the Ordbog over Dansk Tegnsprog dataset (see Table 2.1). A sign 'DESTROY' has two phonemes, separated by the change in the hand speed, which also corresponds to a change in the handshape phonological parameter. As it can be seen, not all the phonological parameters are annotated in the Ordbog over Dansk Tegnsprog dataset with non-manual features not being annotated at all since these are usually more subtle and require a great amount of effort from a linguist, familiar with the language.

1.3 Research Questions

Sign language resources such as signing videos and their various methods of annotation as described above are central to this thesis. The main research question of this thesis is thus:

> “ Can data mining used in data exploration assist modelling of sign language? ”

This research question is broken down into finer-grained questions in Table 1.2, which will be addressed in turn in this thesis.

Table 1.2: Research Questions and corresponding key concepts together with the chapter numbers where each question is answered.

#	Research Question	Key Concepts	Chapter
RQ1	Can a small number of BSL sentences be modelled at the gloss level using written English language models, knowing that the syntax of the two languages is different?	- Language Modelling - Gloss - Transfer Learning	3
RQ2	Can movement speed tracking be used in identifying sign boundaries in continuous signing?	- Temporal Segmentation - Phonological Parameters - Linguistic Heuristics	4
RQ3	Can similar signs or phrases be found in different continuous signing videos without having the transcription?	- Clustering - Phonological Parameters - Phoneme	5
RQ4	Can automatic classification of phonological parameters (e.g. location and orientation) be improved by exploiting their co-dependencies?	- Hand Classification - Hand Detection - Phonological Parameters - Co-Dependency	6
RQ5	Can two unrelated sign languages be compared using location and orientation phonological parameters?	- Co-Dependency - Phonological Parameters	7

1.4 Thesis Structure

This research develops novel methods for modelling and mining for sign language data. Before getting into experimentation, the thesis outlines existing research on sign language in computing and in linguistics and describes the existing sign language resources alongside the potential use of online user-generated resources (Chapter 2).

In order to answer the main research question, first sign language glosses will be modelled using techniques described in Chapter 3. This will be achieved by pre-training English language models and fine-tunning them on transcribed British Sign Language (BSL) using transfer learning techniques, answering the first research question (RQ1). We will show that signed languages have grammatical aspects that are not present in written language, which limits how much written languages can be used in modelling sign languages.

Identifying sign boundaries is important for isolating individual signs for the downstream sign classification task. Chapter 4 will look at spatial segmentation of phonological parameters such as location and orientation, along with the temporal segmentation of individual signs from continuous signing videos. This chapter will show that it is sufficient to use sign linguistic heuristics in identifying sign boundaries, answering the second research question (RQ2). These segmentation techniques will be used in the following two chapters.

Using temporal and spatial segmentation methods, Chapter 5 will show that phonemes can be clustered by comparing phonological parameters, answering the third research question (RQ3). This clustering method is then extended to show that similar signed phrases can be found in videos by automatically analysing interpreted songs on social media.

Next, in Chapter 6, an end-to-end model is used to classify multiple phonological parameters and model co-dependence of the two phonological parameters, answering the fourth research question (RQ4), showing that encoding co-dependence of classes into a model helps to reduce the search space and reduce the training time.

Having acknowledged the importance of the phonological parameters for modelling, they will then be used to compare different sign languages in the case study chapter (Chapter 7). This method will show that by discovering significant occurrences of location and orientation phonological parameters, it is possible to compare the signing trends among the two languages, answering the fifth research question (RQ5). Finally, limitations of the thesis will be presented in the conclusion chapter alongside a discussion of future directions. In addition, appendices provide additional information to the reader about how discussed techniques in this thesis can be applied to specific problems. In Appendix A, we discuss spatial segmentation at various levels (torso, limbs, fingers) and a frame similarity encoding that compares each frame to another in a video, which can show similar patterns across video segments. In Appendix B, we show how temporal segmentation can be used to segment isolated signs in continuous signing video and discuss whether phonological parameters, such as non-manual features, contribute to the accuracy of the model.

Chapter 2

Prior Work

This chapter will begin by reviewing available sign language resources used in the literature in both computing and linguistics research. Often, such resources are limited or are collected for very specific goals. Such sign language resources require annotation and transcription, where the annotation can be done at different levels of detail and the transcription is mostly represented as translation into written language. Therefore, this chapter will look at the various annotation schemes for sign language. Phonological parameters and epenthesis need to be identified from raw signing footage in order to process the data for later analysis. That is why this chapter will also discuss approaches to spatial and temporal segmentation. Later, various modelling techniques for language modelling, phonological parameters, isolated and continuous signing will be discussed. Finally, the chapter will show how data mining has been used with sign language resources in the context of data analysis for verifying linguistic theories. To summarise, the presented literature will be discussed and a gap in sign language research identified, which will serve as the basis for this thesis.

2.1 Computing and Sign Language

Early research on vision-based sign language recognition in computing tackled finger-spelling as can be seen in Figure 2.1.

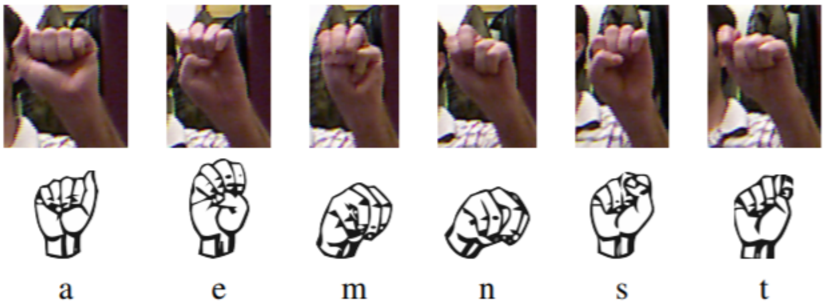
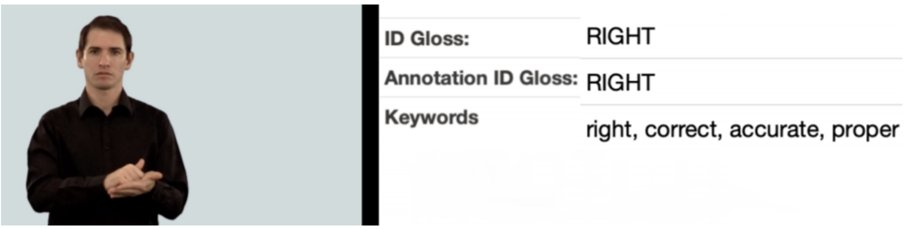


Figure 2.1: Screenshot from ASL fingerspelling dataset that shows six letters from the English alphabet signed in ASL [207].

It was successful because it can be described with simple data such as static images [23, 111] with the main challenge being changes in hand orientation. Recently, research in fingerspelling is tackling the challenge of recognising hand shapes in dynamic environments with various backgrounds, hand sizes, hand orientations, and skin colours [225, 203, 138] with varying success tackling problems of variable hand sizes by adjusting the size of the search window [1] or using features that are invariant to small variations in translation, rotation, scale, and colour [152, 7]. Traditionally, research in sign language in computing has looked at the recognition of individual signs, whether in isolation or as a part of continuous signing, providing one gloss for each sign [52]. However, limited attention has been paid to epentheses modelling as modern techniques are assumed to generalise over the variations present in signs when epenthesis is considered a part of the sign [263]. With the rise of encoder-decoder models [131, 10], free translation of continuous signing has gained attention [52, 139, 191, 216].

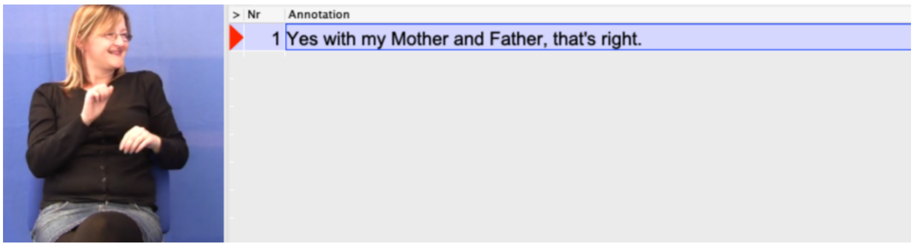
To clarify the difference between recognition and free translation in sign language, recognition is a transition from raw video to a gloss or ID-Gloss or a set of glosses as shown in Figure 2.2.



ID Gloss:	RIGHT
Annotation ID Gloss:	RIGHT
Keywords	right, correct, accurate, proper

Figure 2.2: Screenshot from BSL SignBank [91] of lexical database and English ID-Gloss for BSL sign (‘RIGHT’).

On the other hand, free translation in sign language is a transition from raw video with a sequence of signs to a sequence of words in English (or any other spoken language) where the order of words may vary [179] as shown in Figure 2.3.



1 Yes with my Mother and Father, that's right.

Figure 2.3: Screenshot from BSL Corpus [219] of Free Translation from BSL to English.

Some research focuses on reverse translation from unconstrained text to sign language glosses [200] or avatar movement [167]. Generating avatar movement by combining phonological parameters poses novel problems to the community. The challenge with this approach is that the synthesised signs do not look like natural signing due to the lack of variation in motion trajectories and speed. There are solutions to this problem including explicitly inserting epentheses into avatar animation [222] or training on motion capture data [127]. Despite the diversity of efforts directed towards sign language in computing, this thesis does not explore fingerspelling, translation, and synthesis of sign language any further and instead it covers *sign language modelling* both for isolated and continuous signs as well as epentheses modelling. Similarly, data mining is a broad field with many methods and techniques [259]. Here, however, the focus is on *clustering* and data analysis for discovery of patterns in data. At the same time, this research has been influenced significantly by the methods taken in aforementioned research directions. In particular, the learned convolutional filters previously trained on myriad of images have been utilised in this research in extracting invariant features from raw images while the end-to-end models were utilised for reducing the complexity of the signing.

2.2 Sign Language Resources

High-quality corpora is useful to both linguists and computer scientists and is used for different purposes. Linguists justify or form hypotheses about sign languages by looking at signs that signers are producing [26, 197, 27], while computer scientists model signs [195], pauses between the signs [263], or synthesise signing by learning kinematics of signing from the data [127]. Unfortunately, access to collected corpora is strictly regulated for most of the resources mainly for privacy (e.g. difficulty anonymising participants [121]), technical (e.g. space requirements, data representation methods), historical (e.g. lack of standards), or financial (e.g. sponsor terms and conditions) reasons [122]. Nevertheless, for the purposes of fair scientific research, larger and better resources frequently become available despite the fact that a standard evaluation datasets has not yet been established in Computer Science research community [83]. Furthermore, a list of prominent corpora is presented that appears often in the literature in both linguistic and Computer Science research. This list identifies the country and language of every resource along with whether the signers are native (L1) or not (L2), the number of signs in the data, number of signers recorded, whether the data contains continuous signing or isolated signs in every video, type of the data, and annotation style.

Table 2.1: A list of resources that often appear in research on sign language. The list shows the name of the resource, its country as the language of the resource, the language proficiency of its participants, vocabulary size, and the number of participants. Isolated (I) or continuous (C) column tells whether the footage is segmented into isolated signs or whether it is continuous. Data type column indicates the format of the resource. Annotation column specifies the type of annotation the resource provides.

Name	**Country**	**L1/ L2**	**Lexicon**	**Signers**	**Isolated (I)/ Continuous (C)**	**Data Type**	**Annotation**
SSLC-L2 [177]	Sweden	L2	-	38	C	Video	Gloss, POS, Free Translation, Phonological Parameters
Ordbog over Dansk Tegnsprog [245]	Denmark	L1	1600	-	I	Video	Gloss, Phonological Parameters
POLYTROPON [84]	Greece	-	2000	-	C	RGB-D	Gloss, Translation, Phonological Parameters
ASL Signbank [117]	USA	L1	1000	-	I	Video	ID Glosses, Phonological Parameters, Relationships
- [119]	Korea	mix	2400	60	I	Video	ID-GLoss
LESCO Corpus [201]	Costa Rica	-	960	27	C	Video	ID-Gloss, Translation, Phonological Parameters
Cologne Corpus [3]	Germany	L2	281	350	C	Video	ID-Glosses, Translation, Phonological Parameters
SSLC [178]	Sweden	L1	-	42	C	Video	ID-Gloss
NGT [63]	Netherlands	L1	3900	100	C	Video	ID-Gloss, Translation
BSLCP [219]	UK	L1	1800	249	I & C	Video	ID-Gloss, Translation
RWTH-PHOENIX-Weather [94]	Germany	L1	911	7	C	Video	Gloss
RWTH-PHOENIX-Weather 2014 [95]	Germany	L1	1558	9	C	Video	Gloss
MS-ASL [126]	USA	-	1000	222	I	Video	Gloss
DEVISIGN [44]	China	-	4414	30	I	RGB-D	Gloss
SMILE [81]	Switzerland	mix	100	30	I	RGB-D	Phonological Parameters, Gloss
CSL Cyberglove [98]	China	-	5113	6	I	Cybergloves Data	Gloss, Phonological Parameters
SIGNUM [254]	Germany	L1	450	20	I & C	Video	Gloss
ATIS [38]	-	-	400	-	C	Video	Gloss
RWTH-BOSTON-104 [76]	USA	-	104	3	C	Video	Gloss
ASLLVD [190]	USA	L1	3300	6	I	Video	Gloss, Phonological Parameters, Epentheses
A3LIS-147 [86]	Italy	-	147	10	I	Video	Gloss
WLASL [160]	USA	-	2000	119	I	Video	Gloss, start/end frames per sign, signer bounding box

Table 2.1 shows a list of resources that often appear in research on sign language. The list shows the name of the resource if it has one and a reference to the original work that presented this resource. Country indicates the language of the resource as well as the language proficiency of participants, with L1 being native signers and L2 being non-native. The 'lexicon' column shows the vocabulary size of the resource and the 'signers' column shows the number of participants. The 'Isolated' (I) or 'Continuous' (C) column tells whether the footage is segmented into isolated signs or whether it is continuous, which usually involves some task, such as signing regarding a specific topic. From the table, it can be seen that most of the resources use RGB cameras for sign language data representation because vision is a cheap, reliable, and widely-available sensor that provides rich content [256]. However, since signing is happening in 3D space, depth information is being lost when recording with a single video camera [185, 184]. Such approaches are, therefore, less efficient at capturing more complete signing information. To overcome this limitation, other approaches use either more than one camera [247] or devices such as Microsoft Kinect [262], motion capture systems [123], or Leap motion sensor [51, 88] that capture depth information by emitting and capturing reflected light. Direct measurement devices, such as data gloves are also used for capturing signs [151]. Although these methods improve recognition precision [195], they obstruct the signing process as signing involves contact between the hands and the body [199] as well as the excess of wires in the older versions of data gloves [111] or use of markers needed for motion capture devices [123]. Recent computer vision approaches attempt to reconstruct 3D information from RGB cameras [243], however these techniques are yet to make their appearance in sign language research. This thesis will make use of some of the resources from the Table 2.1, such as:

- ID-Gloss annotation from the BSLCP corpora for modelling sign language glosses using English language model since this resource is easily available and the availability of an expert on BSL sign language.
- ID-Gloss annotation and raw video footage from the NGT for segmenting isolated signs from continuous signing videos since this resource is easily available and provides a large amount of continuous footage.
- Phonological parameter annotation and isolated raw sign videos from the Ordbog over Dansk Tegnsprog for modelling phonological parameters since this resource

provides substantial quantity of high-quality isolated sign videos with corresponding well-documented annotations.

Sign language corpora of various data types is usually acquired and annotated in a laborious and diligent manner with a team of experts to ensure high quality and broad utility [219, 123]. While it is imperative to attentively follow the developments of the existing and the new corpora that is collected in such a fashion to promote standards, it is worth noting that lower-quality (but still useful) signing data is available online. For example, songs are often being interpreted in different sign languages with the songs having publicly accessible lyrics in written languages. While deaf-specific music performers are rare compared to vocal singers, sign language users resort to 'listening' to interpretations of the songs found in the spoken or written languages that are being interpreted by those who can both hear and sign. This is evidenced by the relatively large amount of the content found in online resources such as TikTok, YouTube, or Instagram. Such content makes the interpretation of the spoken songs possible for the deaf community, also encouraging visualisation of music [72]. In Chapter 7, we present one such use of the YouTube online resource for comparison of two sign languages. A pipeline is devised that downloads appropriate videos from the online resource, checks it using multiple quality rules and analyses the data, keeping only the higher-level video description with no user information [183].

2.2.1 Sign Language Annotation

Different annotation techniques are listed in [146], but usually sign languages are annotated using glosses, ID-Glosses, free translation, and occasionally phonological parameters [53]. The annotation is usually dictated by the research aims [122, 136] and there is no agreed standard for how the sign language corpus should be annotated to serve the needs of everyone, which in itself can be a problem when comparing collected sign language resources [13]. Annotation of non-manual activity, such as eye gaze or mouthing gets low attention despite the fact that it may contribute the most to understanding of signing [60]. This is reflected in the annotation schemas in Table 2.1 with some resources employing more annotations schemes than others. This work will mostly make use of ID-Gloss and annotation of phonological parameters using HamNoSys notation, which is analogous to the International Phonetic Alphabet (IPA) [211] used for writing spoken languages. This thesis advocates for the division of efforts into feature encoding either using with the help of the linguistic annotation HamNoSys or skeletal information and modelling.

Table 2.2: Approximate number of symbols for every phonological parameter in HamNoSys notation system. The number is approximate because the notation system defines combinations of potential configurations or actions for each parameter. Special cases with low frequency are not listed.

Phonological Parameter	**HamNoSys Phonological Parameter Sub-type**	**Approximate Number**
Handshape	Hand shapes	72
Orientation	Extended finger directions	18
	Palm orientations	8
Location	Hand locations	46
	Hand location sides	5
	Hand distances	5
Movement	Hand movements	7
	Other movements	1
	Movement directions	6
	Movement speeds	5
	Movement repetitions	7
Non-manuals	Eye gaze Facial expression Mouth gestures	12

While the glosses, ID-Glosses, and free translation annotations are shown in Figures 1.3, 2.2, and 2.3 respectively, Table 2.2 shows the approximate number for HamNoSys sub-types. Despite the fact that non-manual features, such as facial gestures, play an essential part in interpretation of the sign languages, HamNoSys has a relatively poor notation system for them as these are much more complicated and subtle than other phonological parameters. The notation aims to enumerate hand shape observations from all sign languages and presents combinations of palm curvature, extension of the thumb, and finger configuration to describe a handshape. As for the orientation and location phonological parameters, the orientation is split into extended finger direction and palm orientations, which can point to either of the four cardinal directions. Location, on the other hand, defines specific body parts (e.g. shoulder, head, etc.) and uses those relative body locations to identify the hand location. Movement, on the other hand, categorises hand movement types and movement speeds.

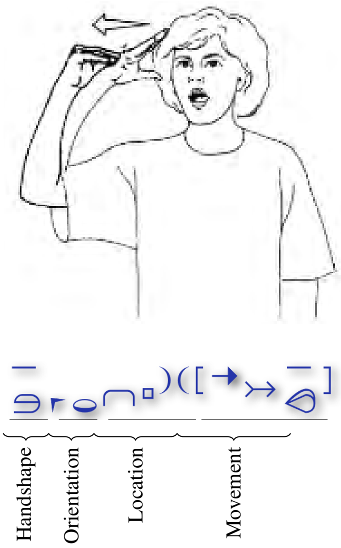


Figure 2.4: Example of a sign ‘HAMBURG’ in German Sign Language and its HamNoSys notation (Source: HamNoSys documentation)

Figure 2.4 shows an example of a sign ‘HAMBURG’ in German Sign Language (GSL), described using the HamNoSys notation. The notation system does not require the annotator to use all the HamNoSys phonological sub-types and the annotation is usually task-specific [112].

While manual annotation of sign language resources requires linguistic skills to identify and translate individual signs and provide free translation for the signed utterances, the annotation process could be accelerated with the help of previously annotated corpora. Ebling et al. [82] find semantic relationships between the signs using a semantic knowledge base. Curiel and Collet take raw signing videos and return the propositional logic that describes the signing [65]. Dubot and Collet attempt to build an automated sign language parser to represent how lexical units come together in a similar way as is done in spoken languages [80]. Koller et al. perform automated mouthing annotations from glosses [143]. Still, these techniques have not been integrated into annotation tools used by linguists potentially due to the insufficient collaboration between researchers in these two fields.

Chapter 3 will make use of ID-Gloss annotation for the BSLCP corpus, while Chapter 6 will rely on annotation of phonological parameters for the OODT corpus. Chapters 4, 5, and 7 will rely on the raw video footage.

2.3 Sign Language Modelling

Modelling is a process of turning information into knowledge that can be understood by humans and machines where a model is described by variables and their correlations [109]. Finding parameters of a complex model through data-driven optimisation requires iterative approaches by sampling the data and using the samples to estimate the parameter space [28]. The landscape of the parameter space grows as the complexity of the model grows and if large amounts of data are at hand, then the model should be complex enough to capture the essential variance in the data [24]. This chapter shows which sign language data has been modelled previously and how, but as important background, transfer learning techniques will be discussed that will be used extensively in this thesis.

Transfer learning attempts to speed up the search for parameters by optimising the parameters of the same model trained on similar data [202]. In spoken languages, for example, cross-domain adaptation of language models is used in the language modelling literature [69, 170]. Models are usually trained on some specific domain that consists of a specific topic and genre. For example, in a restaurant domain when a new type of a restaurant is created then the system needs to be able to adapt and be able to understand and discuss this new type of the restaurant. Unfortunately, it is nearly impossible to train a model for all possible configuration of topics and genres. When these change, re-training of the model is required. In neural networks, model-based adaptation to the new domains, on the other hand, is achieved either by fine-tuning or the introduction of one or more adaptation layer [268]. Fine-tuning involves further training the already pre-trained model using data from the new domain. The intuition behind fine-tuning is that it is much quicker to learn new information using related knowledge. The adaptation layer approach incorporates new knowledge by re-training only the adaptation layer, whereas the rest of the model remains exactly the same as if it was used in the original domain and acts as a feature extractor for the new domain [69]. Transfer learning has been applied to sign language modelling for various purposes to demonstrate that the method is suitable for the task due to the lack of substantial domain-specific sign language data. Transfer learning has been successfully applied to image or video frame recognition to recognise hand shape using convolutional neural networks [155] in the context of static pose estimation [100] and classification of finger-spelled letters in ASL [99, 132, 45, 187].

2.3.1 Spatial Segmentation

Segmentation of a scene refers to the identification of objects and detection of movement [116]. The objects of interest can either be recognised by a pre-trained system using transfer learning [105] or primitives that may belong to these objects such as lines or corners [115]. Movement can be detected by tracking objects or primitives across frames where tracking can assist object detection, and vice versa. For example, knowing the object, its trajectory could be predicted [16] or the object could be predicted knowing the trajectory [153]. Tracking is sometimes employed to support spatial segmentation as the object of interest may disappear at certain times, commonly due to occlusion [54]. Generally, object identification in an image is performed using regression at each spatial location and scale [224, 105, 114, 166].

From sign language grammar, described in Section 1.2, it is now known that signs consist of manual and non-manual phonological parameters. Manual parameters have attracted the most attention due to their dominant significance in the phonology of sign language [34] by estimating the skeletal configuration for hand shape [137] and hand location classification [264]. One of the sign language research directions in computing focuses on applying computer vision techniques to pose estimation and tracking [185, 184]. Depth sensors, such as Kinect, Leap, motion capture, or their combinations have been helpful in making use of skeletal data for the sign language research [150, 262, 85, 252] at the expense of either requiring a projection of light or being too invasive and affecting the naturalness of signing [49, 261]. Therefore, non-intrusive, markerless, and passive sensing in the form of a single camera has gained more interest for its low cost and versatility [49].

Although cameras can provide rich information about a scene, the information obtained from a camera has to be processed in order to acquire the skeletal data [184]. In Appendix A we show one such application of computer vision technique by the author that uses AdaBoost [97] for classifying Haar features [250] to detect face and upper body and then use background subtraction technique to identify moving limbs that originate from the upper body. One such recent general framework is *OpenPose* [39, 227, 40, 257], which has gained a lot of attention in both computing and linguistics communities who do research on sign language. The library detects 2D or 3D anatomical key-points, associated with the human body, hands, face, and feet on a single image. The library provides 21 (x,y) key-points for every part of the hand, 25 key-points for the whole body skeleton, and 70 key-points for the face. In addition, the library provides a confidence value for every (x,y) pair. The OpenPose library has been used extensively in this thesis in Chapters 4, 5, 6, and 7 as it provides skeletal data similar to the accuracy of the active sensors such as Kinect or Leap.

Despite the use of the skeletal data in classification of phonological parameters, other works rely on techniques such as skin/glove colour segmentation [230, 215, 17], shape descriptors [269, 165], or learned convolutional filters [142].

2.3.2 Temporal Segmentation

Continuous signing consists of individual signs, similar to how sentences consist of words in spoken languages. There are two schools of thought regarding the individual signs and the relationship of movement between them [113]. The first school of thought states that individual signs and the transition between the signs (epenthesis) are coupled together, while the second school of thought indicates that the epenthesis does not belong to the signs themselves and therefore carries no lexical meaning. In the sign language grammar section (Section 1.2), it was said that manual phonological parameters contribute the most to signing and that the only ‘punctuation’ is the presence of epentheses. This thesis follows the idea that the individual signs and the movement between the signs could be separated. This means that there is less variation between signs with the same meaning after the movement between the signs is discarded, but there is still variation present due to co-articulation. This is useful in modelling the individual signs as it means that there is less noise for models to be confused by.

Table 2.3: Reported temporal segmentation results (in accuracy %) applied to continuous signing on different datasets using two different methods with the one tracking the hand movement trajectory and another noting the handshape change.

Data set	Model	Accuracy (in %)
ECHO [64]	Speed and trajectory discontinuity [110]	79
Sign3D [103]	Hand shape transitions [188]	80

In sign language research in computing, epenthesis modelling relies on hand speed or trajectory during signing as seen in Table 2.3. However, detecting epenthesis in continuous signing is non-trivial and could be subjective [188]. The Hold-Movement-Hold model for individual sign modelling was proposed by Liddell and Johnson [163]. Another approach by Choudhury et. al. uses the observation that hands move slower during signing than during motion epenthesis [50]. Similarly, Han et. al. use signing speed for segmentation paying attention to signing trajectory discontinuities [110]. Similar to Choudhury et. al., motion epentheses are identified by looking at the distance travelled by each hand during an interval in Mocialov et. al. [184] as can be seen in Appendix B. The authors identify sign boundaries, group the signs that have the same ID-Gloss and then classify individual signs. The same approach will be used in Chapters 4, 6, and 5. An alternative approach to temporal segmentation looks for changes in hand shapes with Naert et. al. segmenting continuous signing videos when there is a change in the handshape phonological parameter [188]. Such techniques have been employed extensively in sign language research in computing as discontinuities during signing have been suggested by the linguistic community as a good segmentation signal [163]. Analogous to the keyword spotting in automatic speech recognition, [89, 263] propose to segment by spotting a finite list of signs in continuous signing. Another method for temporal segmentation during manual annotation was noted in [58], saying that the human transcribers use the change in direction, orientation, and/or handshape as a cue for segmenting signs by only looking at one specific phonological parameter. This work will rely on the observation that the speed of epenthesis differs from the speed of individual signs.

2.3.3 Modelling Glosses

Another research area in sign language in computing uses natural language understanding techniques applied to sign language modelling at the gloss level often for translation purposes to written or spoken languages, as it is believed that more precise language models positively affect translation models despite the fact this idea has been challenged [2].

Table 2.4: Reported gloss modelling results (in perplexity (PPL)) on different datasets, where lower PPL signifies a better model

Data set	Model	PPL
RWTH-Phoenix [234]	Fourgram with Kneser-Ney discount smoothing [231]	53.3
RWTH-BOSTON-104 [76]	Trigram with modified Kneser-Ney discounting with interpolation [76]	25.1
LSE [73]	Maximum a-posteriori (MAP) [9]	9.04
TSL [260]	Grammar tree and a variable n-gram language model [260]	3.97

Research in sign language modelling predominantly employs n-gram models [43, 94, 174, 234, 76, 260] as can be seen in Table 2.4 despite the availability of many alternatives for language modelling, such as variations of count-based methods [46, 213, 171, 37, 108], hidden Markov models [77, 76], decision forests [93], and neural networks [69, 180]. Chapter 3 will make use of both the n-gram models and the neural networks. The major concern is that majority of the research on sign language modelling uses small amounts of data in the range of 150 [73] to 1000 [260] sentences. Successful language modelling and eventual translation, however, requires many more examples [18]. No significant attempts have been made so far to adopt models that have already been established or just gaining momentum in text and speech for symbol-based sentiment analysis [48], topic classification [157], or summarisation [66].

It is important to point out that there is an obvious disconnect between glosses and signing in that glosses represent the meaning of signs in spoken language, but not the execution. That is why glosses lie between free translation and visual signing with the visual signing having the most information about the execution of a sign and free translation having the least information about the execution of a sign. Therefore, the meaning of what is being signed may be lost when modelling glosses since the gloss annotation is often scarce and does not convey all the information about the signing. To incorporate some additional information on the sign execution into language models, non-manual phonological parameters could be added to gloss modelling as was done in [217, 174, 271, 232].

2.3.4 Phonological Parameters

Multiple previous works have modelled individual phonological parameters taking Stokoe et al.'s visual taxonomy as the basis [236]. Similar methods for modelling individual phonological parameters and constructing linguistic feature vectors have been used for recognising individual signs by Bowden et al. [29]. Their work operates on handshape, location, and movement by modelling them as Markov chains using a single example. This work only describes accuracy for the handshape classifier as 75% for eight handshapes from BSL. A sliding window is used to get the highest activation throughout the temporal dimension, as a way to spot and classify individual signs during continuous signing. Cooper and Bowden [55] modelled location, movement, and hand arrangement and called them the sub-sign units. They have shown that tracking of the phonological parameters does not contribute to the sign classification accuracy, with location accuracy at 31% using the AdaBoost [97] classifier after applying the grid on the image to see which part of the grid fires when a hand is close to a body part. Cooper et al. [57] relied on handshape, location, movement, and hand arrangement in their work on recognition of the individual signs in BSL using the random forest model [161] trained on the histogram of oriented gradients (HOG). They report a confusion matrix for handshape without reporting the overall accuracy. The confusion table shows quite high recognition accuracy of three out of the twelve hand shapes, but quite poor performance for the other three hand shapes. They also showed that location information contributes the most to the recognition of a sign while handshape has the least effect. Buehler et al. [36] resorted to movement, handshape, and orientation while matching the phonemes to find similar signs. In [35], Buehler et al. used location and handshape in the multiple instance learning problem. Koller et al. [142] focus on three data sets (Danish, New Zealand, and German Sign Languages) and sixty hand shapes. The model is a chain of convolutional neural networks pre-trained on the ImageNet data [228] and fine-tuned on their handshape data using the transfer learning technique. After fine-tuning the pre-trained model on one million cropped images of sixty different hand shapes, the model achieves 63% accuracy. This work will model phonological parameters in Chapter 6 and make use of phonological parameters in Chapter 7 for the purpose of comparing sign languages and in Chapter 5 for comparing phonemes using phonological parameters.

Table 2.5: Reported modelling of phonological parameters (in accuracy %) on different datasets using different data types (2D and 3D). Literature on modelling handedness and orientation is more difficult to find. Most of the works are focused on modelling handshape.

Phonological Parameter	Result (in %)	Data set
Handedness	—	
Handshape	75 (2D) Bowden et al. [29]	Single signer, 49 isolated signs
	63 (2D) Koller et al. [142]	RWTH-PHOENIX-Weather [95] & SIGNUM [253]
	82 (3D) Dilsizian et al. [74]	Synthetic Dataset [74]
	99 (3D) Demircioglu et al. [70]	18 Handshapes [70]
	68 (2D) Kim et al. [164]	RWTH-BOSTON-104 [75]
Orientation	—	
Location	31 (2D) Cooper and Bowden [55]	Single signer, 164 isolated signs [128]

From Table 2.5 it can be seen that those who model phonological parameters either focus on one parameter or do not report recognition accuracy of the individual phonological parameters. It is common to see research that performs classification or clustering of the signs directly without considering that the phonological parameter classification error propagates into the classification of individual signs. As individual phonological parameters are executed in parallel during signing, there is a chance that they are also co-dependent. There is very little research that pays attention to relationships between phonological parameters because of the tendency to attempt the modelling before sufficient understanding of the data. Awad et al. [8] comes close by sharing parts of a model among different phonological parameters, but they do not tell explicitly which parameters could benefit from the shared features.

One side note about phonological parameter modelling, Gineke et al. question whether all phonological parameters are equally important for sign recognition and how much variation is acceptable for a sign to be recognised [104]. In fact, Ojala et al. have shown that the index finger is the salient finger during signing and determines the speed and amplitude of signing with other fingers following the motion of the index finger [193]. This theory is supported by Ann [4], who shows that physiologically it should be the case that not all fingers are dominant during signing, which makes it applicable to research in sign language recognition. There has been little to no research in computing that attempts to ablate data and compare accuracies of resulting methods. If such studies in the future show that not all data is required when training successful models, then reducing the data could potentially accelerate the training, improve performance of such models, and reduce the amount of space required for data storage, considering that the sign language data is already relatively complex.

2.3.5 Signing

Sign language modelling is a complex task that may involve spatial and temporal segmentation (Sections 2.3.1 and 2.3.2) as well as the modelling of the phonological parameters (Section 2.3.4) with the purpose of recognising the individual signs [184], sign spotting in continuous signing [194], or modelling sequences of phonological parameters [98] with each set of phonological parameters corresponding to a frame in a signing video. The research in sign language modelling is split into isolated and continuous sign modelling and the major challenge comes from trying to model for signer-independent cases, where the trained model would perform well on unseen subjects during the model testing and inference phases [49].

Table 2.6: Reported results on modelling of isolated signs (in accuracy %) on different datasets, where *SI* stands for Signer-Independent modelling. The table shows which features extracted from the raw videos and which methods were used for modelling isolated signs

Data set	**Features**	**Method**	*SI**	**Accuracy (in %)**
DEVISIGN [44]	Sparse coding for handshape and hand movement	SVM [267]	✓	64
SMILE [81]	Handshape and hand movement descriptors	HMM [244]	✓	66.8
DEVISIGN [44]	Handshape and hand movement descriptors	iRDML [266]	✓	56.85
CSL Cyberglove [98]	hand shape, hand position, and hand orientation	Fuzzy Decision Tree [98]	✓	91.6
ASLLVD [190]	Learned CNN features	CNN with LSTM [15]	✗	91
A3LIS-147 [86]	Hand Centroids, non-manual features, hand orientation, etc.	HMM [87]	✓	48.06
NGT [63]	Pose key-points and non-manual features	Stacked LSTMs [184]	✗	80
MS-ASL [126]	Raw frames	I3D [126]	✓	57.69

Isolated signs with a similar gloss are grouped together and the aim is to train a model that would output a gloss given a sequence of frames of a single sign ignoring the existence of epentheses [56], as also can be seen in Appendix B, where we train a model to recognise individual signs using the kinematic features. From Table 2.6, it can be seen that in sign language modelling, features from each frame are extracted and modelled using variety of models. In [98], self-organising maps (SOM) [140] are used for extraction of features before modelling them with hidden Markov Models (HMM) [208]. The approach goes further by proposing to use a hierarchical decision trees to accelerate the classification process by first classifying whether a sign uses one or two hands, then classifying the hand shape, extracting the features and modelling the signs using combined objective function. Fagiani et al., on the other hand, engineer a range of manual features for both hands and the face, and models those features using HMMs [87]. It is believed that the choice of the right features would benefit recognition models [272]. That is why researchers either let the method learn the features or manually engineer the features themselves, with the latter yielding typically better performance, as can be seen in Appendix A, where we used separate classifiers to detect features. Generally speaking, features are extracted to reduce the complexity of the data and create a representation of a scene, discarding all the features that do not yield better results for the higher-level tasks. Letting experts design representations useful for the higher-level task is labour intensive, prone to error, and lacks generalisation. Whereas learned features are usually more representative for the higher-level task at hand [19]. For example, Yin et al. try to capture signer-invariant individual sign features by generating a sparse coding for hand shape and movement phonological parameters and train support vector method (SVM) [248] to recognise individual signs [267], reporting 64% accuracy on signer-independent sign recognition dataset with 1000 signs. Mocialov et al. extract and augment skeletal features by adding small random perturbations [184]. The augmented data is then modelled using stacked recurrent neural networks with long short-term memory (LSTM) [118] as can be seen in Appendix B. The results show 80% recognition accuracy on the signer-dependent data with 40 signs. However, the accuracy decreases twice as facial features are added. Bantupalli and Xie train LSTM on learned features from the pre-trained Inception model [241] on isolated signs performed by one signer in varying lighting conditions involving frame-based augmentation such as rotation and scaling [15]. Tornay et al. model handshapes and hand movement encoded as velocity between three frames using HMM [244]. Yin et al. perform classification of signs by matching their trajectory and handshape descriptors to a space of sign references, where the sign reference space is learned by maximising the distance between different signs and minimising the distance between the same signs, thus reducing the inter-signer variance [266]. Finally, Joze and Koller use raw frames to do video classification using 3D CNN [126]. These attempt showed us that modelling individual signs is futile as there is potentially an indefinite vocabulary that needs to be modelled. However, better features can be learned from the data for any downstream task.

Commonly, in the NLP community, Word Error Rate (WER) metric is used when comparing different methods for recognition of continuous signing. WER measures the ratio of sign insertion, substitution, and deletion errors compared to the ground truth [205].

Table 2.7: Reported results on modelling of continuous signing using word error rate (WER) metric on different datasets, where *SI* stands for Signer-Independent modelling. The table shows which features extracted from the raw videos (if any) and which methods were used for modelling continuous signs

Data set	**Features**	**Method**	*SI**	**WER**
RWTH-PHOENIX-Weather 2014 [95]	Raw Frames	Re-Sign [145]	✗	26.8
SIGNUM [254]	Raw Frames	Re-Sign [145]	✗	4.8
SIGNUM [254]	Raw Frames	CNN-HMM [144]	✗	7.4
RWTH-PHOENIX-Weather 2014 [95]	Raw Frames	CNN-HMM [144]	✗	38.3
RWTH-PHOENIX-Weather [94]	Raw Frames	CNN-HMM [144]	✗	30
RWTH-BOSTON-104 [76]	Raw Frames	HMM [79]	✓	17
RWTH-PHOENIX-Weather 2014 [95]	Raw Frames	CNN-LSTM-HMM [141]	✗	26

Continuous sign language modelling techniques, on the other hand, take note of epentheses and incorporate them in recognition systems in one way or another [56]. The state of the art approaches seem not to differ much from the approaches for isolated sign recognition (see Table 2.7). In particular, the HMM method seems to be a common approach to model learned features. Koller et al. use a pre-trained CNN and embed it into the HMM by first training the CNN on handshapes [144]. The authors acknowledge that basing the recognition of signs on a single hand is insufficient for recognition of continuous signing. Iterative re-alignment of weakly annotated sequences of frames is performed by Koller et al. by training pre-trained CNN with a bidirectional LSTM during maximisation step and embedding the classifier into the HMM during the expectation step [145]. Koller et al. train CNN that is embedded in HMM for handshape and non-manual phonological parameters, as well as the glosses and models them as multi-stream HMM with the final feature fusion at the end [141]. Dreuw et al. explore various features and find that combining global features with hand positon, trajectory, and velocity features gives the best performance when using HMM for sign modelling [79].

From the literature, it can be seen that most of the research on signer-independent sign language recognition has been done in the context of recognising isolated signs with significant performance drop compared to signer-dependent research. Work on continuous sign language recognition still uses signer-dependent evaluation for the most part, mainly due to the lack of large annotated corpora with only recent attempts at extending existing corpora for sign language translation purposes [95].

2.4 Sign Language Mining

Data mining is the process of knowledge discovery from traditionally large amount of data with Leskovec et al. [158] listing methods that were developed due to the inability to manually process the data due to its size and often the rate of the new information being generated from many sources such as user-generated data on social networks or sensory data used for monitoring. The authors state that knowledge discovery is performed by applying pre-trained models to the data, where the model is trained on the data from the same distribution. Data itself is usually reduced to keep most prominent features that describe each data point. Such features can either be engineered by studying the data beforehand or learned through summarisation (e.g. determining cluster centres or embedding data into simpler data structures). The fundamental data mining problem is finding similarity between the data points, as each data point can have many dimensions and there may be no apparent metric to measure similarity between the data points. This becomes even more complicated when there is large amount of data and it cannot be compared to each another.

Section 1.2 discusses that signs in sign language are sequential for each hand, with every sign consisting of one or more phonemes, and every phoneme consisting of phonological parameters. There is a possibility of two hands signing different signs at the same time. In the data mining community, sequences of event-intervals describe events that are not instantaneous but have a time duration with Kostakis and Papapetrou [147] telling that the challenge in data mining is to design a similarity measure to compare two such events and once a measure is in place, it would allow clustering. In sign language, as was mentioned above, sequences of temporal intervals can be formed by events corresponding to individual phonological parameters. Multiple distance metrics have been proposed for the event comparison including dynamic time warping (DTW) [102], Euclidean distance [135], graph matching [101], or combinations of metrics [251]. DTW has also been used in sign language community for comparison of two signs [5]. The data mining community is concerned with the more general case of event matching, as two events may overlap and have the same or different labels. However, in sign language as described in Section 1.2, phonemes signed by each hand cannot overlap since individual signs cannot overlap. Due to the search for a more general solution applicable in the data mining community, techniques such as turning events into symbolic representation (e.g. strings) and applying Levenshtein distance are not applicable [147], but could be applicable to sign language data when signing of an individual hand is viewed in isolation.

Surprisingly, mining and data analysis in sign language research does not appear often in literature as opposed to sign language modelling due to the limited amounts of data available and the data mining techniques designed to tackle large amount of data. Nevertheless, mining for variations in sign language has attracted researchers previously with Börstell and Östling using available data to search annotation for lexical variations across different signer groups, concluding that certain signs are signed more often than others depending on the gender of the signer [26]. Similarly, Crasborn et al. compare mouth gestures in three sign languages and compare the mouth gestures against those in spoken languages [61]. Papapetrou et al. performed associative co-dependence learning and found co-dependences of certain manual and non-manual parameters in ASL that were already known to linguists [204]. Östling et al. provide quantitative way to prove the existence of iconicity (e.g. the choice of phonological parameters in signs is not arbitrary) across different sign languages [197]. They show that the choice of two hands for signing prevails across sign languages for plural signs and hand location correlates with signs associated with that body part.

Clustering has been used in sign language research for various purposes. For example, Oszust and Wysocki perform multi-objective optimisation using evolutionary algorithms to find optimal phoneme boundaries and assign phonemes to clusters by optimising distance metric based on the DTW distance [198]. Nayak et al. cluster invariant parts of a sign using DTW distance metric and hand colour and edge information [189]. Tsumoto and Hirano captured structural similarity between signing trajectories of selected signs even in the presence of inter-signer signing differences [246]. They used hand location information and compare segments based on their gradient, angle, and velocity. Kostakis and Gionis encode data into graphs and present an algorithm for sequence matching in data that could be applicable to sign language [148]. However, they do not perform experiments on actual signing data.

Chapter 5 proposes a linguistically-inspired distance metric that is based on phonological parameters and is used to compare phonemes and show how the same distance can be used to compare signs and phrases.

2.5 Discussion

Diverse research efforts in sign language modelling and sign language mining have contributed to sign language research in computing and Linguistics, allowing modelling of signs and translation more recently (as discussed in Section 2.3), as well as finding patterns in video data to support linguistic hypotheses (as discussed in Section 2.4). The biggest challenge for data-driven modelling approaches is that of the insufficiency of sign language resources and no easy way to compare the results. On one hand, looking at the past work it becomes apparent that the data is the issue in sign language research in computing and linguistics in that the amount of video data is sparse and annotations are often incomplete, as in the case of the often non-existent epenthesis annotation or annotations specific to narrow research goals. However, recent research has found new ways to provide increasing amounts of data for machine learning methods by utilising online resources [126, 183]. On the other side of the spectrum, such techniques as transfer learning and data augmentation, can strengthen sign language models when resources are insufficient [226, 255], for example with image recognition models pre-trained on ImageNet [71] and used as feature extractors or pose estimation models [184, 142, 197]. Nevertheless, there must be a limit to how much knowledge can be transferred and how much data can be augmented or generated to keep the models resilient to input fluctuations, yet precise at their predictions especially in the signer-independent case. Another approach to regularise models could be via preliminary data analysis and encoding of the discovered patterns into the model. By doing so, the optimisation space could be reduced and the initialisation of model parameters could be closer to the true optimum just like with transfer learning techniques.

As for the existing resources, some of which are mentioned in Section 2.2, there is no easy way to tell which of the available corpora is more appropriate than another, which often results in tremendous efforts spent on insignificant amounts of data collected for individual research projects. Moreover, corpora that is collected in controlled conditions does not reflect the reality of real live signing. Therefore, the data that has been collected from the real conversations in various settings in the real world could prove to be more useful when developing systems that work in the real world. Only very recently, there have been initial efforts at targeting these noisy resources [126, 183].

Very few methods and techniques have been applied to sign language resources in the data mining community often using sign language as an example use-case scenario without testing developed algorithms on the actual signing data, as mentioned in Section 2.4. Data mining and modelling are supposed to compliment each other. However, from what can be seen in literature, there could be more overlap between the two fields when it comes to research in sign language. Sign language resources will most certainly pose new challenges to the data mining community, especially in clustering or modelling co-dependencies in not only visual, but also linguistic data.

The above reasons lead to the motivation of this work that investigates data-mining techniques that are efficient and lend themselves effectively to sign language modelling.

Chapter 3

Modelling Glosses by Transferring Knowledge from English Language Models

From Section 2.2, it is known that glosses and ID-Glosses are a common annotation schemes for sign language with ID-Glosses providing labels for each lexical unit that is being used during signing, independent of the context (see Figure 1.3 caption, where gloss and ID-gloss are the same). Language models capture the distribution of similar lexical units at any given place in a spoken utterance, given the previous lexical units used in the same utterance [258]. These models capture the context and serve as building blocks for speech recognition, translation, and generative models. This chapter explores modelling of BSL ID-Glosses with the help of an English language model using a transfer learning technique, which is a more recent technique for model adaptation as opposed to methods mentioned in [18]. Furthermore, the method presented here would enable one to leverage the large amounts of data available for written/spoken language for modelling the low-resource sign language data. First, both BSL corpus and the English dataset will be presented and methods for pre-processing discussed. Second, a standard method for language modelling will be presented.

Finally, the results section will compare and contrast results using each method. The discussion section at the end will show what limitations such an approach has. In summary, this chapter attempts to answer the first research question (RQ1) from Section 1.3, which asks:

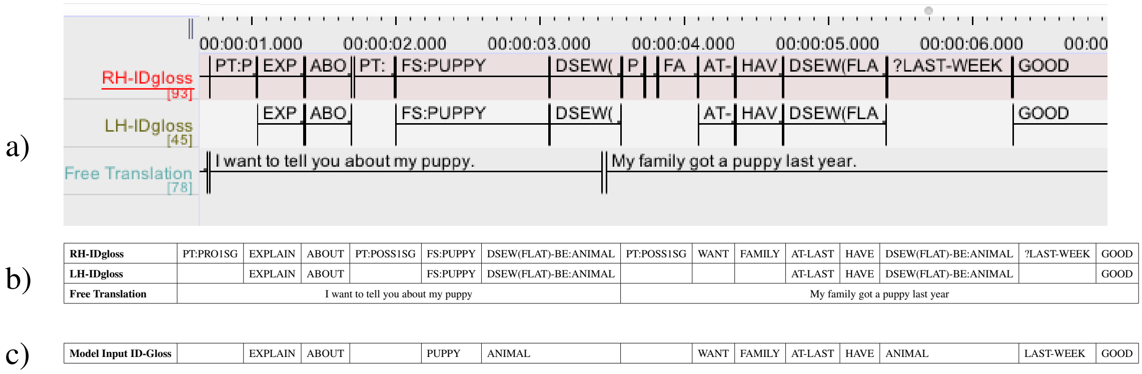
“ Can a small number of BSL sentences be modelled at the gloss level using written English language models, knowing that the syntax of the two languages is different? ”

3.1 Data Pre-processing

The BSLCP corpus (Table 2.1) and the pre-processed Penn Treebank (PTB) corpus [180] were chosen for this research. The monolingual PTB dataset consists of telephone speech, newswire, microphone speech, and transcribed speech in English language. The dataset is preprocessed to eliminate letters, numbers, or punctuation and was used by Mikolov et al. [180]. The BSLCP corpus contains video conversations among deaf native, near-native and fluent signers across the United Kingdom. Almost all of the approximately one hundred recorded conversations come with thirty second annotations at the gloss level for each video [219].

Figure 3.1: BSL Corpus Project Sample Video Snippets showing participants from different recordings (Source: https://bslcorpusproject.org/cava/)

All recordings of the signers were made using up to four standard video cameras with a plain backdrop, as shown in Figure 3.1, to provide a full body view of the individuals as well as views from above to capture the use of signing space. The conversations between the signers included signing personal experience, anecdotes, and spontaneous conversations [219]. The selected BSL data was narratives between two participants, where one person had to think of a topic to sign about to another participant.



RH-IDgloss	PT:PRO1SG	EXPLAIN	ABOUT	PT:POSS1SG	FS:PUPPY	DSEW(FLAT)-BE:ANIMAL	PT:POSS1SG	WANT	FAMILY	AT-LAST	HAVE	DSEW(FLAT)-BE:ANIMAL	?LAST-WEEK	GOOD
LH-IDgloss		EXPLAIN	ABOUT		FS:PUPPY	DSEW(FLAT)-BE:ANIMAL				AT-LAST	HAVE	DSEW(FLAT)-BE:ANIMAL		GOOD
Free Translation	I want to tell you about my puppy						My family got a puppy last year							

Model Input ID-Gloss		EXPLAIN	ABOUT		PUPPY	ANIMAL		WANT	FAMILY	AT-LAST	HAVE	ANIMAL	LAST-WEEK	GOOD

Figure 3.2: A snippet of BSL Corpus Annotation in ELAN for two sentences. a) BSL Corpus Annotation in ELAN including annotation in ID-Gloss for both hands and the free translation; b) Table shows full text of the annotated ID-Glosses for the two first sentences from the ELAN annotation (since not all annotations are visible in the a); c) ID-Glosses that were used for BSL modelling for the two sentences in this example.

The corpus is annotated with ID-Glosses, taken from BSL SignBank in ELAN as shown in Figure 3.2 a) and Figure 3.2 b) shows all the ID-Glosses of the first sentence. In the corpus, the ID-Glosses are identified throughout the videos for both left and right hands because sometimes two different signs can be signed at the same time with different hands. Apart from the ID-Gloss, annotations include corresponding free English translation preserving the meaning, which is split into sentences as seen in Figure 3.2 a). Figure 3.2 c) shows ID-Glosses that are taken for modelling BSL from the first two sentences. Some ID-Glosses were ignored to match the vocabulary of the PTB corpus in order to apply transfer learning successfully. For example, in Figure 3.2 b), right-hand ID-Glosses identify the following order of the signs: good, explain, about, puppy, etc. ID-Glosses, like PT:PRO for pointing signs or PT:POSS for possessives and similar are excluded, which are explained in more detail in Fenlon et al. [92]. The exclusion is done based on the fact that these language constructs refer to the execution of signs and do not resemble any words in English language. Since the ID-Gloss annotation does not include explicit punctuation, it is impossible to tell where a signed sentence begins and where it stops. To overcome this limitation of the ID-Gloss annotation, Free Translation annotation was used, which gives the boundaries of sentences in videos. Later, the extracted ID-Glosses were split into sentences using these sentence boundaries. ID-Glosses were left by the end of the pre-processing stage in the order that the corresponding signs were executed in the video, split into sentences.

As a result, 810 nominal sentences were extracted from BSL corpus with an average length of the sentence being 4.31 signs, minimum and maximum lengths of 1 and 13 signs respectively. A monolingual dataset has been created with the extracted sentences. As obtained from the PTB dataset [176], the English language corpus has 23.09 words on average per sentence with minimum being 3 and maximum 84 words per sentence. The pre-processed BSL corpus has a vocabulary of 666 words, while the PTB dataset has a vocabulary of 10002 words.

Both monolingual datasets were split into training, validation, and testing sets as required for training and evaluation of the statistical models. Both datasets were split using ratio 85:15. The smaller subset, in turn, was split 50:50 for validation and testing for the two datasets. Speaker independence was not ensured during the split due to the low number of examples collected.

3.2 Methodology

Perplexity measure is a common measure for language models that measures the expected word error rate and it was used for evaluation and comparison purposes of different models with better models having lower perplexity. Entropy was used in calculating the perplexity as there is a relationship between the two in the form $e^{Cross-Entropy}$ as used in [20] and the discussion about the relationship between perplexity and entropy can be found in [258]. The formula approximates geometric average of the predicted word probabilities on the test set. Out-of-vocabulary (OOV) was explicitly modelled with an $< unk >$ placeholder in all the experiments.

3.2.1 SRILM Linear Model Interpolation

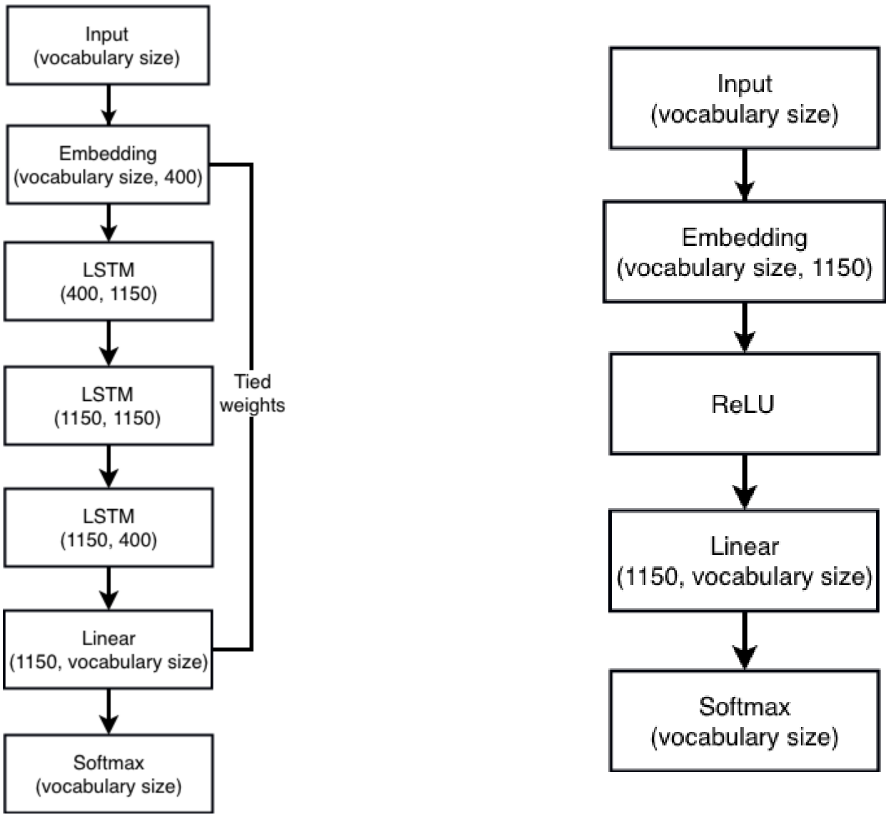
Using SRILM Toolkit, two language models are trained, first with the PTB dataset and another with BSL corpus and both models evaluated on to the test data. To perform the interpolation, both trained models are interpolated using the *mix -lm* parameter in SRILM Toolkit. Further, *lambda* parameter, used with the *mix -lm* parameter that indicates the weighting in favour of one or another model when model interpolation is used. A similar approach was applied to English using simplified English Wikipedia articles in [133], with the aim of achieving lower perplexity than the simplified English language by interpolating the two models.

3.2.2 Neural Models

For model comparison two methods were used, Stacked Long Short Term Memory (LSTM) and Feedforward (FFNN) neural network types to train BSL language models. All models are implemented in PyTorch with weight decay recurrent regularisation scheme for the LSTMs, which is important for overcoming commonly known LSTM model generalisation issues [176, 175]. The FFNN model, on the other hand, has no regularisations as it is less susceptible to overfitting due to a much smaller parameter space.

Parameters that were modified to achieve the lowest perplexity were the input size of the overall input sequence for the recurrent neural network, batch size, learning rate, and the optimizer. The parameters were selected using the grid search approach using perplexity evaluation metric. As a result, for the stacked LSTMs, input size was set to 5 tokens, batch size was set to 16, discounted learning rate was set to 30, and the optimizer was set to Stochastic Gradient Descent. In case of the FFNN, input was set to 5 words, batch size was set to 16, discounted learning rate was set to 30, and the optimizer was set to Stochastic Gradient Descent. All the neural models were trained for 100 epochs.

In the case of the neural networks, the sequences of words were tokenised and the tokenisation was stored to ensure the same tokenisation during the transfer learning phase.



a) Stacked LSTMs model

b) FFNN model

Figure 3.3: The two types of neural models (Stacked LSTMs in a) and FFNN in b)) used in modelling BSL.

3.2.2.1 Stacked Long Short Term Memory Networks (LSTM)

Figure 3.3 a) shows the architecture of the stacked LSTM model. The model consists of an embedding layer of 400 nodes, which turns words into a vector of real numbers. Secondly, three LSTM layers with 1150 nodes each are stacked vertically for deeper feature extraction. Thirdly, the linear layer downsizes the stacked LSTMs output to the vocabulary size and applies softmax normalisation. The weights of the embedding and the linear layers are tied. This means that the two layers share the same weights, which reduces the number of parameters of the network and makes the convergence during training faster. The same architecture was used in [176] to model the PTB dataset, reporting 57.3 perplexity, utilising cache in the model from recent predictions.

3.2.2.2 Feedforward Neural Networks (FFNN)

Figure 3.3 b) shows the FFNN model architecture. The model does not have the stacked

LSTMs layers. Instead, the stacked LSTMs are substituted with one hidden fully-connected rectifier layer, which is known to overcome the vanishing gradient problem. The weights of the embedding and the output layers are not tied together. Similar architectures have been used for language modelling in [154], [181], and [67] with the hidden layer having different activation functions with the PTB dataset being used in [6], reporting 137.32 perplexity.

3.2.3 Training Models

Transfer learning was achieved with both fine-tuning and substitution. Both FFNN and LSTM were trained on the PTB dataset and then either fine-tuned or the last layer was substituted with the new adaptation layer, freezing the rest of the weights, and further training on BSL dataset.

To achieve fine-tuning the best model is saved after training of both the FFNN and the stacked LSTMs on the PTB dataset. Then the training is restarted on BSL corpus, having initialised the model with the weights, trained on the PTB dataset.

To perform layer substitution as a transfer learning approach, the same first step as with the fine-tuning is repeated and the model, trained on the PTB, is saved. When the training is restarted on BSL dataset, the saved model is loaded and the last linear layer is substituted with a layer that has as many nodes as BSL vocabulary. Later, all the weights of the network are locked and will not be modified during the optimisation. Only the weights of the last substituted layer will be modified. This method uses the pretrained network as a feature extractor and only modifies the last layer weights to train the model on BSL dataset.

3.3 Results

This section is split into three parts. First, results without transfer learning are presented, namely both the FFNN and the stacked LSTM models trained and tested on the PTB dataset as well as the same models trained and tested on BSL. Second, results from two interpolated SRILM models are shown, one trained on PTB and another trained on BSL. Third, results with the transfer learning are shown with both FFNN and the stacked LSTMs models trained on the PTB dataset and then fine-tuned and tested on BSL.

To show that the word order of the two languages is different, a model trained on one language is applied to another language and vice versa. As a result, the n-gram model trained on English language and applied to BSL scored 1051.91 in perplexity using SRILM toolkit [237]. Conversely, a model trained on BSL has been applied to the English language and scored 1447.23 in perplexity. As expected, the perplexity is high in both cases, which means that the probability distribution over the next word in one language is far from the distribution of words in the other language. Such result is not surprising given that the two languages have different syntactic structures.

3.3.1 Results Without Transfer Learning

Table 3.1: Achieved perplexities on both the PTB or BSL test sets using models trained and tested on the same corpus. For example, the second column (PTB) shows results of the models trained and tested on the PTB dataset, while the third column (BSL) shows results of the models trained and tested on BSL dataset. OOV stands for out of vocabulary words that appear in test set, but are not present in the traning set.

Method	**Penn Treebank** **(PTB)**	**BSL** **Corpus Project**
FFNN	190.46	**258.1**
Stacked LSTMs	**65.91**	274.03
OOV	6.09%	25.18%

Table 3.1 shows perplexities on two datasets with two statistical models. From the table, it can be inferred that the trained models on the PTB dataset have lower perplexity than the same architectures trained on BSL dataset. This can be explained by the fact that the PTB dataset has more data than BSL dataset and, therefore, statistical models can generalise better. Furthermore, the amount of data is further reduced in BSL case as the OOV covers a quarter of the overall dataset with OOV being a set of words not seen by the model during training, but present in the test set.

3.3.2 Results for SRILM Linear Model Interpolation

Table 3.2: Perplexities on either the PTB or BSL test sets using SRILM modelling tool, followed by the perplexity on BSL test set by interpolating models, trained on PTB and BSL. For example, the second column (PTB) and the first row (SRILM 5-gram with smoothing) shows results of the model trained and tested on PTB dataset, while the second column (BSL) and the first row (SRILM 5-gram with smoothing) shows results of the model trained and tested on BSL dataset. However, the second row shows results of two models, one trained on PTB and another on BSL, then interpolated, and tested on BSL dataset using SRILM model interpolation method, where both models are required to contribute during the testing step for each example.

Method	**Penn Treebank (PTB)**	**BSL Corpus Project**
SRILM Kneser-Ney Smoothing 5-gram	**141.46**	**269.31**
OOV	6.09%	25.18%
SRILM Model interpolation Kneser-Ney smoothing 5-gram	326.56	
OOV	12.71%	

Table 3.2 shows results of two models. First row (SRILM 5-gram with Kneser-Ney smoothing) shows results when two models are trained for each one language and tested on the same language (BSL or English). The second row (SRILM 5-gram model interpolation with Kneser-Ney smoothing) shows results when two models are trained for each one language, then interpolated with each other, and tested on BSL. From the table it can be seen that using the SRILM toolkit on either PTB or BSL data produces similar results as in Table 3.1. That is, the perplexity on the PTB dataset is smaller than the perplexity on BSL dataset, which can be explained by the data sparsity of BSL dataset. The effect of the model interpolation parameter (lambda) is discussed further.

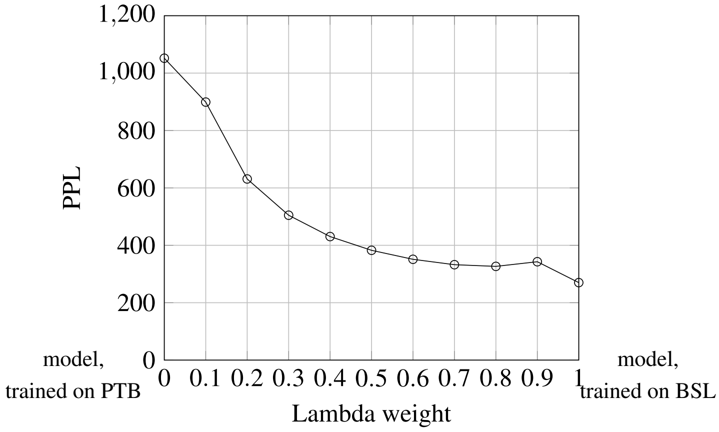


Figure 3.4: Perplexity of the two linearly interpolated models applied to BSL test set. Lambda=0 corresponds to the model, trained on the PTB dataset, while lambda=1 corresponds top the model, trained on BSL dataset

Figure 3.4 shows the effect of lambda parameter for the interpolation on the perplexity. The perplexity evolution is as expected, the model, trained on PTB dataset performs the worst on BSL dataset, while the model, trained on BSL dataset has the lowest perplexity on BSL training set. None of the combinations of the two models result in a superior performance than that of the model trained only on BSL as can be seen in Table 3.2 where perplexity is 326.56.

3.3.3 Results With Transfer Learning

Table 3.3 shows perplexities on the two datasets with two statistical models, applying the transfer learning method described in Section 2.3. From this table, it can be seen that the substitution approach gives very similar results independent of the whether FFNN or stacked LSTMs model is used (123.92 versus 125.32). The best result is achieved with the fine-tuning approach on the stacked LSTMs model, while the higher perplexity result is on the FFNN model with the fine-tuning approach. Similar results have been reported in [120], where fine-tuned GRU performed worse than fine-tuned LSTM model. In addition, the OOV count differs from that of the Table 3.1 due to the fact that a subset of the vocabulary, observed in the PTB dataset during training is then identified in BSL dataset during testing.

Table 3.3: Perplexities on BSL test set after applying the transfer learning for both FFNN and LSTMs models, where both models were pre-trained on the PTB dataset first and then trained further on BSL dataset and tested on BSL dataset.

Method	**Fine-tuning**	**Substitution**
FFNN	179.3	123.92
Stacked LSTMs	**121.46**	125.32
OOV	12.71%	

The main question of this chapter is whether transfer learning is a legitimate method for modelling one language with the knowledge of another, assuming the languages are different, but share some common properties, such as vocabulary. This theory is intuitive and has been discussed in linguistics for spoken languages [129]. In our case, the PTB corpus covers most of the vocabulary found in BSL corpus (12.71% OOV) by the virtue of the gloss annotation of BSL corpus [219]. However, the languages are assumed to be different as they evolved independently of one another [33]. As language models inherently model the order of words and syntactic structure, a simplistic use of spoken language model on BSL is not effective, as evidenced in Table 3.2.

The results obtained are different from reported in similar research. For example, for the FFNN model, Audhkhasi et al. [6] report 137.32 versus our achieved 190.46 perplexity and for the stacked LSTMs model, Merity et al. [176] report 57.3 versus our achieved 65.91 perplexity. This can be explained by the fact that the model training had been restricted to 100 epochs. Further training may further reduce the perplexity to that reported in Merity et al. [176].

From the results, it can be seen that the transfer learning leads to more superior models than the models trained on BSL directly (121.46 from Table 3.3 versus 258.1 from Table 3.1 respectively). Since the quality of the trained models using either of the approaches is similar in the case of the stacked LSTMs model (121.46 versus 125.32 in Table 3.3), the choice between the fine-tuning and substitution can be guided based on the convergence speed. During the substitution, only one layer of the network is replaced with a new one and the rest of the weights in the network are locked, therefore, one set of weights will be optimised. This is in contrast to the fine-tuning method, which optimises all of the weights, which may, in turn, require more interactions, depending on how different the distribution of the new data is.

3.4 Discussion

A transfer learning technique was developed due to the need for large amounts of data for traditional models, which is difficult to obtain. The shared vocabulary between annotated sign language resources and the English language dataset presents the second motivation for the use of transfer learning techniques. Count-based, feedforward, and recurrent neural language models have been evaluated and compared, showing that the recurrent neural language model outperforms the feedforward neural language model by more than 30%. Moreover, the recurrent neural language model outperforms count-based language model by more than 50% in terms of perplexity[1].

Table 3.4: Reported perplexity (PPL) scores in modelling sign language glosses compared to our proposed transfer learning approach.

Data set	Model	PPL
RWTH-Phoenix [234]	Four-gram with Kneser-Ney discount smoothing [231]	53.3
RWTH-BOSTON-104 [76]	Trigram with modified Kneser-Ney discounting with interpolation [76]	25.1
LSE [73]	Maximum a-posteriori (MAP) [9]	**9.04**
TSL [260]	Grammar tree and a variable n-gram language model [260]	3.97
BSLCP [219]	Our proposed method	121.46

Previously reported results have lower perplexity (53.3 [234] and 25.1 [76]) as can be seen in Table 3.4. Using the same methods as the ones reported in the literature when modelling BSL achieves 269.31 perplexity as shown in Table 3.2, which raises a question whether modelling different datasets has an effect on variance in perplexity. Our proposed

[1] Both results are significant at $p = 0.00512 < 0.05$ by the Wilcoxon signed-rank test

approach of using transfer learning with a neural language model outperforms the baseline (SRILM Toolkit language model) on BSLCP data. This shows that transfer learning can lead to superior language models, despite low amount of resources.

It is important to note that our pre-processed BSL corpus lacks constructs that are essential for a sign language, such as pointing signs, possessives, etc. Inclusion of these constructs into language modelling using transfer learning would increase the OOV count as the English language does not have equivalent language constructs. This raises a question whether sign languages can be modelled using written languages. Similar questions have been partly explored for the written languages in the field of machine translation [107] by bringing words of different languages close to each other in the latent space. However, nothing similar has yet been done for sign languages and is future work in this thesis. During the course of this research sign language resources were explored in detail and even though language modelling may not be appropriate at the gloss level, it can be explored as an option for classification of phonological parameters in the context of the information retrieval, where the past observations affect consecutive classifications.

From the methodological side, future work could include further experimentation to achieve greater quality of the generated models, such as attention mechanisms [249] for the recurrent neural networks or sequence-to-sequence models, which have achieved much success as generative models [238].

Despite the shortcomings of modelling sign language glosses, language models are capable of modelling the distribution of lexical items from an a priori known fixed vocabulary from a monolingual corpora. A trained model with low perplexity could become a part of an annotation tool, which could provide suggestions to the annotators given past annotated lexical items, which could in turn save the annotation efforts. Moreover, a language model with low perplexity could be used in a downstream classification task whether classifying phonological parameters, phonemes, signs, or epentheses in order to counteract any potential classification errors.

Chapter 4

Spatial and Temporal Segmentation using Pose Estimation Information

After modelling sequences of sign language glosses in Chapter 3, it became clear that the execution of signs during continuous signing is more complex than simply putting together a collection of isolated signs sequentially. Continuous signing has a syntactic structure that guides the construction of utterances, which may or may not have counterparts in English written language. Therefore, this thesis will focus on the execution of signs rather than their written interpretation. Having acknowledged the presence of sign language constructs, such as co-articulation observed in the annotation, this chapter shows how classification of phonological parameters, such as location and orientation phonological parameters are achieved using *pose key-point information* extracted directly from videos. The OpenPose library is used as it provides accuracy similar to that of commercial depth sensors and was described in Section 2.3.1.

Using this pose key-point information, the work described here performs classification of orientation and location phonological parameters following the HamNoSys notation system as discussed in Section 2.2.1. Secondly, temporal segmentation of continuous signing is explored using observation that the hand movement speed varies between signs and epentheses. This observation is then used in segmenting at both the sign and phoneme level using varying thresholds.

Finally, the drawbacks of the presented methods are discussed. In summary, the aim of this chapter is to answer the second research question (RQ2) from Section 1.3, which asks:

“ Can movement speed tracking be used in identifying sign boundaries in continuous signing? ”

4.1 Classifying Orientation and Location Phonological Parameters from Pose Estimation Information

This section is going to take a look at how pose estimation information can be used to classify phonological parameters, as defined by the HamNoSys. Looking at phonological parameters allows the analysis of sign language in its raw form and thus not relying on subjective annotation, which could lack subtle signing information. Moreover, distilling phonological parameters from the raw video frames reduces the size and complexity of the data and removes potential confusion when looking at some video frames affected by the motion blur.

4.1.1 Orientation Classification

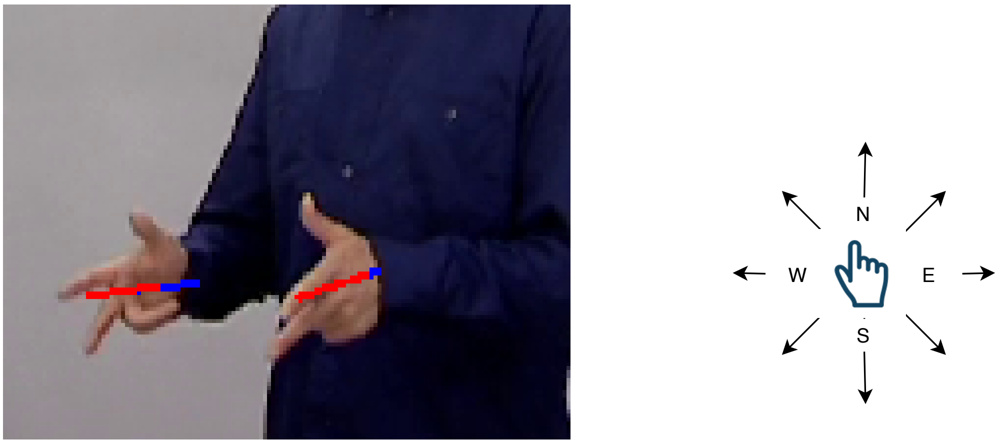
According to the HamNoSys notation system, orientation has two sub-types, namely, extended finger orientation and palm orientation, as mentioned in Table 2.2. In this section, only extended finger orientation will be considered with the following values:

North	North-East
East	South-East
South	South-West
West	North-West

Thus a total of eight orientations have been used for the extended finger orientation, as defined in the HamNoSys notation with each orientation having 45 degrees freedom. Despite the HamNoSys defining more orientations (e.g. towards or away from the body), having 2D data makes it difficult to estimate additional orientations. The angle has been calculated using the inverse trigonometric function between the radius and middle finger coordinates as can be seen in Figure 4.1 and defined by the following formula:

$$-\pi/2 < \arctan(q_y - p_y, q_x - p_x) < \pi/2$$

where q and p are the (x, y) coordinates of radius and middle finger metacarpal bones with every orientation having $\pi/4$ freedom.



Orientation Pre-Processing

Orientation Categories

Figure 4.1: An example demonstrating how orientation is calculated by considering the angle between the radius and the middle finger metacarpal bones. The red line in the image connects identified points from the OpenPose library for calculation of the extended finger orientation. Orientation categories show eight pre-defined orientations, some of which are also defined by the HamNoSys notation.

4.1.2 Location Classification

The HamNoSys notation system defines three different location sub-types, namely, hand locations, hand location sides, and hand distances as mentioned in Table 2.2. This section focuses on hand locations relative to six selected body parts as opposed to forty-six, defined by the HamNoSys notation system to simplify the detection and to comply with the OpenPose library standards. These selected body parts are:

Ears Eyes
Nose Neck
Shoulder Abdominal

In order to assign the relative hand location, a threshold has to be assigned as to how far the centroid of a hand can be from a specific body location to still be relatively close to that body part. All the distances are measured in pixels and the threshold is set to be 10% of the diagonal of the image frame, which is approximately 100 pixels. In the future work, the threshold would have to be modified based on the resolution of the camera used for data capture (if the resolution is known), but for now this threshold is fixed.

$$M_r \ldots N_r = |q_{m \ldots n} - centroid_{right}|$$

$$M_l \ldots N_l = |q_{m \ldots n} - centroid_{left}|$$

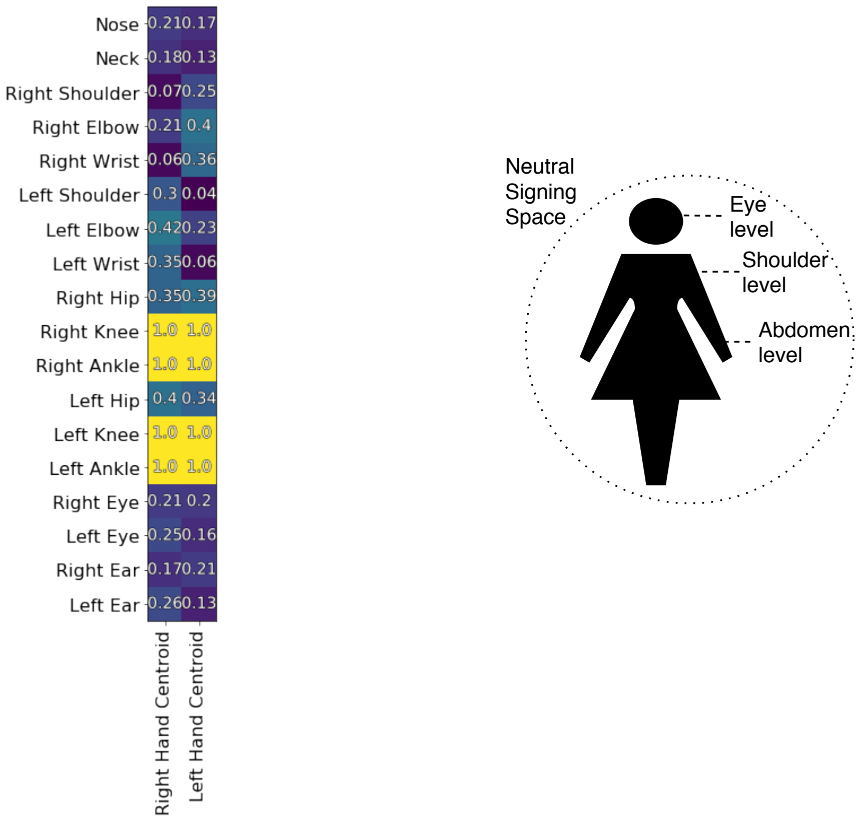
$$D = \begin{pmatrix} M_r & M_l \\ \vdots & \vdots \\ N_r & N_l \end{pmatrix}$$

Body parts ($q_{m \ldots n}$) are defined by their (x, y) positions, provided by the OpenPose library (e.g. nose, neck, shoulder, elbow, etc.) and the $M_r \ldots N_r$ and $M_l \ldots N_l$ are the Euclidean distances between the body parts and right and left hand centroids. In order to find the body part B_{right} or B_{left}, which is closest to the centroid of the right or left hand, body part index is chosen that has the smallest distance in the D matrix.

$$B_{right} = \operatorname{argmax} D_{i,1}$$

$$B_{left} = \operatorname{argmax} D_{i,2}$$

The distances are then compared to a threshold to determine if a hand is near a particular body part or is in the ‘neutral signing space’ anywhere around the body. The darker the colour in the heatmap, the closer the hand is to that body part.



Location Distances

Location Levels

Figure 4.2: An example demonstrating how location and distances are calculated by considering the distances between the centroid of each hand and each body part. The heatmap shows the distances in pixels normalised by the diagonal of a frame. The heatmap ranges from yellow to dark blue colour symbolising long and short distance respectively. Location level show three pre-defined location levels around the body, some of which are also defined by the HamNoSys notation.

Figure 4.2 shows a heatmap for both hand locations relative to all the body parts, normalised by dividing the distances by the diagonal of the frame. Yellow parts, where the distance is 1.0 means that these body parts are not visible in the frame (infinite distance).

4.2 Determining Sign Boundaries using Pose Estimation Information

Temporal segmentation in sign language does not have a specific definition as it can be accomplished at different levels (i.e. sign level [113], phoneme level [206], or sentence level [27]) and the preference depends on the project aims. That is why the epentheses annotation is rare in the past datasets (see Table 2.1). This section concerns itself with segmenting at the sign level, but the same technique with different parameters will be used for phoneme segmentation in Chapter 5. According to [50, 265], the speed of hand transition between signs is usually different from the signs themselves. Therefore, the change of the hand movement speed can correspond to the boundary between phonemes and signs.

4.2.1 Data Pre-processing

A portion of the NGT corpus was used to test the temporal segmentation method at the sign level. Annotation at the phoneme level is uncommon due to the fact that it would require more effort as many signs might contain more than one phoneme. The NGT corpus contains approximately 100 participants telling stories, or having discussions. Part of the corpus with participants retelling the Canary Row cartoon of Tweety & Sylvester by the Warner Brothers Pictures was chosen. Details about the recording setup for the corpus can be found in [62].

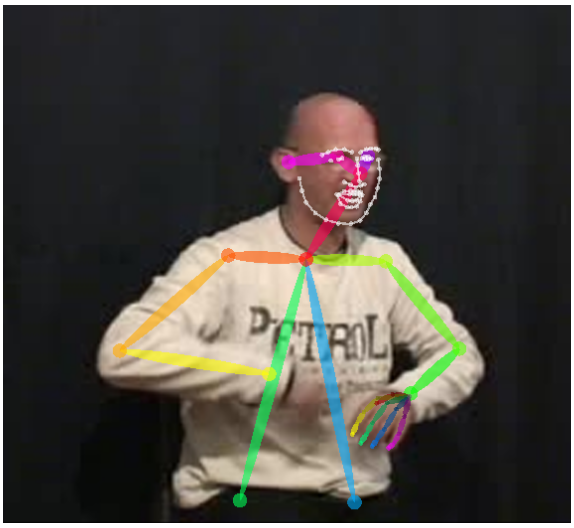


Figure 4.3: An example screenshot from the NGT corpus overlaid with the key-point information inferred by the OpenPose library. The library identifies pose key-points such as neck, shoulder, wrist, etc.

Single continuous signing footage by one participant was used for evaluation of the segmentation method. Raw video was processed with the OpenPose library in order to extract pose key-points as shown in Figure 4.3. A single video was chosen for evaluation due to the fact that normalisation of the kinematic key-points needs to be normalised for each participant because of the variations in the distance from the camera, sitting angle, and height of the participant. However, the segmentation approach is scalable assuming all participants have same distances to the camera. Chapter 5 will be performing approximate normalisation of extracted features for clustering purposes.

Like most of the datasets available, the NGT corpus does not have annotation at the epenthesis level. Therefore, gloss annotations were used to estimate which frame corresponds to beginning of a sign and which frame corresponds to the end of the sign. The ground truth was extracted from the gloss annotation of the video clips. The time between every gloss in the annotation file was considered to be an epenthesis instance with overall of 206 motion epentheses occurrences collected. It is worth noting that sign boundaries acquired in such fashion are not precise and largely dependent on the precision of the annotator.

4.2.2 Methodology

Motion epentheses are identified by looking at the distance travelled by each hand during an interval. In this particular experiment, 5 frames are chosen for this interval for detection of the motion epenthesis as was reported in [50].

$$centroid_{right} = (\sum_{i=1}^{N} x_{i_{right}}/N, \sum y_{i_{right}}/N)$$

$$centroid_{left} = (\sum_{i=1}^{N} x_{i_{left}}/N, \sum y_{i_{left}}/N)$$

First, using extracted key-point information, as can be seen in Figure 4.3, a centroid for each hand is calculated ($centroid_{handedness}$) by averaging over all the points N provided by the OpenPose library and plotted for the period of 5 frames (T1-T5 on Figure 4.4).

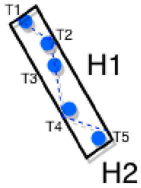


Figure 4.4: An example of hand trajectory during signing that is used to decide whether the motion is epenthesis or a part of a sign. T1-T5 correspond to centroids of hand contour, acquired during feature extraction; H1 and H2 are height and width of the minimum bounding box for the T1-T5 trajectory.

Second, a minimum bounding box is calculated for the hand trajectory over 5 frames (black rectangle on Figure 4.4). At the end, the longest side of the minimum bounding box (either H1 or H2 from the Figure 4.4) are taken to decide whether the segment is motion epenthesis or a part of a sign. Both H1 and H2 are considered, because the hand may travel in any direction during signing. Using similar techniques as in [50], the segment is labelled as epenthesis if the longest side of the minimum bounding box is between 18 and 60 pixels as advised by [50]. This epenthesis detection method provides frame number for the beginning and the the end of an epenthesis.

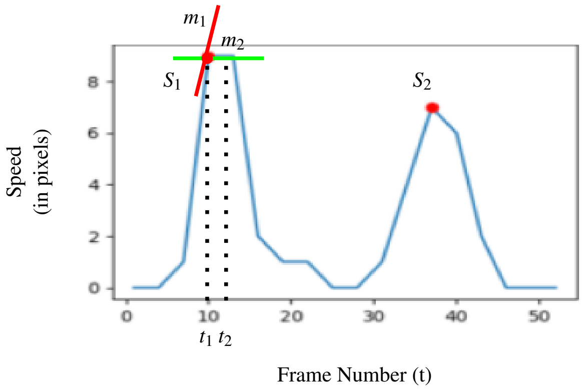
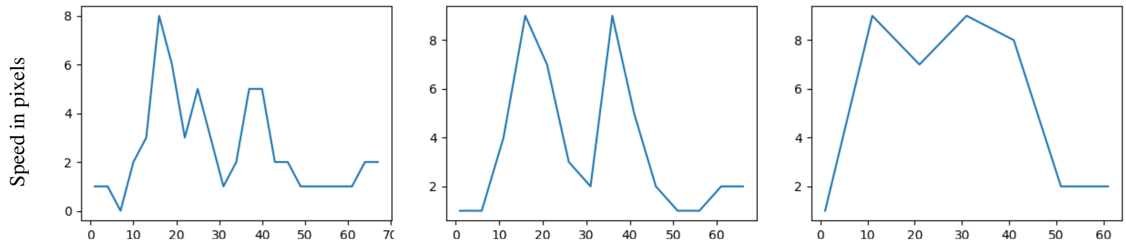


Figure 4.5: Right hand centroid ($centroid_{right}$) speed (in pixels) for an arbitrary video. Red dots indicate the potential start and the end of an actual sign. Frames $START =< t =< S_1$ and $S_2 <= t <= END$ correspond to movement from a resting position and movement into a resting position respectively, while $S_1 <= t <= S_2$ correspond to an actual sign. m_1 and m_2 indicate the slopes of the curve at consecutive frames t_1 and t_2

Using observations that signing and epentheses vary in hand movement speed, Figure 4.5 shows how the sign segmentation is achieved. In order to perform segmentation using hand movement speed graph (S) to separate epentheses and signs, extrema in the graphs are found where there is a change in slopes (m_1 and m_2) at consecutive times (t_1 and t_2). Such behaviour corresponds to changes in hand movement speed and may indicate the boundary between a sign and an epenthesis.



(a) 3 frames

(b) 5 frames

(c) 10 frames

Figure 4.6: An example of the right hand movement speed graphs in a video of a single sign of about 70 frames with different sliding window setting starting from 3 to 10 frames per window. As a result, the more frames are considered for the sliding window, the smoother the speed graph becomes, resulting in fewer points for temporal segmentation.

Increasing the number of frames in the sliding window results in smoothing of the speed graph (S) as can be seen in Figure 4.6, where the speed graph is plotted for the same short video of approximately 100 frames. Increasing the size of the sliding window from 3 frames (Figure 4.6 a) to 10 frames (Figure 4.6 c) results in a smoother graph with fewer speed changes. Smoothing of the speed graph makes it more difficult to spot the changes in the hand movement speed as there are less volatility in graphs and, therefore, less segmentation points.

4.2.3 Results

To calculate the accuracy in terms of F-measure, the returned epenthesis interval is compared to the ground truth, extracted from the annotated video. As a result, the algorithm identified 201 True Positives TP that were found within the ground truth (Predicted $\in$ GT). Some of the identified intervals are repeated, due to the fact that both hands are tracked and analysed for the epenthesis identification. The algorithm identified 39 False Positives FP that did not match epentheses in the ground truth (Predicted $\notin$ GT). All the intervals that were not included in the predicted TP are assumed to be True Negatives TN (Predicted $\in \neg$ GT). The algorithm identified 210 TN intervals. The intervals that were considered and were not in the ground truth were assumed to be False Negatives FN (Predicted $\notin \neg$ GT). The algorithm identified 46 FN intervals.

It is worth noting that the frames identified by the method do not precisely match the annotated ground truth, but can overlap with the ground truth to be considered as true positives.

$$F - measure \quad = (2 * Precision * Recall)/(Precision + Recall) = 0.825, where$$

$$Precision = TP/(TP + FP) = 0.837 \;\; and$$

$$Recall = TP/(TP + FN) = 0.813$$

4.3 Discussion

This chapter focused on spatial segmentation of phonological parameters in raw video frames and the use of identified skeletal key-points in performing temporal segmentation.

When the pose can be estimated successfully, it has the potential to reduce the complexity of sign language resources and make movement information salient for modelling or analysis purposes. Moreover, reducing the data even further by applying HamNoSys classification, allows one to expose crucial linguistic information about signing, eliminating uncertainty when looking at the raw signing videos, which can be affected by motion blur in cases where the camera quality is low. This technique will be used later in the thesis when performing phoneme clustering in Chapter 5, generating datasets for phonological parameters in Chapter 6, and when comparing sign languages using their phonological parameters in Chapter 7. Correct classification of location and orientation phonological parameters using trigonometric functions may not be enough as the HamNoSys notation describes additional proximity to the body information, which is much more difficult to estimate from the 2D images. OpenPose library can predict the 3D key-point data from 2D images, however, the accuracy of such predictions is lower, but it could improve in the future releases.

Table 4.1: Reported accuracy scores when performing temporal segmentation compared to our proposed speed tracking approach.

Data set	Model	Accuracy (in %)
ECHO [64]	Speed and trajectory discontinuity [110]	79
Sign3D [103]	Hand shape transitions [188]	80
NGT [63]	Our linguistic heuristics method	80

Sign segmentation techniques reported in this chapter, indicate that hand movement speed information can be used to detect epenthesis and is comparable to similar approaches applied to different datasets that have videos of more than one signer, as can be seen in Table 4.1. However, it should be apparent that there is a high degree of variation in signing speeds among different signers as well as the speed of signing for every signer depending on the situation, conversation topic, etc. Therefore, a single threshold may be not sufficient in capturing all possible variations.

Varying parameters of the segmentation may yield different results by manipulating the threshold of the H1 and/or H2 and simultaneously changing the number of frames for which H1 and H2 are computed. By allowing more frames, it would be more likely for the H1 or H2 to increase. Therefore, it is important to use the information about the average sign length and choose the number of frames to be fewer than the average number of frames per sign.

As more and more available datasets have continuous signing, segmentation becomes essential for sign spotting or sentence boundary detection purposes. This becomes even more important when having continuous streams of data from, for example, live events. In such cases when the data is being transmitted over the network, reducing the size of such data by extracting HamNoSys phonological parameters and then performing temporal segmentation would be required for the purposes of translation or sign spotting.

Chapter 5

Phoneme Clustering

Chapter 4 showed that continuous signing can be split into individual signs by tracking the speed of hand movement. Experimentation demonstrated that certain static thresholds were sufficient at segmenting continuous signing into individual signs. By adjusting the thresholds (as can be seen in Figure 4.6) it could be possible to segment at a phoneme level by detecting small variations in hand movement. At the same time, Section 4.1 showed how location and orientation phonological parameters can be calculated using pose estimation information.

This chapter explores clustering as it is an important part of data mining when details about the data are not known in advance and the data needs to be analysed for modelling purposes or be prepared for rapid annotation. In this chapter two methods for clustering phonemes are explored. We describe phonemes by their corresponding location and orientation phonological parameters over a number of frames, which are determined by the segmentation method from Chapter 4. The handshape phonological parameter has been left out for this study due to the fact that the location phonological parameter is the most immutable phonological parameter despite the occurrence of co-articulations [196], whereas the handshape can be greatly affected by the co-articulation phenomena, described in Section 1.2. Nayak et al. [189] have done work on clustering of invariant parts of each sign. However, they used hand colour and edge information for clustering and not phonological parameters as in the approach described in this chapter.

First, four videos of the same interpreted song are collected from YouTube and pre-processed to extract phonemes. Second, phonemes for each video are clustered using a similarity metric. Third, similar sequences of consecutive phonemes are identified using the same phoneme similarity metric, which shows that it is possible to identify where similar phrases are signed in a single video. Finally, a discussion is given into how clustering can be used to generate datasets for modelling purposes. By doing so, this chapter aims to answer the third research question (RQ3) from Section 1.3, which asks:

> “ Can similar signs or phrases be found in different continuous signing videos without having the transcription? ”

5.1 Data Pre-processing

Four videos of different signers interpreting the song ‘Halo’ by Beyoncé in American Sign Language were collected from YouTube with average time length of 4 minutes 22 seconds. The audio of the song plays in the background, the lyrics of which were used as ground truth when referring to the translation of what is being signed in the videos.

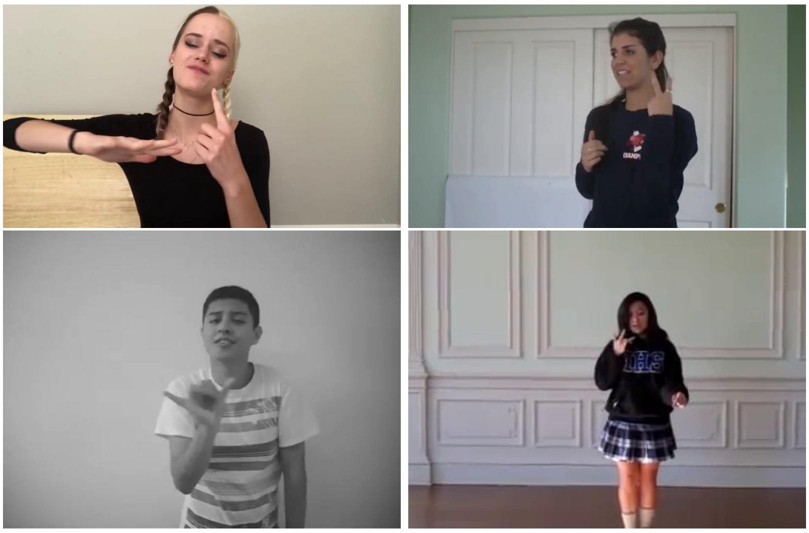


Figure 5.1: Screenshots from the used song interpretation videos from YouTube

Figure 5.1 shows screenshots from the four collected videos. Due to the fact that the data is collected from online resources, it is more unpredictable than the corpora as described in [219, 245, 63] that was collected in controlled environments. After a short qualitative analysis of the collected videos, the following factors were noted to vary across videos:

- Video aspect ratio
- Video quality
- Camera proximity
- Background
- Body orientations
- Dialects
- Signing fluency

This work takes advantage of the nature of the data since music usually has verses, which repeat themselves over a course of a song. This is useful when performing clustering since it can be assumed that every verse could be a cluster.

Listed in Section 3, normalisation similar to Fragkiadakis et al. [96] is applied to extracted key-points to force fixed shoulder distance in all videos by resizing all (x,y) positions of the inferred key-points $(\hat{x},\hat{y}) \rightarrow (x*r_x, y*r_y)$ and bring the signer to the centre of the screen by adding the difference between desired and current positions $(x,y) \rightarrow (\hat{x}+t_x, \hat{y}+t_y)$, where r and t are the resize ratio and translation pixels. Such normalisation counteracts the fact that the signers appear at different distances away from the camera.

The key-point information is then used for the phoneme segmentation purposes using method described in Section 4.2 with the sliding window size of 3 frames since it provides more segmentation points as can be seen in Figure 4.6. Information about the start and the end frame numbers of each phoneme for the right and left hands individually becomes available after the segmentation. Finally, using key-points from the first stage, extended finger orientation and the hand location relative to the body are estimated for both right and left hands for every frame as described in Section 4.1.

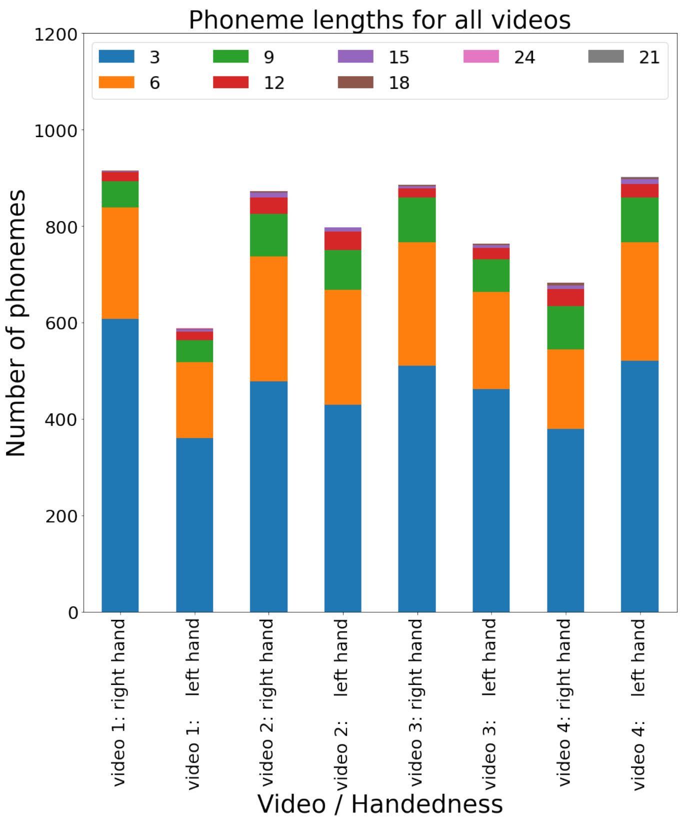


Figure 5.2: Total number of phonemes for each video after the segmentation step and relative phoneme sizes in the number of frames.

By the end of the pre-processing of continuous raw videos, two lists one for right hand and another for the left hand, are generated for each video. Both lists for a single video with each phoneme having a start and end frame information as well as the location and orientation of a hand for every frame. Figure 5.2 shows that the majority of the phonemes are short (3 frames) and the longest phoneme discovered having 24 frames. From the figure it can be seen that using the same phoneme segmentation method on different videos produces slightly different results. For example, video 4 has much fewer phonemes identified for the right hand as compared to other 3 videos. Similarly, video 1 has much fewer phonemes identified for the left hand than other 3 videos. Moreover, the largest share of the overall number of phonemes belongs to the short phonemes consisting of three and six frames, while the average sign length has been reported to be of 3 seconds, or 75 frames in [162].

5.2 Methodology

Clustering can be used when the data is unknown to get insights into the data. Choice of a clustering algorithm largely depends on the problem as every approach has its assumptions [192]. Since there is no way of knowing how many similar phonemes there are in a video prior to watching the video, the clustering approach selected should not assume the number of clusters. Two clustering methods will be compared. The first clustering method is an iterative grouping clustering, which iterates a list of phonemes, compares them and puts the two phonemes into the same cluster and repeats until all the phonemes are have been iterated. The second clustering method is the general clustering method DBSCAN. This density-based clustering method searches its neighbourhood for the members of the same cluster and can detect arbitrary-shaped cluster shapes. The main differentiating factor between this density-based clustering algorithm and the traditional clustering algorithms (like K-Means) is the DBSCAN's ability to detect clusters of arbitrary shapes and avoid outliers. The ε parameter specifies radius of the neighbourhood around each point and minimum number of neighbors around that point. For the DBSCAN, the number of points in the neighbourhood was modified $[1-5]$ for a point to be considered a core point using fixed $\varepsilon = 0.5$ since the largest distance between the two phonemes is in $[0-1]$ range. The algorithm works by calculating a pairwise distance between all the points, selecting core points of each neighbourhood and either creating a new cluster or assigning core points to an existing cluster while searching for connected points and assigning them to the same cluster as the core point. The complexity of both algorithms is $\mathcal{O}(n^2)$ [221]. Similar phonemes are grouped together given a similarity threshold. Generally the similarity between the two phonemes is expressed as:

$$\exists p_n, p_m \in P, p_n \sim^T p_m \iff p_n \subsetneq^T p_m \wedge p_m \subsetneq^T p_n$$

where p is a phoneme, P are all pre-processed phonemes, and T is a threshold. This means that the two phonemes (p_n and p_m) are similar if and only if there exists a subset of phonological parameters ($p_n \subsetneq^T p_m \wedge p_m \subsetneq^T p_n$) within both phonemes that is equal and the length of this subset is more or equal to the threshold T. To realise the comparison of the phonemes, the weighted Levenshtein distance [210] is used to measure the difference between two phonemes as it was found to be appropriate in Section 2.4 considering the grammar of sign language. Weighting of distances is specified manually and reflects the linguistic decomposition of signing into phonological parameters. Currently, however, not all phonological parameters are identified. For example, the information on whether a hand is next to the body or away from the body and non-manual gestures are yet to be identified and encoded in the phoneme definition. The Levenshtein distance metric finds the minimum number of edits where an edit can be a substitution or deletion of phonological parameter and the weights determine the cost of such edits. Since the majority of clustering algorithms operate on the affinity matrices, where every item corresponds to a distance between the two items, a distance metric is needed for the orientation and location phonological parameters.

Orientation is divided in eight equal ranges (North, North-East, East, South-East, South, South-West, West, and North-West), while the location is divided into three ranges (eye, shoulder, and abdomen levels) as described in Section 4.1. Therefore, the orientation distance range is between 0 and 4 while the location distance range is between 0 and 2 as shown in Figure 4.1. Distances between the two items also become the weights for weighted Levenshtein distance matrix.

The Silhouette metric [214] is used to evaluate the two clustering approaches as it has been deemed to resemble human judgement [159]. The metric is bound between 1 and -1 values with 1 signifying a good clustering results and -1 indicating that the items in a cluster are assigned to the wrong cluster. 0 represents many overlapping clusters in the data, which makes clustering challenging.

5.3 Results

5.3.1 Phoneme Clustering Analysis

This section shows how relaxing the phoneme similarity metric affects the number of clusters and cluster size with both the DBSCAN and the proposed grouping methods.

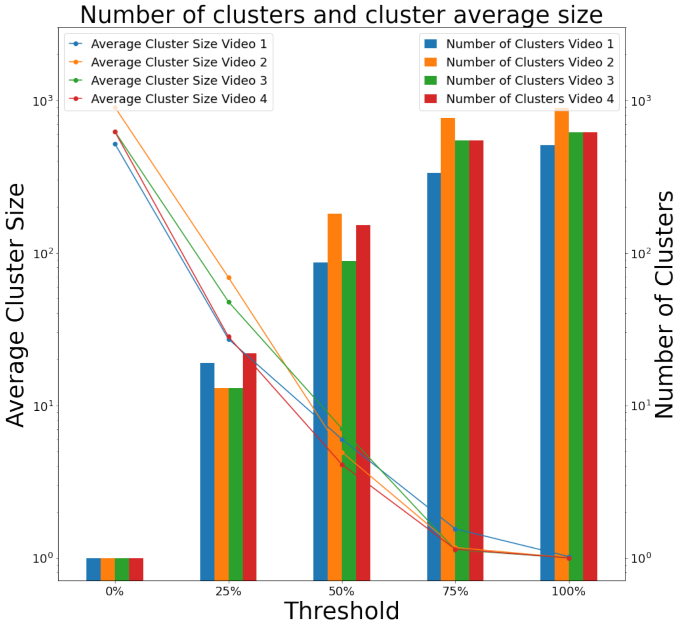


Figure 5.3: Number of clusters (bars) and average cluster size (line) using grouping clustering method with various similarity thresholds.

Figure 5.3 shows identified clusters and their average sizes with varying phoneme similarity threshold. Setting the threshold to 0%, puts all the phonemes in one cluster, whereas setting the threshold to 100% results in almost every phoneme being assigned to an individual cluster. This means that throughout each video, there are very few similar phonemes using proposed distance metric. This supports the fact that sign languages are fluid and the same sign can have inter-signer [168] as well as intra-signer [22] variations.

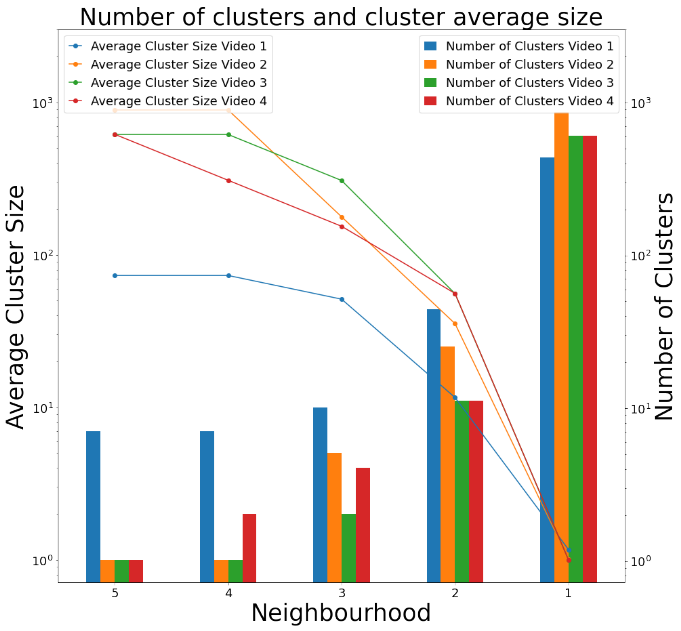


Figure 5.4: Number of clusters (bars) and average cluster size (line) using DBSCAN clustering method with various neighbourhood sizes.

Similar behaviour is observed when clustering using the DBSCAN method, as can be seen in Figure 5.4. As the neighbourhood size decreases, the number of clusters increase. However, it is important to notice the rate of change in the average cluster size as the similarity thresholds change (convex vs concave curve shape). Similarly, the rate of change in number of clusters is not as steep when clustered using the DBSCAN method (25%-75% vs 4-2 neighbours). This shows that the DBSCAN produces clusters where the number of phonemes are more uniformly distributed than with the grouping clustering method.

5.3.2 Quantitative Clustering Evaluation

Both DBSCAN and the proposed grouping clustering methods are compared using the Silhouette metric [214]. The Silhouette metric has been deemed to resemble human judgement [159]. The metric is bound in range $[1, -1]$ with 1 signifying a good clustering results and -1 indicating that the items in a cluster are assigned to the wrong cluster. 0 represents many overlapping clusters in the data, which makes clustering challenging.

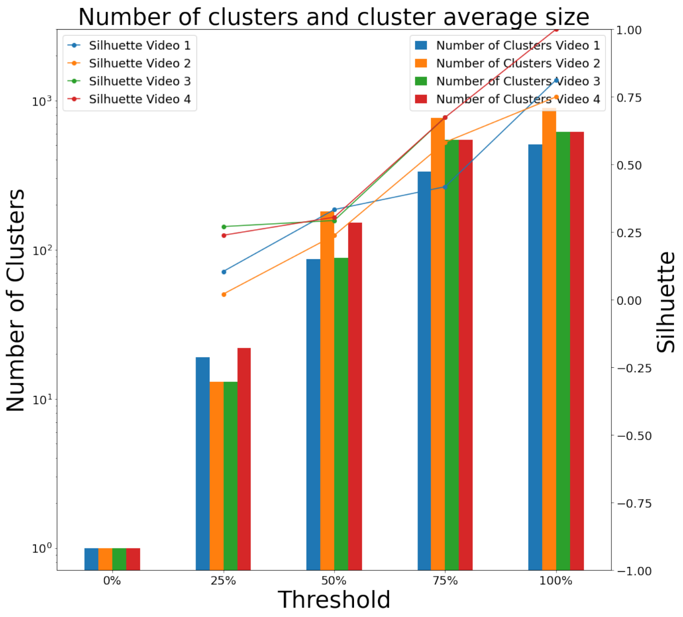


Figure 5.5: Silhouette clustering evaluation metric applied to grouping clustering algorithm at various similarity thresholds. Every phoneme is assigned to individual cluster at similarity threshold=0 results in undefined Silhouette value.

Figure 5.5 shows how the Silhouette metrics changes with respect to the phoneme similarity threshold when using the proposed grouping clustering method. As the threshold is becoming more restrictive, identified clusters become clustered better at the expense of increasing number of clusters (Figure 5.3).

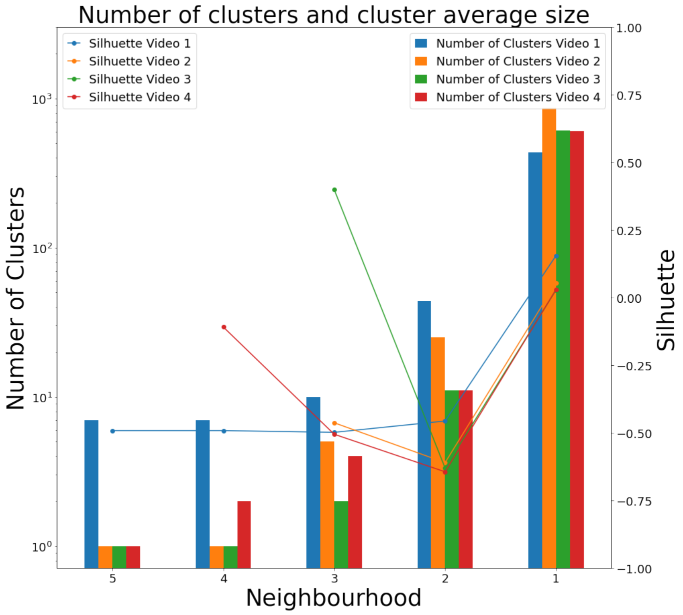


Figure 5.6: Silhouette clustering evaluation metric applied to DBSCAN clustering algorithm with different neighbourhood sizes. Every phoneme is assigned to individual cluster at similarity threshold=0 for videos 2,3,and 4 resulting in undefined Silhouette value.

Figure 5.6 shows DBSCAN's clustering performance with respect to the number of neighbours while having $\varepsilon = 0.5$. The Silhouette coefficient is more volatile to the neighbourhood size and the number of identified clusters increases as the neighbourhood becomes smaller.

What is worth noting is that for every video, the proposed grouping clustering method has relatively similar Silhouette value while DBSCAN at $neighbourhood = 2$ and $neighbourhood = 3$ has very different Silhouette values for each video, which makes it unpredictable to use in the general case. On the other hand, having more uniformly distributed cluster sizes at different thresholds and relatively good Silhouette metric makes it an attractive clustering option for sign language data.

5.3.3 Qualitative Clustering Evaluation

This section shows example phonemes from different clusters when clustered using the grouping method with the similarity threshold set to 50%.

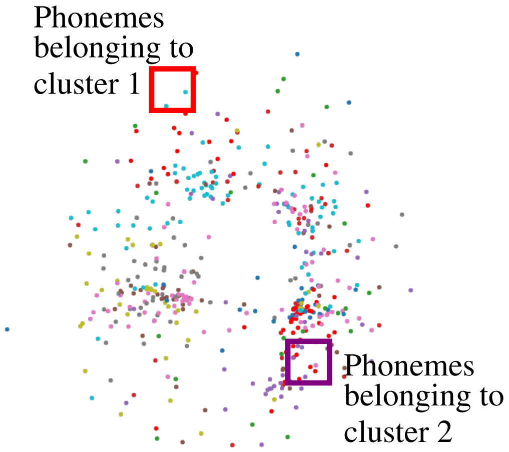


Figure 5.7: Phonemes from a single video clustered using the iterative grouping method with 50% threshold with each cluster having a distinct colour. Examples from the two bounding boxes are shown below.

Figure 5.7 shows visualisation of all the clusters for a single video when plotting two principal components on (x,y) axes. Clusters are identified using the iterative grouping method with 50% threshold. Figures below show two examples for each region of interest (red and violet).

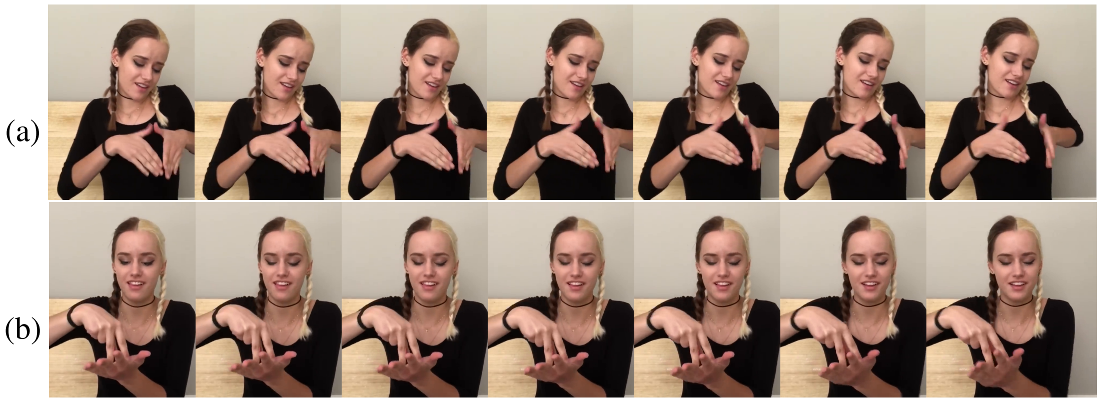


Figure 5.8: Two similar phonemes (Frames 1196-1202 for the phoneme (a) & 3556-3562 for the phoneme (b)) from cluster A (red). Phoneme (a) being a part of the 'I found a way to let you in' lyrics and phoneme (b) being a part of the 'to pull me back to the ground again' lyrics.

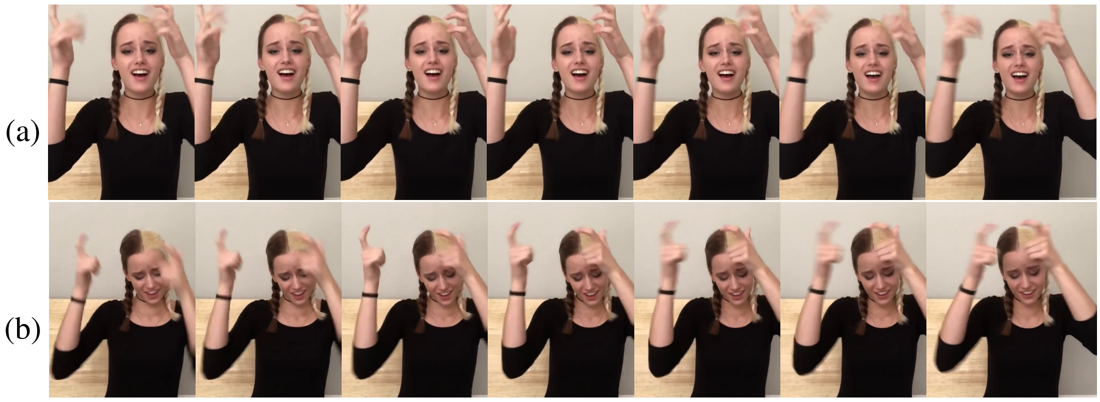


Figure 5.9: Two similar phonemes (Frames 4736-4742 for the phoneme (a) & 7126-7132 for the phoneme (b)) from cluster B (violet). Phonemes (a) and (b) being a part of the 'Halo, halo' lyrics.

From the Figures 5.8 and 5.9 it can be seen that the phonemes are visually similar where every phoneme is described with the location and orientation phonological parameters. It is worth noting that the meaning can or cannot be the same, but the phonological parameters are similar having at least 50% phoneme similarity.

5.3.4 Matching Consecutive Sequences of Phonemes

A single video is chosen to search for similar consecutive phonemes in different parts of the video. The lyrics in the background music in the video are used as the ground truth for verifying the results. Phoneme similarity threshold is chosen to be 50%.

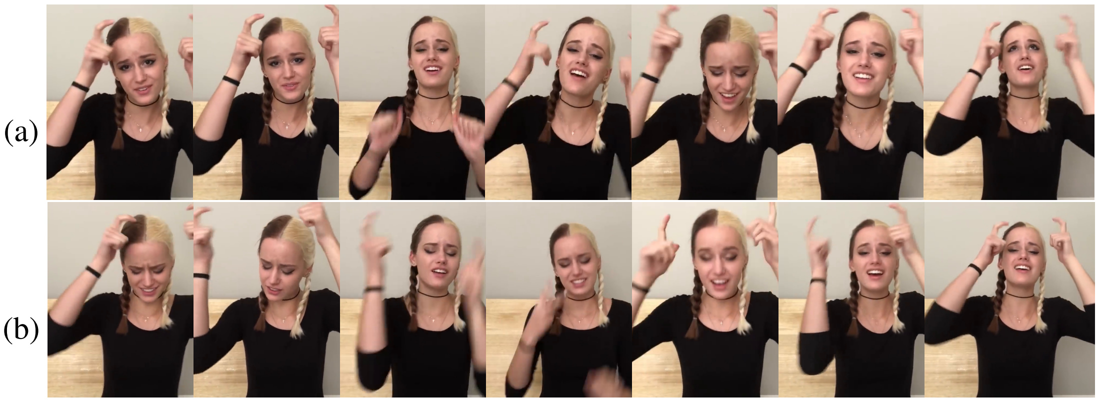


Figure 5.10: Two similar combinations of consecutive phonemes (Frames 5012-5207 for consecutive phonemes (a) & 2451-2839 for consecutive phonemes (b)) corresponding to the same lyrics:

"I can feel your Halo, Halo, Halo. I can see your Halo, Halo, Halo
I can feel your Halo, Halo, Halo. I can see your Halo, Halo, Halo"

As a result it can be seen that the same phoneme similarity approach described in Section 4.4 can be used to find similar consecutive phonemes in the same video as shown in 5.10 (a) and (b). Note that every sequence is between 200 and 400 phonemes with an average sign length having 75 frames [162].

5.4 Discussion

This chapter has shown that with the help of the location and orientation phonological parameters it is possible to design a distance metric to compare and cluster phonemes by combining similar phonemes without the need for an expensive transcription. This means that it is possible to search for phonemes, signs, and even phrases within videos. An iterative grouping method and the DBSCAN methods were compared, showing that the DBSCAN is more unpredictable for the same signed content when the phoneme similarity conditions (neighbourhood) are relaxed. Nevertheless, DBSCAN has a more even distribution of identified clusters and the average cluster size when similarity conditions are relaxed, as compared to the iterative grouping method. This work has shown that it is possible to use a linguistically viable distance metric for general purpose clustering algorithms to work on sign language data. To improve the clustering approach, it would be beneficial to include other phonological parameters, such as handshape and even non-manual features and compare the results clustering using different combinations of phonological parameters to verify hypothesis that the location phonological parameter contributes the most to the recognition of a sign [161]. Despite the kinematic key-point data normalisations applied to the extracted features, orientation is much more difficult to normalise for, as it would require additional thresholds (e.g. body width) and iterative optimisations (e.g. step-wise interpolation of all the kinematic key-points in either direction).

On one hand, the benefit of using clustering on music videos is that it is possible to identify repeating patterns in a video, which correspond to verses in music. This knowledge can be used to generate datasets for sign language translation models, as more and more song interpretations in sign languages performed by different signers become available online. On the other hand, there is little to no information about signers (i.e. signing fluency, background), which is often a significant factor when selecting signers for linguistic studies and data collection. This raises a question whether signing videos found online can be exploited in conducting sign language research and what measures should be put in place to ensure the quality of the data.

Using the distance metric for comparing phonemes, signs, and even phrases is scalable to other datasets. ASL interpretations of music videos were chosen due to the fact that it was already known that there will be repetitions in signing (verses in music), which were identified using the clustering approach. Other types of signing videos (e.g. storytelling or spontaneous conversations) are less likely to have phrase repetitions. However, similar phonemes or signs can be identified in these videos. Therefore, showing that sign language data could be compared using a distance metric, paves the way towards sign language data processing from the real-time streaming services for such purposes as detection of certain signed signs or even phrases as well as dataset creation through phoneme/sign/phrase clustering.

Chapter 6

Automatic Classification of Phonological Parameters using End-To-End Models

6.1 Introduction

Section 4.1 showed how key-point information from pose estimation can be turned into location and orientation HamNoSys phonological parameters by applying trigonometry. Despite the fact that the precision of the OpenPose library resembles that of the commercial sensors, the library does not provide information about handshape phonological parameters and this cannot be easily calculated using the pose and finger configuration information. Nevertheless, estimating the shape of a hand from the pose key-points has been explored in human-robot interaction studies [47, 212], especially in the presence of occlusion while manipulating an object. This chapter aims to train end-to-end HamNoSys location, orientation, and handshape phonological parameter classifier by incorporating co-dependency among location and orientation into a single model with the aim of reducing the search space and improving recognition accuracy. With regards to the handshape classification, multiple methods are compared including transfer learning techniques.

Finally, this chapter shows how such an end-to-end model could be extended to incorporate the remaining phonological parameters to cover motion phonological parameter. This chapter thus answers the fourth research question (RQ4) posed in Section 1.3, which asks:

" Can automatic classification of phonological parameters (e.g. location and orientation) be improved by exploiting their co-dependencies? "

6.2 Effectiveness of Various Features in Training Supervised Models for Handshape Recognition

Handshape phonological parameters will be explored in this section including which features lead to a superior model when classifying hand shapes. These features would include the raw OpenPose kinematic key-points, distances between the raw key-points, binary skeleton image of a hand, and raw cropped image of the hand, showing that the classification of the raw cropped images of hands using pre-trained model achieves the highest accuracy.

6.2.1 Data Pre-processing

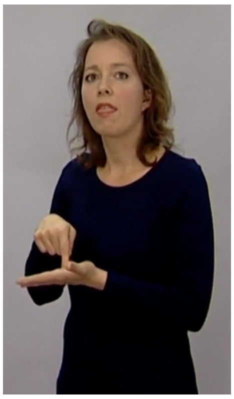


Figure 6.1: An example screenshot of a signer signing 'STED' ('PLACE' in English) from the OODT dataset

Ordbog over Dansk Tegnsprog (OODT) data set was chosen for modelling phonological parameters. It is a digital dictionary with a web interface that allows searching for a specific sign using phonological parameters and a gloss in Danish written language. Figure 6.1 shows an example frame from one isolated sign from the dataset.

```
<Entry>
  <EntryNo>7</EntryNo>
  <Gloss>TAPPE-VIDEO</Gloss>
  <SignVideo>t_2542.mp4</SignVideo>
  <Phonology>
    <Seq>
      <SeqNo>1 </SeqNo>
      <SignType>2-hand parallel </SignType>
      <Handshape1>paedagog-hand aben </Handshape1>
      <HandshapeFinal>paedagog-hand </HandshapeFinal>
      <OrientationFingers>skrat frem op</OrientationFingers>
      <OrientationPalm>op</OrientationPalm>
      <Location>neutralt rum</Location>
      <Movement>ned </Movement>
      <Relation>ved siden af</Relation>
      <Repeat/>
    </Seq>
  </Phonology>
</Entry>
```

Listing 6.1: One Sample from the OODT data set showing phonological transcription of the sign ‘OPTAGE’ (‘ABSORB’ in English), containing one set ($SeqNo = 1$) of phonological parameters

Listing 6.1 shows a single entry from the annotation file that contains all the information about the data set. Each entry contains such information as the gloss, path to a video clip, handshape, orientation, location, and movement. It is important to note that the annotation does not have the exact timing information about when a certain phoneme is being used in a clip. Section 1.2 discusses that signs may contain more than one phoneme per each hand and there could be more than one phonological parameter for each phoneme. For the purposes of this study, signs that have one phoneme were collected due to the inability assigning individual phonemes to specific signs as it is challenging to say which phoneme belongs to which sign.

The OODT data set contains isolated videos of people signing one sign at a time without any additional context. Therefore, every video begins with signers being in a resting position (having their hands down at the abdominal level or outside the frame) and end with the same position. Since the annotation does not provide any information about the exact timing when a certain phoneme is taking place in the video, it is required to filter the data set to exclude the frames that are recorded while the signers are in the resting position.

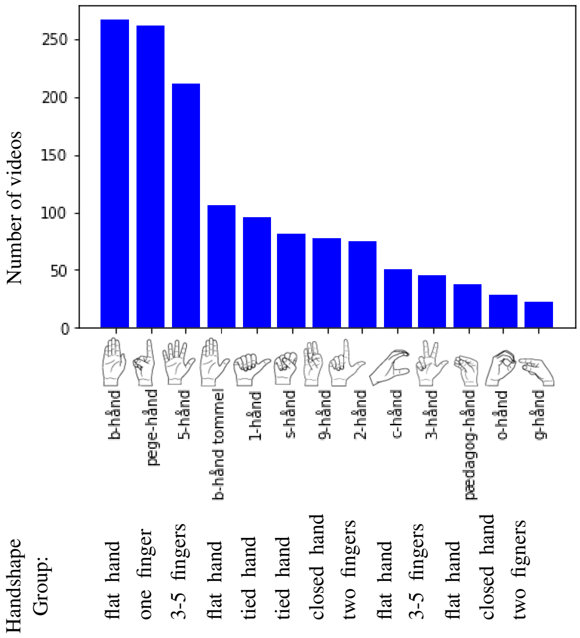


Figure 6.2: Number of selected videos from the OODT data set with one sign per video and one phoneme per sign where the handshape phonological parameter does not change throughout the phoneme

Figure 6.2 shows the handshapes, which were selected based on the number of videos in the overall data set as well as the coverage of all the hand shape groups in the OODT, which are the tied hand (s-hand, 1-hand), flat hand (b-hand, b-hand tommel, c-hand, paedagog-hand), 1-finger (pege-hand), 2-fingers (2-hand, g-hand), 3-5 fingers (3-hand, 5-hand), and closed-hand (9-hand, o-hand).

Pre-processing of the OODT data is done to extract frames from the raw videos, identify pose and hand information in each frame and to eliminate the frames that correspond to epentheses or resting position. For this purpose, the same segmentation algorithm was used as in Section 4.2 and later in Chapter 5 for determining phoneme boundaries.

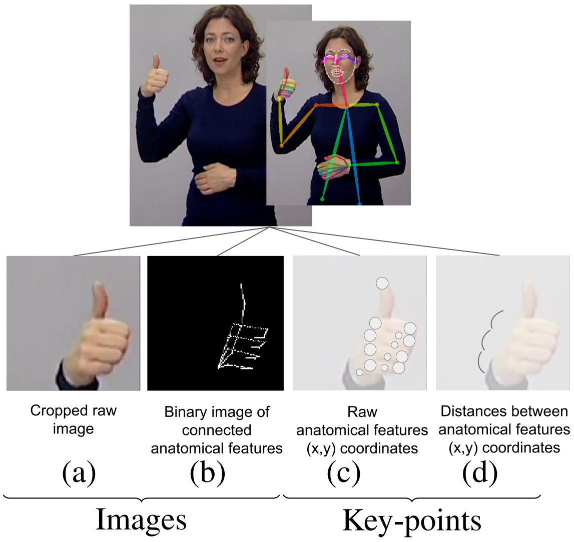

Cropped raw image
Binary image of connected anatomical features
Raw anatomical features (x,y) coordinates
Distances between anatomical features (x,y) coordinates
(a)
(b)
(c)
(d)
Images
Key-points

Figure 6.3: Generated data sets from the original OODT data grouped into images and key-points groups. Data sets from the images group corresponding to images while the key-points group contains either raw key-points, provided by the OpenPose library or the distances between the raw key-points

Figure 6.3 shows the generated data sets from the original OODT data. On the top is a single frame from the original data and to its right is the output from the OpenPose library. Using this information, four data sets were generated to test which features lead to a more accurate model for handshape recognition.

The first data set consists of cropped (128 × 128 pixels) raw images of each hand as seen in Figure 6.3 (a).

The second data set consists of binary images of connected anatomic features of each hand as seen in Figure 6.3 (b). The anatomic features, provided by the OpenPose library, are connected using linear regression [186] as shown below:

$$p(x) = a_0 + a_1 x$$

$$S_r = \sum_{i=0}^{n} |p(x_j) - y_j|^2$$

where the aim is to find a_0 (where the line intersects the axis) and a_1 (the slope of the line), while minimising the sum of the square of the residuals S_r from the two data points. Later, the points are generated to fill the $p(x)$ line for k steps, which is chosen to be 10.

Type	**Hand Part** (x,y)				
phalanx	thumb	index	middle	ring	little
proximal	thumb	index	middle	ring	little
metacarpals	thumb	index	middle	ring	little
carpals		index	middle	ring	little
other	radius	trapezium			

Table 6.1: The third data set consists of the 21 (x,y) coordinates of the anatomic features provided by the OpenPose library

The third data set consists of raw (x,y) coordinates of the anatomic features as seen in Figure 6.3 (c) and as described in Table 6.1.

From (x_1, y_1)	**To** (x_2, y_2)				
radius	thumb	index	middle	ring	little
thumb		index	middle	ring	little
index			middle	ring	little
middle				ring	little
ring					little

Table 6.2: The fourth data set consists of 15 distances between (x_1,y_1) and (x_2,y_2) coordinates of the anatomic features provided by the OpenPose library

The fourth data set consists of the 15 distances (in pixels) between the raw coordinates of the anatomic features (x_1,y_1) and (x_2,y_2) as seen in Figure 6.3 (d) and as described in Table 6.2.

To discard the frames where the signers have their hands in the resting position, the hypothesis that the hand movement speed differs between the phonemes and the epentheses (hand movements between signs) was used [50, 265]. Sliding window over a set of 3 frames was used and hand movement speed was calculated using approach described in Section 4.2.

After discarding frames that correspond to epentheses, all the data for every data set was split 67% / 16.5% / 16.5% for the training/validation/testing respectively. Also, the testing data is verified manually and 78% of the samples made it into the final test set.

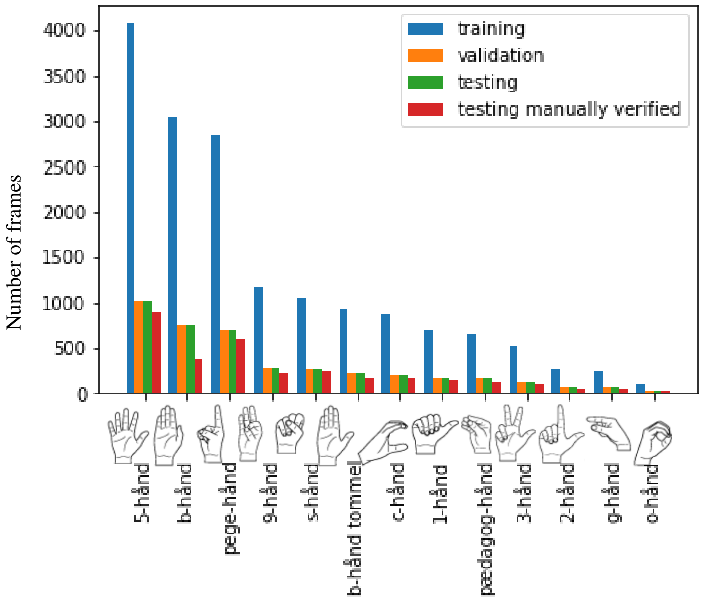


Figure 6.4: Every generated data set split for model training, validation, and testing. The testing data set was manually verified, resulting in refined testing data set

Since the author of this work does not have linguistic background in the sign languages and, in particular, in DTS, every frame was judged subjectively, but conservatively. In such cases when hands in the frames appeared blurry due to the motion blur or where not all the fingers were visible due to occlusions, these frames were discarded.

6.2.2 Methodology

The key-points datasets consisting of raw key-points (referred to as ‘Raw’) and distance (referred to as ‘Distance’) were used to train the nearest neighbour [59], random forest [32], and feed-forward neural network [21]. The image datasets consisting of raw images (referred to as ‘Raw Image’) and synthesised binary images (referred to as ‘Binary Image’) were used to train convolutional neural network [155] to recognise different hand shapes.

K-Fold cross-validation with five folds for the Nearest Neighbour, Random Forest, and Feed-Forward Neural Network methods was used to tune the parameters and report the average prediction accuracies over all the folds. In the case, where a parameter has alternatives, separated by comma, the parameter in bold denotes the selected parameters for the best model. The best model is determined by the low overfitting (small difference between training and testing) and high accuracy.

6.2.2.1 Nearest Neighbour (k-NN)

Nearest Neighbour is a clustering algorithm, where every new unseen data point is subjected to the *k*-nearest neighbours vote using some distance metric similar to the 'Neighbourhood' parameter of the DBSCAN clustering algorithm used in Chapter 5.

$$\text{Neighbours} \quad 1, \mathbf{5}, 10, 15, 20$$

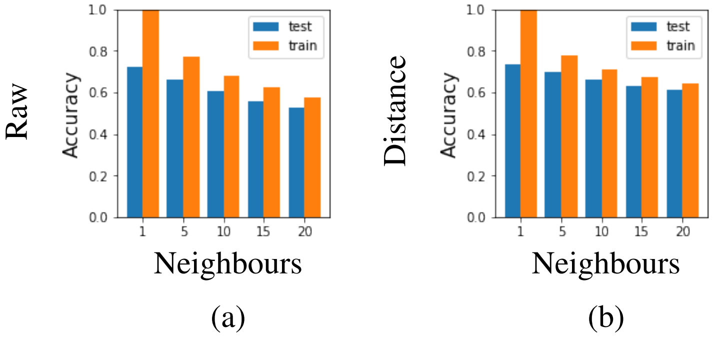


Figure 6.5: k-NN with different k is applied to both (a) Raw and (b) Distance extracted hand features as described in Figure 6.3. $Neigbrours = 5$ parameter (in bold) is the most optimal as it shows high accuracy and does not overfit excessively for both Raw and Distance datasets, which is evident by looking at the difference between train and test accuracies.

The number of neighbours affects the classification accuracy for both a) raw features and the b) distance features, as is shown in Figure 6.5. With only one neighbour, it can be seen that the model overfits as there is a big difference between the training (raw 99% and distance 99%) and the testing (raw 72% and distance 73%) accuracies. Therefore, $Neighbours = 5$ parameter is chosen as the most optimal one as it has high accuracy and does not overfit as must as $Neighbours = 1$ setting.

6.2.2.2 Random Forest (RF)

A Random Forest classifier trains one or more decision trees on sub-samples of the overall data using bagging and bootstrapping and uses aggregation to improve the predictive accuracy and control over-fitting. Decision tree pruning is also used to control over-fitting. Decision tree is a tree-based data structure, where every node learns a decision rule that partitions the overall decision space.

Maximum Leaf Nodes	100, 300, 500, **800**
Number of Estimators	5, 10, 15, 20, 25, **30**
Maximum Tree Depth	5, 10, 15, **20**
Minimum Samples per Leaf	10, **50**, 100
Minimum Samples per Split	10, **50**, 100
Maximum Features	0.1

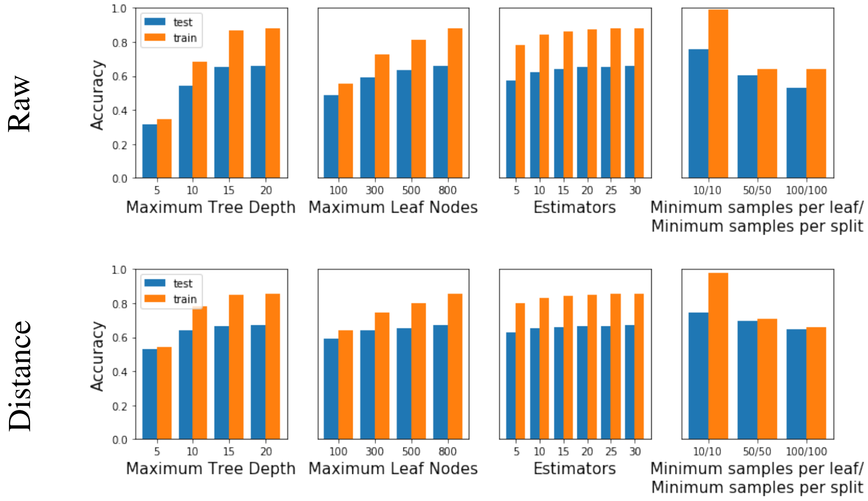


Figure 6.6: Random forest with different parameters is applied to both Raw and Distance hand features as described in Figure 6.3. 800 maximum leaf nodes, 30 estimators, depth of 20 levels per tree, 50 minimum samples per leaf, and 50 minimum samples to make a split at a decision node are all optimal parameters for the Random Forest model for both Raw and Distance datasets as overfitting is low and accuracy is high with these parameters.

The update rules and the structure affect the generalisation of the model. From Figure 6.6 it can be seen that the model overfits if it is allowed to make decision nodes using very few samples (e.g. 10) with a big difference between the training (raw 99% and distance 97%) and the testing (raw 75% and distance 74%) accuracies. As a rule of thumb, the more estimators the model has, the better is the performance, which comes at the expense of the training time (training raw 88% and distance 85% while testing raw 65% and distance 66% accuracies with 30 estimators). Tree depth (training raw 88% and distance 85% while testing raw 65% and distance 66% accuracies with 20 levels deep) and the maximum number of leaf nodes (training raw 88% and distance 85% while testing raw 65% and distance 66% accuracies with 800 leaf nodes) have the biggest impact on the accuracy of the model, applied both to the raw and distance hand features, but also contribute the most to the overfitting of the model.

Raw Feature	Raw Feature Importance (in %)	Distance Feature (between two fingers)	Distance Feature Importance (in %)
index_phalange_y	4	index_middle	9
ring_phalange_y	4	index_ring	7
thumb_phalange_y	3	thumb_little	7
thumb_phalange_x	3	thumb_middle	7
index_proximal_y	3	thumb_index	7

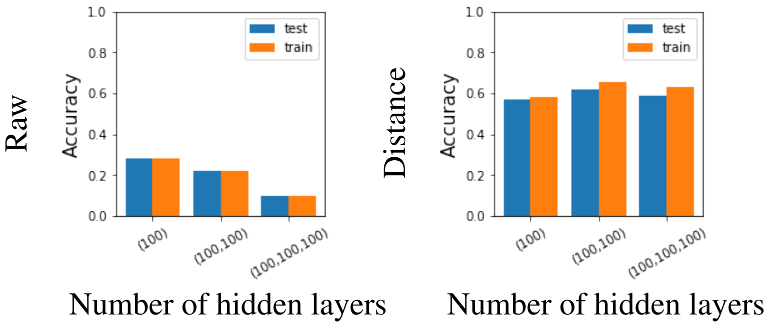
Table 6.3: Top 5 most significant features and their importance % (out of the total possible 100%), identified by the Random Forest model for the correct classification of the Raw and Distance datasets as described in the Figure 6.3 with all features summing to 100%

Table 6.3 shows feature importance for Raw and Distance datasets as described in Figure 6.3, inferred by the Random Forest model. Interestingly, index finger and the thumb play an important role in distinguishing the hand shapes using either raw key-points or distances features. The fact that the thumb is an important feature in sign languages is also supported in [193, 4].

6.2.2.3 Feed-Forward Neural Network (FFNN)

Feed-forward neural network is a data structure with every new layer introducing more non-linearity into the decision space. The error function is reduced over the epochs using stochastic gradient descent optimisation method.

Structure	**Input-1×Hidden-Output**, *Input*-2×*Hidden-Output*, *Input*-3×*Hidden-Output*
Activation Function	*ReLu*
Learning Rate initial	0.01
Cosine annealing	*False*
Optimiser	*Adam*
β_1	0.9
β_2	0.999
Epochs	200
Batch Size	32
Validation Fraction	0.1
Testing Fraction	0
Data Augmentation	*None*



(a) (b)

Figure 6.7: Feed-forward neural network is applied to both Raw and Distance hand features as described in Figure 6.3. Using one hidden layer with 100 nodes (in bold) is along with the rest of the hyperparameters shown above is optimal for both raw and distance datasets as it provides relatively good accuracy and no overfitting.

The choice of the parameters impacts the performance of the model. Figure 6.7 shows how different number of hidden layers affects the classification of both raw and distance features. Model, trained and tested on distance features performs better than the model trained and tested on the raw features and having a relatively shallow network performs better than having a deeper network (training raw 28% and distance 58% while testing raw 27% and distance 57% accuracies with one hidden layer with 100 nodes).

6.2.2.4 Convolutional Neural Network (CNN)

Convolutional neural network is composed of one or many convolution layers. These layers contain a set of kernels. Kernels are optimised during training and each kernel produces a feature map, which acts as a feature extractor for the raw images. In contrast, classical image processing uses hand-engineered kernels (e.g. vertical, horizontal, gaussian, laplacian filters) to transform the raw images [229]. However, learned kernels have shown to be superior to classical hand-crafted kernels [220].

CNN Filters	8,**32**
CNN Kernel Size	3,**5**
Dropout Rate	0.25
Structure	$Input - 1 \times CNN - Output$, $\mathbf{Input - 3\times\ CNN - Output}$
Activation Function	$ReLu$
Learning Rate initial	$1e-4$,$\mathbf{1e-2}$
Cosine Annealing	$True$, **False**
Optimiser	$Adam$
β_1	0.9
β_2	0.999
Epochs	200
Batch size	32
Validation Fraction	0.2
Testing Fraction	0.2
Data Augmentation	$(feature\text{-}wise)\ normalisation$

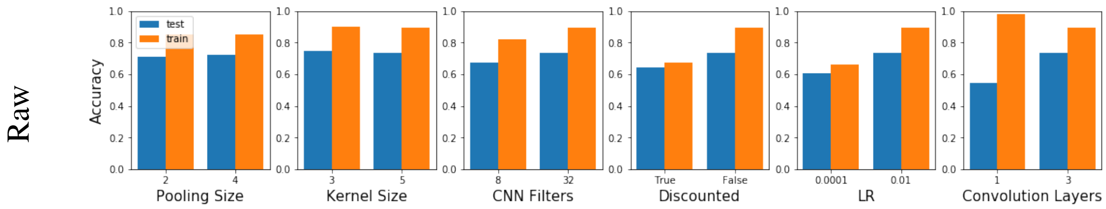


Figure 6.8: The convolutional neural network is applied to Raw Image dataset as shown in Figure 6.3. 32 filters of size 5 in each of 3 convolutional layers with constant learning rate have shown to provide the most optimal classification accuracy results.

Choice of the parameters affect the accuracy of the model as shown in Figure 6.8. Using low number of convolution layers (e.g. 1 layer) makes the model overfit as the difference between the training and test set is relatively big (training 98% and testing 54%). If the learning rate is discounted, the model is underfitting as it does not reach the same accuracy level as the model that does not discount the learning rate (training 67% and testing 64%). All the other parameters have little affect on the accuracy of the model trained on both raw and binary images.

6.2.2.5 Transfer Learning

CNN Filters	*Simonyan and Zisserman* [228]
CNN Kernel Size	
Convolution Layers	
Dropout Rate	
Structure	
Activation Function	
Learning Rate initial	$1e-4$
Cosine Annealing	*False*
Optimiser	*Adam*
β_1	0.9
β_2	0.999
Epochs	100
Batch size	32
Validation Fraction	0.2
Testing Fraction	0.2
Data Augmentation	$(feature\text{-}wise)$ *normalisation*

Figure 6.9: Parameters of the transfer learning model that uses the pre-trained Inception model for the feature extraction and fine-tuned on the Raw image dataset as described in Figure 6.3

In the case of the data set with the binary images, the model has been pre-trained on the MNIST data set [156] by modifying the size of the input to 128×128 to match the hand shape data set input size. The idea behind the pre-training on the MNIST data set was to train the model to learn such features as corners and edges, which could also benefit in classification of the hand shapes on the binary data set. In the case of the Inception network [241], the model has been pre-trained on the ImageNet data set and fine-tuned on the raw image dataset using parameters shown in Figure 6.9.

The idea behind the transfer learning is to provide a useful weight initialisation, which could result in the training beginning near the local or even global minimum in the search space. This, in turn, allows for much less data to be used to reach the minimum or can result in much faster convergence.

6.2.3 Results

Table 6.4: Combined accuracy results in % accuracy and confidence intervals for three optimisation runs on the test set of the various methods on both the images data set (Figure 6.3 a-b) and the raw key-points data set (Figure 6.3 c-d)

		Method				
	Data	k-NN	RF	FFNN	CNN	Transfer Learning
Images	Cropped Raw Images	—			76±0.010	**90±0.008**
	Binary Skeleton Hand Images	—			73±0.005	73±0.007
Raw Key-points	Raw Features	72±0.0	62±0.004	38±0.008	—	
	Distances Between Features	76±0.0	68±0.002	65±0.001	—	

Table 6.4 shows the performance of every model on the test set using the best parameter settings (marked in bold) from Sections 6.2.2.1-6.2.2.5. The average results are reported with the standard deviation after the three runs.

The most accurate model is the one fine-tuned on cropped raw images data set. Surprisingly, the binary data does not perform as well as expected and pre-training the model on the MNIST data set does not improve the performance of the fine-tuned model ($p = 0.5 > 0.05$ by the Wilcoxon signed-rank test). Models trained on the distances between the hand features data set perform better than the same models trained on the raw hand features ($p = 0.00338 < 0.05$ for k-NN, $p = 0.005868 < 0.05$ for RF, $p = 0.0003 < 0.05$ for FFNN by the Wilcoxon signed-rank test). This is expected as the distance data is more invariant to changes and contains fewer features.

6.3 Modelling Phonological Parameters using Multi-Label End-to-End Model

After modelling orientation and location phonological parameters in Section 4.1, as well as the handshape phonological parameters in Section 6.2, this section explores a single multi-label end-to-end model that would perform classification of location, orientation, and handshape phonological parameters at the same time using the HamNoSys notation. The dataset that was used for this modelling consists of the raw images as it was discovered that the classification of raw images using pre-trained model provides best accuracies in Section 6.2.

There has already been an investigation by Awad et al. [8] into sharing the features between the individual phonological parameters and how this improves modelling. This motivates the choice of an end-to-end model, which supports sharing of the learned features among the individual phonological parameters. Moreover, the model utilises the knowledge that location and orientation phonological parameters are co-dependent as will be shown in Section 6.3.1 and explicitly allows the classifier of one phonological parameter to affect another classifier of another phonological parameter during training and inference.

The advantage of an end-to-end model is that it can be trained all at once to describe different features and it is expected that a single model, trained end-to-end should incorporate all the necessary information to simultaneously describe these different complex features [106].

6.3.1 Data Pre-processing

Similarly to the hypothesis that not all fingers are independent and dominant during signing [193, 4], this section hypothesises that the phonological parameters have co-dependencies among them that could also be explained from the limited physiology point of view. This section sets out to verify whether there are co-dependencies among location and orientation phonological parameters. This means that if one phonological parameter is of a certain kind, then it is more likely for another phonological parameter to assume a specific configuration (e.g. it should be more complicated to have a hand above the head and point downwards than pointing downwards while having a hand at the the torso level). In that case, why modelling of signing should even consider such cases during optimisation?

$$C_{O_N,L_M} = \begin{pmatrix} c_{O_1,L_1} & \cdots & c_{O_1,L_M} \\ \vdots & \ddots & \vdots \\ c_{O_N,L_1} & & c_{O_N,L_M} \end{pmatrix}$$

$$S_{O_N,L_M} = \sum C_{O_N,L_M} \cap (C_{O_i,L_j} \cup \sum O_i \cup \sum L_j)$$

$$C_{2\times2} = \begin{pmatrix} C_{O_i,L_j} & \sum O_i \\ \sum L_j & S_{O_N,L_M} \end{pmatrix}$$

To show co-dependencies of phonological parameters formally, a global C_{O_N,L_M} contingency table is constructed that counts the occurrences for both orientation (O) and location (L) variables for every category that occur in the collected data (e.g. North, North-East, etc. for orientation and Shoulder, Neck, etc. for location, as defined by HamNoSys). Second, a series of local contingency tables $C_{2\times2}$ are constructed from the global C_{O_N,L_M} contingency table for every category of every variable as a post-hoc step. Finally, Bonferroni-adjusted p-value [25] was used to check if the presence of a particular orientation/location combination (C_{O_i,L_j}) in the data set is significant as opposed to other orientation/location combinations (S_{O_N,L_M}) by performing the Chi-square test of independence of variables for every $C_{2\times2}$ contingency table.

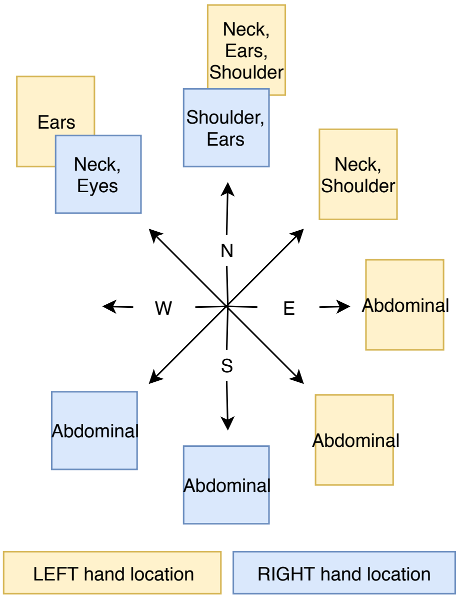


Figure 6.10: Chi-square test with Bonferroni-adjusted p-value= 0.0029 for significant co-dependencies in the OODT dataset between the location and orientation phonological parameters.

Figure 6.10 shows the significant co-dependencies among the orientation and location phonological parameters in the OODT dataset. The results indicate that it is more common in the data to encounter right hand pointing towards the western side as well as the north and the south, while it is more common for the left hand to point to the eastern side as well as the north and the south, but it is uncommon to point to the eastern side with the left hand and to the western side with the right hand. This has been pointed out by Cooper et al. [57] that only a subset of 'comfortable' combinations occurs in practice during signing.

Moreover, for both hands it is common to point to the northern side at the upper side of the body, while it is common for the both hands to point to the southern side at the lower part of the body.

6.3.2 Methodology

After discovering the existing co-dependence in the phonological parameter data in the OODT dataset in Section 6.3.1, this section shows how a single end-to-end model can be used to classify individual phonological parameters, taking the discovered co-dependence of location and orientation phonological parameters into the account.

video_frame	x	y	handedness	handshape	orientation	location
2169_0021.png	224	288	right	1	NE	shoulder
1604_0030.png	192	320	right	1	E	neutral
1249_0017.png	224	256	right	1	N	shoulder
444_0008.png	160	416	right	1	SE	neutral
182_0040.png	96	288	right	1	N	neutral

Table 6.5: Five samples from the multi-label dataset, where x and y columns refer to the fixed-size bounding box of a hand in the frame, handedness refers to whether it is a right or left hand, handshape refers to the handshape phonological parameter in Danish sign language, orientation and location refer to the orientation and location phonological parameters as defined by HamNoSys notation.

Table 6.5 shows five arbitrary instances from the annotation file. The first column shows the video and the frame that is being annotated, the second and the third columns show the (x, y) origin of a bounding box that has shape 128×128 pixels, which inscribes a hand in the frame. The fourth to seventh columns tell the handedness, handshape, orientation, and location phonological parameter categories as motivated by the HamNoSys notation system.

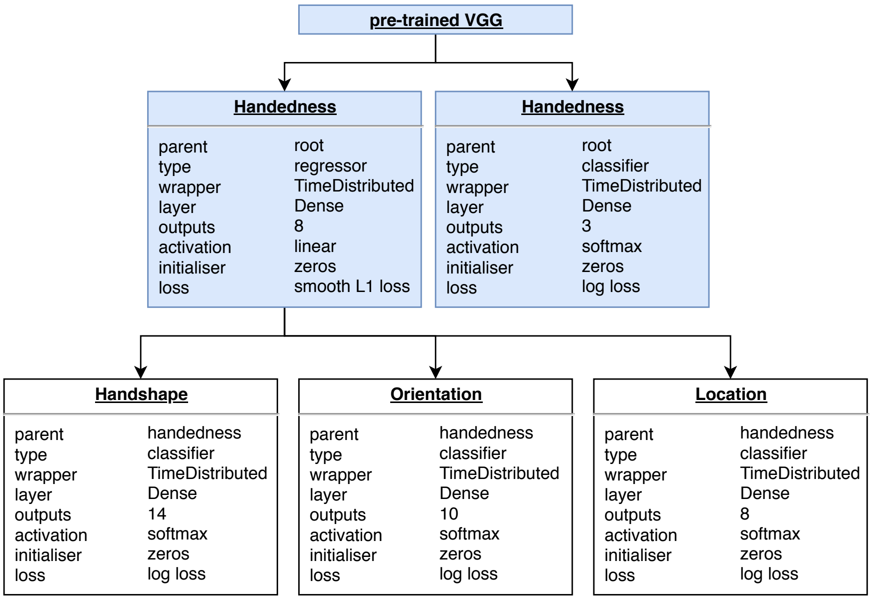


Figure 6.11: Multi-label Fast R-CNN model for detection and classification of individual phonological parameters. The model consists of the base model (in blue) that detects and classifies hands using the pre-trained VGG model. Handshape, orientation, and location correspond to classifiers of individual phonological parameters

Figure 6.11 shows the traditional single-label Fast Region-Based Convolutional Neural Network (Fast R-CNN) model (in blue) [105] that has been extended by adding multiple labels into the model, where every label corresponds to a classifier for individual phonological parameter (handshape, orientation, location). The model uses a pre-trained VGG [228] network as a feature extractor, allowing for both object detection and classification on raw images in a single pass of an input image through the model.

CNN Filters	*Girshick et al.* [105] (in blue) $+Handshape, Orientation, Location$ ($Figure$ 6.11)
CNN Kernel Size	
Convolution Layers	
Dropout Rate	
Structure	
Activation Function	
Learning Rate initial	$1e-5$
Cosine Annealing	$False$
Optimiser	$Adam$
β_1	0.9
β_2	0.999
Epochs	100
Batch size	300
Validation Fraction	0.2
Testing Fraction	0.2
Data Augmentation	$normalisation$

Figure 6.12: Parameters of the end-to-end model that uses the pre-trained VGG model for the feature extraction and fine-tuned on phonological parameters: handshape, orientation, and location

Figure 6.12 shows the parameters that have been used to train the network with the validation performed every 25 epochs to accelerate the training time. The interest in using this model lies in exploiting the label co-dependence that was shown to be present in the data to improve the classification accuracy of the model.

6.3.2.1 Separate Phonological Parameter Training

The first approach is designed to learn different classifiers separately. All the classifiers are trained independently and sequentially, starting with the handshape classifier. This handshape classifier is the first one to be trained as it potentially requires very fine features for the correct classification (e.g. phalanges etc) in comparison to location or orientation classifiers. Once the handshape classifier is trained, all the layers before this classifier are fixed and only the top layers that correspond to classifiers of individual phonological parameters are trained later. This ensures that the features learned by the convolutional layers during the training of the handshape classifier are used by other classifiers as well. Unfortunately, there is no standard quantitative way of showing whether the model has learned filters that focus on fine or coarse features of an image.

6.3.2.2 Joint Phonological Parameter Training

The second approach is to train all the classifiers at the same time with a combined loss function ($Loss_{Handedness} + Loss_{Handshape} + Loss_{Orientation} + Loss_{Location}$). In this case, the learned features have some significance for every single classifier.

6.3.3 Results

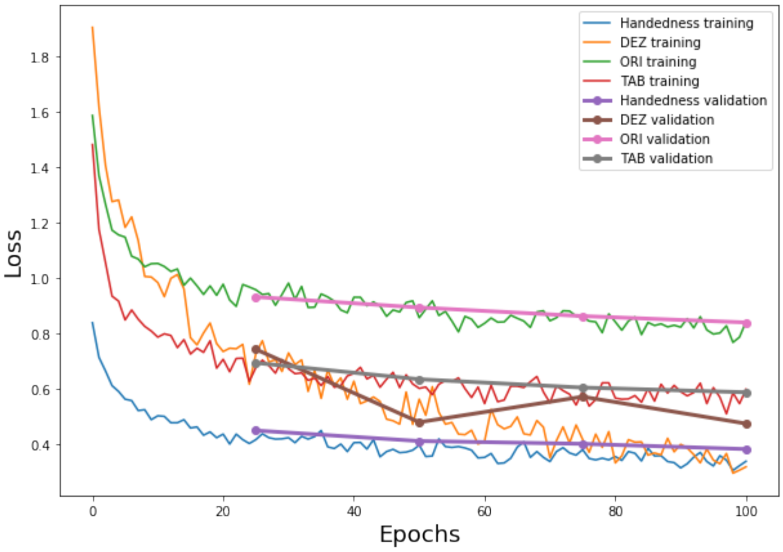


Figure 6.13: Training and validation process of the multi-label F-RCNN model for 100 epochs with every label classifier trained separately with shared weights being fixed after the first (Handshape) classifier is trained

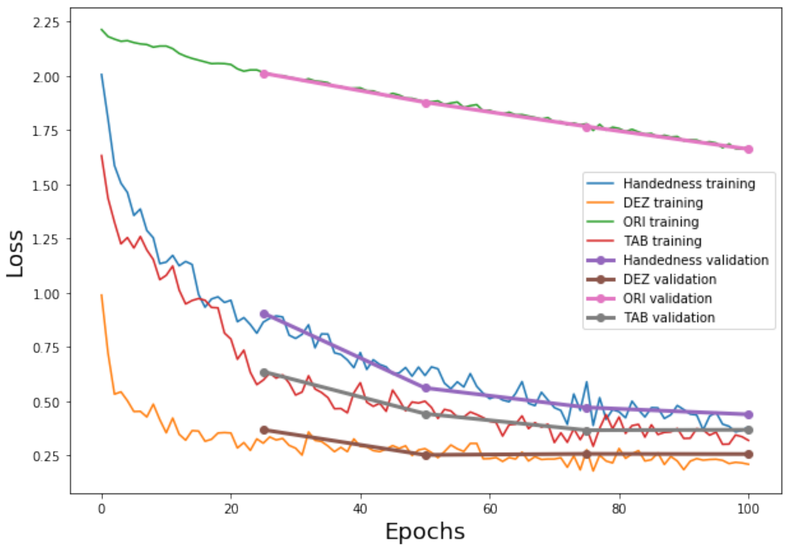


Figure 6.14: Training and validation process of the multi-label F-RCNN model for 100 epochs with all the labels being trained simultaneously

6.3.3.1 Separate Phonological Parameter Training

The first approach results in smooth training for every phonological parameter except for the handshape classifier, which starts to overfit after epoch 50, but then the regularisation is keeping the model from overfitting too much as can be seen from the Figure 6.13. Test set accuracies correspond to 82%, 88%, 27%, and 39% for the handshape, handedness, orientation, and location classifiers respectively.

6.3.3.2 Joint Phonological Parameter Training

The second approach results in the handshape classifier being underfitted, as the validation curve strives down as shown in Figure 6.14. The training is slower as opposed the method described in Section 6.3.3.1. This is understandable since in this approach a combined loss is considered. Test set accuracies result in 77%, 91%, 32%, and 56% for the handshape, Handedness, orientation, and location classifiers respectively.

6.3.3.3 Results Comparison

Table 6.6: Test results (in %) accuracy of the different training variations for the multi-label Fast R-CNN model on the test set

Phonological Parameter Classifier	**Separate**	**Joint**
Handshape	**82**	77
Handedness	88	**91**
Orientation	27	**32**
Location	39	**56**

Table 6.6 shows testing results of the model that was performed on the test set after the model was trained for 100 epochs.

In general, classifiers were underfitted due to the short training time. This can be observed with the handshape classifier when all the classifiers are trained jointly (Figure 6.14). The difference between the results in Tables 6.4 and 6.6 (90% vs 77%) can be explained by too short training time in the latter case (1000 versus 100 epochs) and the fact that cumulative loss in the latter case would require greater improvements for the handshape classifier for every training batch to improve the overall model.

6.3.3.4 Multi-Label Model for Individual Phonological Parameters

Finally, observing that the extra classifiers trained jointly improves the performance of the model as can be seen from Table 6.6, the final model with all the classifiers trained jointly for 500 epochs with handedness achieving 92%, handshape 87%, orientation 68%, and location 60% accuracies.

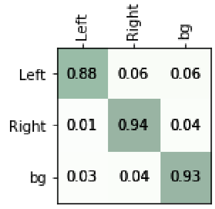


Figure 6.15: Handedness Confusion Matrix on the test set using the final multi-label FR-CNN model

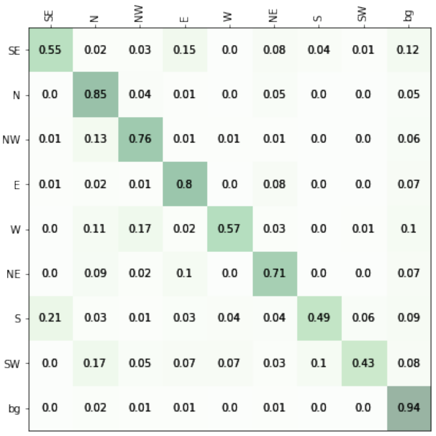


Figure 6.17: Orientation Confusion Matrix on the test set using the final multi-label FR-CNN model

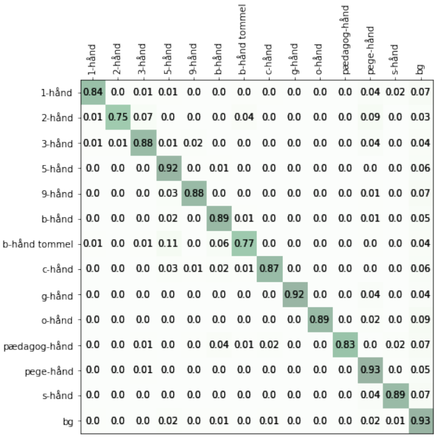


Figure 6.16: Handshape Confusion Matrix on the test set using the final multi-label FR-CNN model

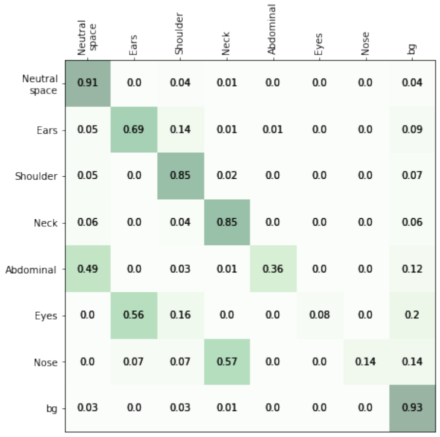


Figure 6.18: Location Confusion Matrix on the test set using the final multi-label FR-CNN model

Figures 6.15-6.18 shows the confusion matrices for the test set for all the classifiers in the multi-label F-RCNN model. There are very few cases of the misclassification for the handedness classifier with high recall and precision for the both hands. In the case of the handshape classifier, the b-hand tommel (b-hand with a finger to the side) is sometimes misclassified as the b-hand or 5-hand, which are the same hand shapes only without the finger to the side or with all the fingers spread out as can be seen in Figure 6.2. Koller et al. [142] reports similar results for the handshapes that are similar in both papers. As for the orientation classifier, lower recall can be seen in the cases of the south-west, south, south-east, and west orientations mostly being confused with the adjacent orientations. Finally, for the location low recall also can be seen for some cases. For the upper body parts, when a hand is next to the eyes the model most of the time thinks that the hand is next to the ears while if the hand is next to the nose, it is seen as if it was next to the neck. For the lower body parts, when a hand is next to the abdomen the system sees it as if it was in the neutral space.

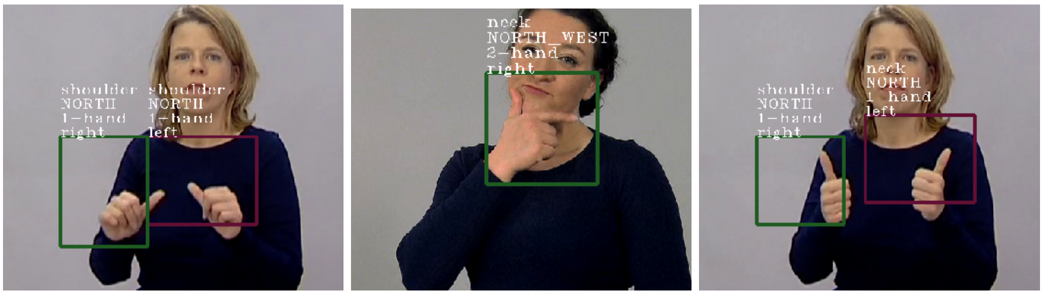


Figure 6.19: Visualisation of the classification of three arbitrary frames from the test set using the final multi-label FRCNN model

Figure 6.19 shows three arbitrary frames from the test set, processed by the multi-label F-RCNN model after it was trained for 500 epochs with all the classifiers trained at the same time and extra connections between the location and orientation classifiers to facilitate their co-dependence.

6.4 Discussion

This chapter explored modelling of handshape phonological parameter, which is difficult to estimate using pose key-point information and trigonometry as was done for location and orientation phonological parameters in Section 4.1.

Table 6.7: Related work results compared to our final model results in % on modelling individual phonological parameters

Phonological Parameter	**Reported result**		**Our Final Model**
Handedness	—		**92**
Handshape	75 (2D)	Bowden et al. [29]	87
	63 (2D)	Koller et al. [142]	
	82 (3D)	Dilsizian et al. [74]	
	99 (3D)	Demircioglu et al. [70]	
	68 (2D)	Kim et al. [164]	
Orientation	—		**68**
Location	31 (2D)	Cooper and Bowden [55]	**60**

Table 6.7 shows together the results of the final model compared to the results found in the literature. It is important to note that the results are not directly comparable, since the past work either focused on different sign languages, which in turn may have different handshapes.

Phonological parameters describe execution of signs and can be inferred directly from videos. The fact that the HamNoSys categories for phonological parameters are universal for all sign languages allows acquisition of data from multiple resources, which can support training of data-greedy deep-learning models, which in turn can produce better classification results. The proposed approach, however, identifies and incorporates co-dependence of two phonological parameters into the model to optimise the training process, showing that the resulting model is superior. It is worth exploring co-dependence of other phonological parameters, which could also be incorporated into the model to reduce the training time and increase the accuracy. For example, relative hand location could benefit from bounding box regressor (see Figure 6.11). However, this could be challenging as it cannot be assumed where exactly a person is positioned in the frame. In the future, the model should also incorporate other phonological parameters, such as movement. The model would have to be extended to include memory units (e.g. LSTM) to mimic sliding window behaviour described in Section 4.2.

One advantage of using a single model for classification of multiple phonological parameters in one pass is that it saves the inference time and can be done in real time with adequate hardware (e.g. GPU support). This is especially important when analysing large streams of sign language data, potentially in real time. Another benefit of a single model is that it has the potential to learn co-dependencies among different phonological parameters and thus make fewer mistakes, as opposed to having one method for each phonological parameter. This was the case in methods previously mentioned, for example in Section 4.1 for modelling location and orientation phonological parameters and Section 6.2 for classifying the handshape phonological parameter.

Chapter 7

Case Study: Data Mining for Sign Language Analysis and Comparison

Section 4.1 has shown how the spatial segmentation of the raw signing videos could be accomplished, transforming signing videos to HamNoSys notation, which is often used for the annotation of sign language resources, as can be seen in Table 2.1. The aim of this chapter is to explore further the significant co-dependent occurrences of the location and orientation phonological parameters, which would allow one to perform the comparison of different sign languages at the level of phonological parameters. The location and orientation phonological parameters are used in this chapter without the handshape as modelled in Chapter 6. This is because it has been shown that the location and orientation are less prone to co-articulation than the handshape phonological parameter [196]. In addition, the direct calculation of these two phonological parameters has been chosen for this chapter as it will lead to more definitive sign language comparison results. This is in contrast to the model reported in Chapter 6 that combines multiple phonological parameters into a single model and requires extensive training.

As deaf-specific music performers are rare, sign language users resort to 'listening' to interpretations of the songs found in the spoken or written languages that are being interpreted by those who can both hear and sign. This is evidenced by the relatively large amount of content found in online resources such as TikTok, YouTube, or Instagram. This chapter takes a look at two sign languages, ASL and Libras, which have relatively little historical relationship. This chapter investigates the signing behaviour of the song interpreters, while looking at three English songs from YouTube: 'Love Yourself' by Justin

Bieber, ‘Halo’ by Beyoncé, and ‘Love On The Brain’ by Rihanna. One video for every interpreted song for each sign language is collected for the analysis, making the total of six videos from six different signers. This chapter will quantify frequently occurring hand positioning during the signing and compare the prevailing hand positions and orientations between the two sign languages, aiming to show that sign languages evolve differently. The reason why interpreted songs are investigated is because we want to compare sign languages by looking at continuous signing in different sign languages that sign the same information. Therefore, this chapter answers the fifth research question (RQ5) posed in Section 1.3, which asks:

“Can two unrelated sign languages be compared using location and orientation phonological parameters?”

7.1 Methodology

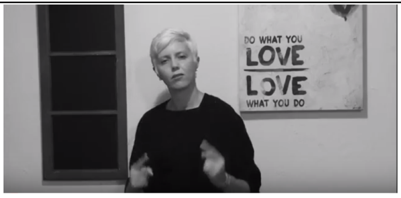
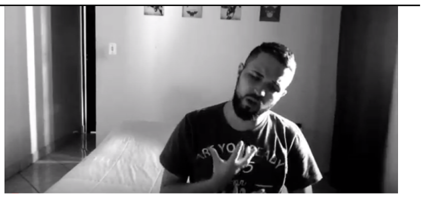
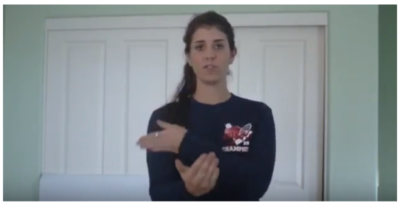
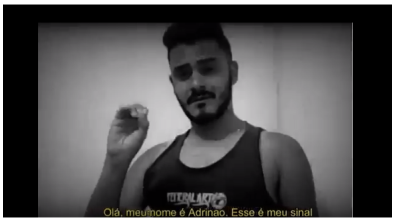
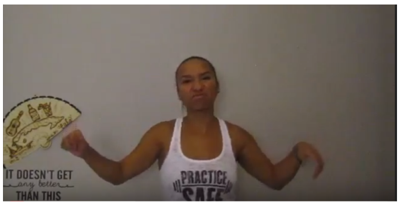
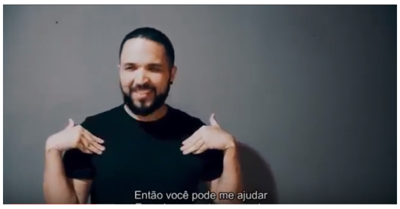
Song	ASL Screenshot	Libras Screenshot
Justin Bieber Love Yourself		
Beyoncé Halo		
Rihanna Love On The Brain	 IT DOESN'T GET any better THAN THIS PRACTICE SAFE	

Table 7.1: Screenshots of the collected online data from YouTube for the three songs by the three different artists with the English spoken language interpreted by six different signers, three signers per sign language

Table 7.1 shows screenshots of the collected online data for three English songs interpreted by six different interpreters in two different sign languages. From the screenshots, it can be seen that the proximity of the signer to the camera varies. Some videos are edited by applying black and white or vintage camera filters. Generally, the video variations are similar to the ones discovered in Chapter 5. As a general rule, there is no camera movement, but the signers usually move slightly to the rhythm of the song on the background. Data from online social-media resources tends to be very unpredictable. Therefore, the collected data has to be filtered. During the filtering, the online data that has no standard is turned into the data that has some pre-defined standard. The OpenPose library helps to apply simple filters to the raw data, discarding all the content that has more than one signer at the same time in one frame or frames affected by any heavy obstructions or occlusions. As a result, the content that has too few key-points visible is discarded, as it is essential to see the upper body and the hands to make sense of the signing [169]. In addition, all the extracted key-points are then normalised by forcing a fixed distance between shoulders and bringing the signer to the centre of the frame, as suggested in Chapter 5. By performing such filtering, a quality standard is enforced upon the collected online data. However, the context and the signer profile remain unknown at this stage.

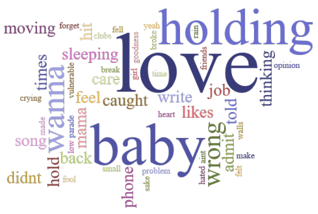
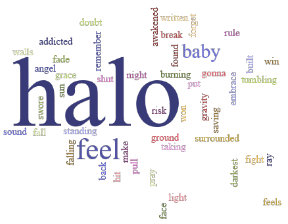
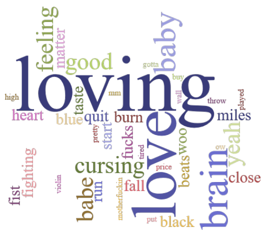
Song	Justin Bieber Love Yourself	Beyoncé Halo	Rihanna Love On The Brain
Lyrics Word Cloud			

Table 7.2: The word cloud generated from the lyrics for the three English language songs obtained online.

Table 7.2 shows the word cloud for the three songs generated from the lyrics obtained online. The purpose of the word cloud is to give insight into which words are frequent in the lyrics. As it can be seen, the lyrics for all the songs often mention love and romantic feelings.

After the filtering of the online data, a frequency analysis is performed on the location and orientation by extracting these phonological parameters, as described in Section 4.1. Later, the frequency of occurrences of specific location/orientation combinations in the collected filtered data is calculated by counting such occurrences. Once the data has been filtered and the patterns have been discovered, the information was acquired on 43,016 hand locations and the same number for the hand orientations for ASL and 38,258 for both hand location and orientation for Libras for the interpreted three songs. This chapter explores and analyses the co-dependence of phonological parameters for each hand, comparing the significant co-dependences across the two sign languages.

7.2 Results

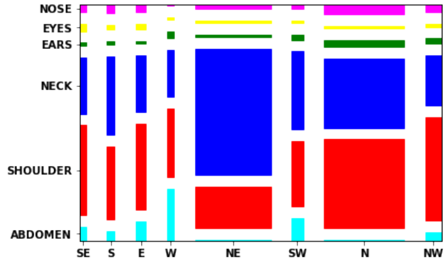
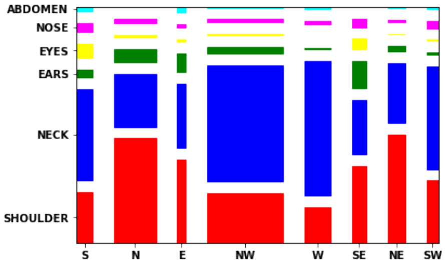
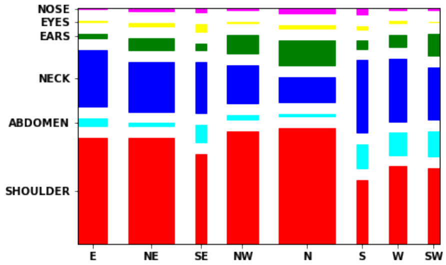
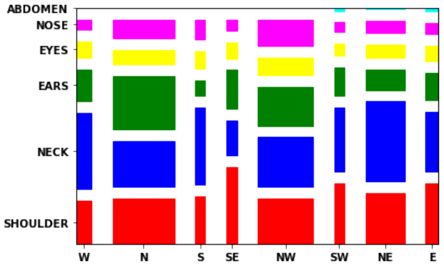
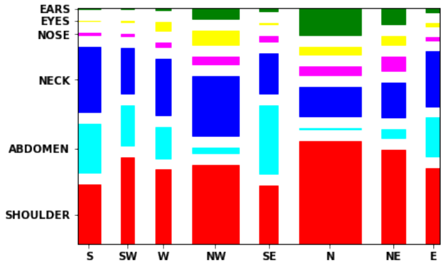
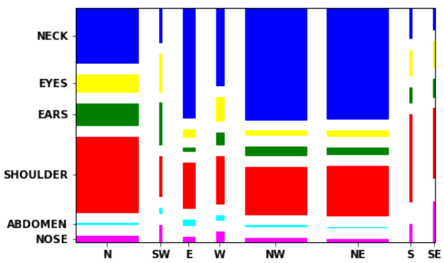
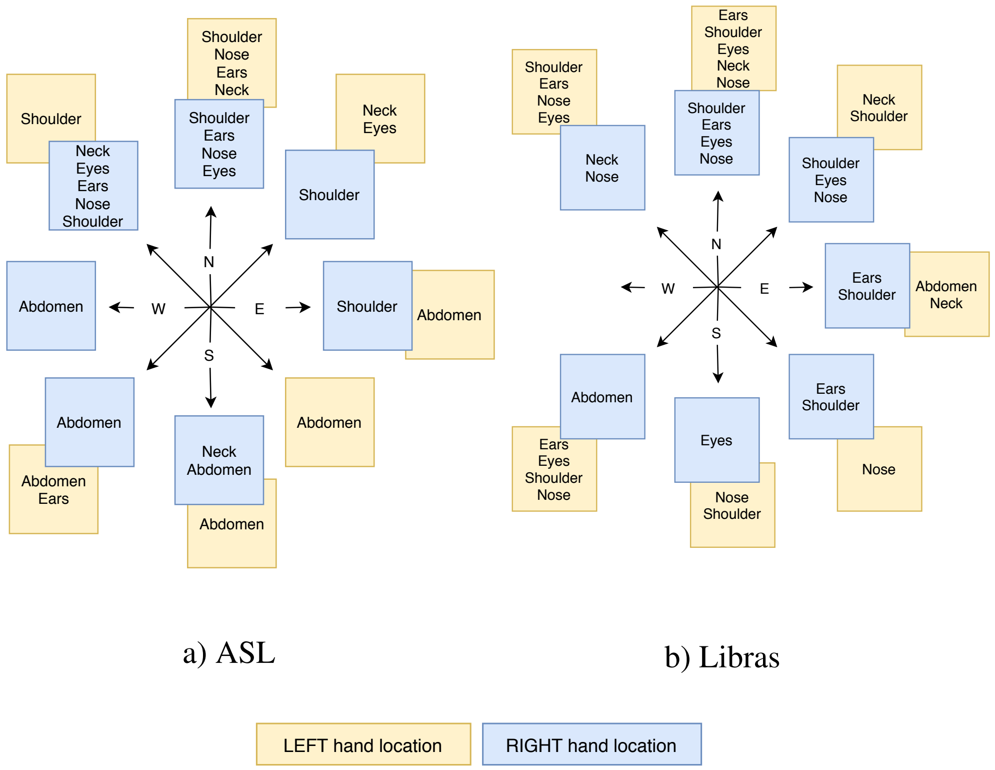
Song	ASL HamNoSys	Libras HamNoSys
Justin Bieber Love Yourself		
Beyoncé Halo		
Rihanna Love On The Brain		

Table 7.3: Location/orientation relative frequencies for each video for each sign language

Table 7.3 shows the relative frequencies of the location/orientation combinations for each video and each sign language. It can be observed that the Libras, on one hand, has less abdomen activity than ASL (indicated in light blue) while, on the other hand, Libras has more neck and ears activity than ASL (dark blue and green respectively). Both sign languages have more pointing up direction of the hands as opposed to the other possible directions (wider NE/N/NW columns).

Having visually analysed the frequencies of the location/orientation combinations, it is now of the interest to find the significant combinations for each hand that prevails in the collected data and compare the two sign languages based on this analysis.



a) ASL

b) Libras

Figure 7.1: Significant location and orientation phonological parameter co-dependences in a) ASL and b) Libras with Bonferroni-adjusted Chi-squared p-value < 0.001 for both sign languages

Figure 7.1 shows significant location/orientation co-dependences for each sign language after the Bonferroni-adjusted Chi-squared p-value analysis. It can be seen that the both hands tend to point up at the upper side of the body, which is similar for both the sign languages. Libras, however, has more activity with both hands at the upper part of the body than ASL. As a matter of fact, Libras has more activity with both hands around all the parts of the body. In ASL, on the other hand, the left hand is less mobile than the right hand.

This could be explained by the fact that the signers in Libras were left-handed, but there is not information available to verify such a speculation. Some significant co-dependences are unusual, for example, pointing down at the upper body level, which may feel unnatural and slightly contradicts the past findings by [57] stating that a subset of the 'comfortable' hand configurations are assumed more often during the signing, independent of the sign language. This can also be explained by the fact that the signers in the video are slightly dancing to the music, which may affect the signing orientation.

7.3 Discussion

This chapter has showed the mining of the sign language data acquired online. A pipeline was created that downloaded videos of the interpreted songs from the internet, applied filtering and analysed the signing patterns using the phonological parameters. The chapter compared the two historically different sign languages (ASL and Libras) by their location/orientation co-dependencies present in the collected data and showed that, despite there being little historical background of the two languages interacting, they still share similar signing patterns with small variations in the flexibility of the hands, which can be explained by the fact that people converge to the usage of the 'comfortable' hand configurations as speculated by Cooper et al. [42]. It is worth mentioning that the co-dependence analysis results of the two languages may change with the data. For example, if songs with a different sentiment were taken for the analysis.

Determining the similarity between the execution of different (unrelated) sign languages has the potential to influence the sign language modelling, where signs are encoded using phonological parameters as in Cooper et al. [57]. For example, Chapter 6 has showed that it is possible to model orientation, location, and handshape phonological parameters using a single end-to-end model. Therefore, having a single model for the recognition of phonological parameters in one language could require minimal re-training or fine-tunning when training the model for a different sign language, assuming that the signs are executed using similar co-dependence of phonological parameters, as in the case with ASL and Libras as can be seen in Figure 7.1.

Chapter 8

Conclusion and Future Work

Throughout the thesis, it became obvious that the raw footage of continuous sign language data should be pre-processed in order to reduce its complexity and make it suitable for sharing on the internet, reducing the storage needs as the overall sign language data present on the internet is growing. Additionally, reducing the complexity of sign language data would allow for faster data mining and data analysis purposes, which would make it possible to process and analyse sign language data in real time as the datasets in the future could be expanded in real time by scraping the web. Two such data pre-processing methods have been used in this thesis: kinematic key-points and phonological parameters using HamNoSys notation. Table 8.1 revisits and summarises the research questions, the findings, and shows the proposed future work for the posed research questions.

Table 8.1: Research questions, discovered findings, and the future work of this thesis

#	Research Question	Findings	Future Work
RQ1	Can a small number of BSL sentences be modelled at the gloss level using written English language models, knowing that the grammar of the two languages is different?	Neural models with transfer techniques are effective for language modelling but difference between language syntax remains a barrier	Temporal probability distribution could be explored in classification of phonological parameters
RQ2	Can movement speed tracking be used in identifying sign boundaries in continuous signing?	Tracking the change in the movement enables identification of phoneme and sign boundaries	Dynamic thresholds could be explored based on a signer's average signing speed
RQ3	Can similar signs or phrases be found in different continuous signing videos without having the transcription?	A distance metric between phonological parameters based on linguistic features allows for searching and comparing of phonemes, signs, and phrases	While co-articulation can be tackled with lower thresholds, a different solution is needed for inter-signer variations (e.g. dialects)
RQ4	Can automatic classification of phonological parameters (e.g. location and orientation) be improved by exploiting their co-dependencies?	Exploitation of co-dependence among phonological parameters in an end-to-end model speeds up the model training and leads to more accurate models	Exploration of additional co-dependencies among phonological parameters with the aim of reducing the search space further when training models based on the phonological parameters
RQ5	Can two unrelated sign languages be compared using location and orientation phonological parameters?	Significant co-dependencies among phonological parameters found in the data were found to be effective when comparing different sign languages	Explore the influence of co-dependence of other phonological parameters when comparing different sign languages

First, this work modelled BSL glosses using an English language model, showing that although it is possible to reuse a model pre-trained on English language for sign language modelling, specific to sign language syntactic constructs that correspond to the movement phonological parameter do not exist in written English. This considerably limits transfer learning for sign language modelling using written languages. Nevertheless, language modelling techniques or modelling the distribution of most likely events could be used when classifying phonological parameters in order to reduce the classification errors in any future work.

Second, temporal segmentation was explored for segmenting both individual signs and phonemes using the variation of the hand movement speed during signing. Despite the method being motivated linguistically, the threshold should be adapted to tackle signing variations among signers, situations, and topics in the future work. Moreover, to perform such segmentations, pre-processed signing data should be normalised as no assumptions should be made about signers' location with respect to the camera, orientation, or body shape.

Third, segmented phonemes were clustered, observing that a geometrical distance that compares phonological parameters produces acceptable clustering results. Moreover, the same phoneme similarity measure was shown to be used for clustering or searching for phonemes, signs, or even phrases in continuous signing. Such unsupervised approach to data mining can be useful for simplifying data annotation processes and dataset creation in the future.

Fourth, acknowledging the importance of the phonological parameters in sign language, an end-to-end model is designed to reflect observed co-dependence among location and orientation phonological parameters. The model is trained to show that the encoding of the co-dependence into the model leads to better classification results. A single model that predicts multiple phonological parameters has an advantage over having multiple models for predicting individual phonological parameters. This is due to the fact that having a single model for multiple phonological parameters saves training time needed for classification. In addition, a single model that is trained to classify multiple phonological parameters can reduce the classification error, as it avoids classification of combinations that do not prevail in the training data.

Fifth, orientation and location phonological parameters were used for comparison of the two different sign languages. The data was collected online and included interpretations of English songs in the two sign languages. The analysis of the signing has showed that although the two languages are unrelated, the common combinations of location and orientation phonological parameters are comparable, which indicates that the sign languages could be converging with respect to similar kinematic configurations. If this is the case, then the models for a particular sign language recognition or translation system that rely on phonological parameters could require minimal re-training to adapt to another sign language.

Over the course of this thesis multiple questions were raised regarding the complexity of sign language data. In particular, it has been recognised that the modelling methods that use static thresholds are not sufficient due to the variance in signing present not only across signers, but also for the same signer in different contexts. It has also been recognised that sign language is fluid and that its vocabulary is large and having a single model for recognising all the lexical items could be infeasible using existing modelling techniques. In addition, resorting to glosses should not be used as a proxy for the actual visual stimuli as glosses lack grammatical constructs specific to sign languages. It is important to acknowledge the fact that any reduced representation of sign language data (glosses, ID-Glosses, kinematic key-points, phonological parameters, etc.) would carry some information loss from the original raw footage due to the fact that sign languages are multi-channel. They convey information about objects, their descriptions, and make use of the signing space simultaneously in order to interact with objects or subjects placed in that space previously, which can be challenging to capture using simpler data structures.

Throughout this thesis, various sign language resources were used mostly due to the easier access to these resources, their quality and annotation schemes appropriate for the individual research questions. The choice of the resources is also a major limitation of this thesis as there is a lack of strong baselines from similar research for these resources. For example, Chapters 3, 5, and 6 are forced to establish their own baselines since similar approaches in literature were tested on different datasets and previously reported results do not reflect the results of the models developed in this thesis. Another limitation of the thesis is the focus on isolated methods and the lack of a single application for the developed methods that would combine the proposed methods. The choice of an application area is left for future work, where individual methods and techniques described in this thesis could be combined together for such purposes as sign language recognition, sign spotting, sign language translation, or assisted sign language resource annotation. We hope that research in these domains would adopt techniques discussed in this thesis.

It was mentioned multiple times that the lack of standards in sign language is a significant issue that prevents collaborations from happening and interferes with the progress in both the Computer Science and Linguistics fields. In the NLP domain, for example, the audio annotation standard comes in the form of text, which allowed for the separate evolution of the fields of speech recognition and text processing. Sign language research would benefit from an intermediate representation level, which would serve as a point of partitioning between the two sides working on the overall problem of sign language processing. Kinematic key-points and phonological parameters could be good candidates, as these two intermediate levels offer detailed description of signing and can be used for sign recognition (continuous and isolated), generation (avatar movement), and translation purposes. Another setback in the standards is the lack of comprehensive datasets and consistent annotations, which would allow resources to be combined, for example, for training a model to recognise various phonological parameters. For this task, different sign languages could potentially be combined thus increasing the numbers and variances in participants, recording qualities, and background conditions.

Due to the fact that sign language has a rather fluid nature, variations in dialects, socio-economic backgrounds, and demographic profiles pose extra challenges to the models developed for sign language processing. Existing models are designed to fit all the variations to an average representation of the data, which could be detrimental to the models deployed in the real-world scenarios (e.g. understanding a new signer or a new dialect). Therefore, current metrics, such as perplexity, word error rate, and accuracy are not sufficient for the problems where deviations from an average tend to be the norm.

It could be tempting to suggest a more researched communication medium (i.e. texting) for the signing community in order to overcome the modelling challenges presented in this thesis. However, just like any of the virtual assistants that listen to voice commands, which is a natural way of communication for hearing people, deaf community would benefit as much utilising such technologies. Since the deaf community is not concentrated in one place and representatives usually spend their time in the hearing world, existing devices should be able to support a diverse range of users with various abilities.

This is the time to switch from the current paradigm of collecting datasets in controlled environments with low number of participants with the aim of collecting restricted numbers of examples. Despite the lack of confidence in the data acquired online, it offers a high degree of variance, which is more representative of the real-world signing. To facilitate the collection of such resources, future work would benefit from creating a platform that would invite signers from all around the world to share their stories, opinions, and creative content in a fun and informative way.

8.1 Published Work derived from this PhD

B. Mocialov, P. A. Vargas, and M. S. Couceiro. Towards the evolution of indirect communication for social robots. *In Proceedings of the 2016 IEEE Symposium Series on Computational Intelligence (SSCI)*, Athens, Greece. 2016.

B. Mocialov, G. Turner, K.S. Lohan, and H. Hastie. Towards Continuous Sign Language Recognition with Deep Learning. *In Proceedings of the Workshop on Creating Meaning with Robot Assistants: The Gap Left by Smart Devices*, Humanoids, UK. 2017.

B. Mocialov, G. Turner, H. and Hastie. Transfer Learning for British Sign Language Modelling. *In Proceedings of the 5th Workshop on NLP for Similar Languages, Varieties and Dialects (VarDial)*, COLING, USA. 2018.

B. Mocialov, G. Turner, H. and Hastie. Towards Large-Scale Data Mining for Data-Driven Analysis of Sign Languages. *In Proceedings of the 9th Workshop on the Representation and Processing of Sign Languages: Sign Language Resources in the Service of the Language Community, Technological Challenges and Application Perspectives*, LREC, France. 2020.

8.2 In submission at time of PhD publication

B. Mocialov, G. Turner, H. Hastie. Classification of Phonological Parameters in Sign Languages.

B. Mocialov, G. Turner, H. Hastie. Unsupervised Sign Language Phoneme Clustering using HamNoSys Notation.

Bibliography

[1] R. Akmeliawati, F. Dadgostar, S. Demidenko, N. Gamage, Y. C. Kuang, C. Messom, M. Ooi, A. Sarrafzadeh, and G. SenGupta. Towards real-time sign language analysis via markerless gesture tracking. In *IEEE Instrumentation and Measurement Technology Conference*, 2009.

[2] Y. Al-Onaizan and K. Papineni. Distortion models for statistical machine translation. In *Proceedings of the International Conference on Computational Linguistics (COLING)*, 2006.

[3] L. K. Alejandro Oviedo, Thomas Kaul and R. Griebel. The Cologne corpus of German Sign Language as L2 (C/CSL2): Current development stand. In *Proceedings of the International Conference on Language Resources and Evaluation (LREC)*, may 2018.

[4] J. Ann. On the relation between ease of articulation and frequency of occurrence of handshapes in two sign languages. *Lingua*, 98(1-3):19–41, 1996.

[5] V. Athitsos, C. Neidle, S. Sclaroff, J. Nash, A. Stefan, A. Thangali, H. Wang, and Q. Yuan. Large lexicon project: American Sign Language video corpus and sign language indexing/retrieval algorithms. In *Proceedings of the Workshop on the Representation and Processing of Sign Languages*, 2010.

[6] K. Audhkhasi, A. Sethy, and B. Ramabhadran. Diverse embedding neural network language models. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2014.

[7] S. Auephanwiriyakul, S. Phitakwinai, W. Suttapak, P. Chanda, and N. Theera-Umpon. Thai sign language translation using scale invariant feature transform and hidden markov models. *Pattern Recognition Letters*, 34(11):1291 – 1298, 2013.

[8] G. Awad, J. Han, and A. Sutherland. Novel boosting framework for subunit-based sign language recognition. In *Proceedings of the International Conference on Image Processing (ICIP)*, pages 2729–2732, Nov 2009.

[9] M. Bacchiani, M. Riley, B. Roark, and R. Sproat. Map adaptation of stochastic grammars. *Computer Speech & Language*, 20(1):41–68, 2006.

[10] D. Bahdanau, K. Cho, and Y. Bengio. Neural machine translation by jointly learning to align and translate. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2014.

[11] A. Baker, B. van den Bogaerde, R. Pfau, and G. Schermer. *The Linguistics of Sign Languages: An Introduction*. John Benjamins Publishing Company, 2016.

[12] C. Baker. How does "Sim-Com" fit into a bilingual approach to education. In *Proceedings of the National Symposium on Sign Language Research and Teaching*, 1978.

[13] A. Balvet. Issues underlying a common sign language corpora annotation scheme. In *Proceedings of the Workshop on the Representation and Processing of Sign Languages*, 2010.

[14] K. Bantupalli and Y. Xie. American sign language recognition using deep learning and computer vision. In *Proceedings of the International Conference on Big Data (BigData)*, pages 4896–4899. IEEE, 2018.

[15] K. Bantupalli and Y. Xie. American Sign Language recognition using deep learning and computer vision. In *Proceedings of the International Conference on Big Data (BigData)*, pages 4896–4899, Dec 2018.

[16] P. Battaglia, R. Pascanu, M. Lai, D. Jimenez Rezende, and k. kavukcuoglu. Interaction networks for learning about objects, relations and physics. In *Proceedings of the Advances in Neural Information Processing Systems (NeurIPS)*, pages 4502–4510, 2016.

[17] B. Bauer and H. Hienz. Relevant features for video-based continuous sign language recognition. In *Proceedings of the International Conference on Automatic Face and Gesture Recognition (FG)*, pages 440–445, 2000.

[18] J. R. Bellegarda. Statistical language model adaptation: review and perspectives. *Speech communication*, 42(1):93–108, 2004.

[19] Y. Bengio, A. Courville, and P. Vincent. Representation learning: A review and new perspectives. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(8):1798–1828, 2013.

[20] Y. Bengio, R. Ducharme, P. Vincent, and C. Jauvin. A neural probabilistic language model. *Machine Learning Research*, 3(Feb):1137–1155, 2003.

[21] Y. Bengio et al. Learning deep architectures for AI. *Foundations and Trends® in Machine Learning*, 2(1):1–127, 2009.

[22] M. Bhuyan, D. Ghosh, and P. Bora. Continuous hand gesture segmentation and co-articulation detection. In *Proceedings of the Computer Vision, Graphics, and Image Processing (CVGIP)*, pages 564–575. Springer, 2006.

[23] H. Birk, T. B. Moeslund, and C. B. Madsen. Real-time recognition of hand alphabet gestures using principal component analysis. In *Proceedings of the Scandinavian Conference on Image Analysis (SCIA)*, 1997.

[24] C. M. Bishop. *Pattern recognition and machine learning*. springer, 2006.

[25] J. M. Bland and D. G. Altman. Multiple significance tests: the Bonferroni method. *British Medical Journal (BMJ)*, 310(6973):170, 1995.

[26] C. Börstell and R. Östling. Visualizing lects in a sign language corpus: Mining lexical variation data in lects of Swedish Sign Language. In *Proceedings of the International Conference on Language Resources and Evaluation (LREC)*, pages 13–18, 2016.

[27] C. Börstell, M. Wirén, J. Mesch, and M. Gärdenfors. Towards an annotation of syntactic structure in the Swedish Sign Language corpus. In *Proceedings of the Workshop on the Representation and Processing of Sign Languages*, pages 19–24, 2016.

[28] L. Bottou, F. E. Curtis, and J. Nocedal. Optimization methods for large-scale machine learning. *SIAM Review (SIREV)*, 60(2):223–311, 2018.

[29] R. Bowden, D. Windridge, T. Kadir, A. Zisserman, and M. Brady. A linguistic feature vector for the visual interpretation of sign language. In T. Pajdla and J. Matas, editors, *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 390–401, Berlin, Heidelberg, 2004. Springer Berlin Heidelberg.

[30] C. Bowern and B. Evans, editors. *Sign Languages in their Historical Context*, pages 442–465. Routledge, 2015.

[31] D. Bragg, O. Koller, M. Bellard, L. Berke, P. Boudreault, A. Braffort, N. Caselli, M. Huenerfauth, H. Kacorri, T. Verhoef, et al. Sign language recognition, generation, and translation: An interdisciplinary perspective. In *Porceedings of the International Conference on Computers and Accessibility (SIGACCESS)*, 2019.

[32] L. Breiman. Random forests. *Machine Learning*, 45(1):5–32, 2001.

[33] M. Brennan. *The visual world of BSL: An introduction.* Faber and Faber, 1992. In David Brien (Ed.), Dictionary of British Sign Language/English.

[34] D. Brentari. Sign language phonology. In *The handbook of phonological theory*. John Wiley & Sons, 2011.

[35] P. Buehler, M. Everingham, and A. Zisserman. Employing signed TV broadcasts for automated learning of British Sign Language. In *Proceedings of the Workshop on the Representation and Processing of Sign Languages*, 2010.

[36] P. Buehler, A. Zisserman, and M. Everingham. Learning sign language by watching TV (using weakly aligned subtitles). In *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2961–2968, 2009.

[37] I. Bulyko, S. Matsoukas, R. Schwartz, L. Nguyen, and J. Makhoul. Language model adaptation in machine translation from speech. In *Proceedings of the International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2007.

[38] J. Bungeroth, D. Stein, P. Dreuw, H. Ney, S. Morrissey, A. Way, and L. van Zijl. The ATIS sign language corpus. In *Proceedings of the International Conference on Language Resources and Evaluation (LREC)*, May 2008.

[39] Z. Cao, G. Hidalgo, T. Simon, S.-E. Wei, and Y. Sheikh. OpenPose: realtime multi-person 2D pose estimation using Part Affinity Fields. In *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[40] Z. Cao, T. Simon, S.-E. Wei, and Y. Sheikh. Realtime multi-person 2D pose estimation using part affinity fields. In *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7291–7299, 2017.

[41] F. C. Capovilla, M. Duduchi, W. D. Raphael, R. D. Luz, D. Rozados, A. G. Capovilla, and E. C. Macedo. Brazilian sign language lexicography and technology: Dictionary, digital encyclopedia, chereme-based sign retrieval, and quadriplegic deaf communication systems. *Sign Language Studies*, pages 393–430, 2003.

[42] H. Cate, F. Dalvi, and Z. Hussain. Sign language recognition using temporal classification. *arXiv preprint arXiv:1701.01875*, 2017.

[43] H. Cate and Z. Hussain. Bidirectional American Sign Language to English translation. *pre-print*, 2017. http://arxiv.org/abs/1701.02795.

[44] X. Chai, H. Wang, and X. Chen. The Devisign large vocabulary of Chinese Sign Language database and baseline evaluations. Technical report, Technical report VIPL-TR-14-SLR-001. Key Lab of Intelligent Information Processing of Chinese Academy of Sciences (CAS), Institute of Computing Technology, CAS, 2014.

[45] B. Chaudhary. Real-time translation of sign language into text. Technical report, Data Science Retreat, apr 2017.

[46] S. F. Chen and J. Goodman. An empirical study of smoothing techniques for language modeling. *Computer Speech & Language*, 13(4):359–394, 1999.

[47] W. Chen, R. Fujiki, D. Arita, and R.-i. Taniguchi. Real-time 3D hand shape estimation based on image feature analysis and inverse kinematics. In *Proceedings of the International Conference on Image Analysis and Processing (ICIAP)*, pages 247–252, 2007.

[48] Y. Chen, J. Yuan, Q. You, and J. Luo. Twitter sentiment analysis via bi-sense emoji embedding and attention-based LSTM. In *Proceedings of the International Conference on Multimedia*, pages 117–125, 2018.

[49] M. J. Cheok, Z. Omar, and M. H. Jaward. A review of hand gesture and sign language recognition techniques. *International Journal of Machine Learning and Cybernetics*, 10(1):131–153, 2019.

[50] A. Choudhury, A. K. Talukdar, M. K. Bhuyan, and K. K. Sarma. Movement epenthesis detection for continuous sign language recognition. *Journal of Intelligent Systems*, 26(3):471–481, 2017.

[51] C. H. Chuan, E. Regina, and C. Guardino. American Sign Language recognition using Leap motion sensor. In *Proceedings of the International Conference on Machine Learning and Applications (ICMLA)*, pages 541–544, Dec 2014.

[52] N. Cihan Camgoz, S. Hadfield, O. Koller, H. Ney, and R. Bowden. Neural sign language translation. In *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7784–7793, 2018.

[53] B. Clark and G. Clark. SiLOrB and Signotate: A proposal for lexicography and corpus-building via the transcription, annotation, and writing of signs. In *Proceedings of the International Conference on Language Resources and Evaluation (LREC)*, may 2018.

[54] B. Coifman, D. Beymer, P. McLauchlan, and J. Malik. A real-time computer vision system for vehicle tracking and traffic surveillance. *Transportation Research – Part C: Emerging Technologies*, 6(4):271–288, 1998.

[55] H. Cooper and R. Bowden. Large lexicon detection of sign language. In *Proceedings of the International Workshop on Human-Computer Interaction (HCI)*, pages 88–97, 2007.

[56] H. Cooper, B. Holt, and R. Bowden. *Sign Language Recognition*. Springer London, London, 2011.

[57] H. Cooper, E.-J. Ong, N. Pugeault, and R. Bowden. Sign language recognition using sub-units. *Machine Learning Research*, 13(Jul):2205–2231, 2012.

[58] K. Cormier, J. Fenlon, S. Gulamani, and S. Smith. BSL Corpus annotation conventions, 2014. Deafness Cognition and Language (DCAL) Research Centre.

[59] T. Cover and P. Hart. Nearest neighbor pattern classification. *IEEE Transactions on Information Theory*, 13(1):21–27, 1967.

[60] O. Crasborn and R. Bank. An annotation scheme for the linguistic study of mouth actions in sign languages. In *Proceedings of the Workshop on the Representation and Processing of Sign Languages*, 2014.

[61] O. Crasborn, E. van der kooij, D. Waters, B. Woll, and J. Mesch. Frequency distribution and spreading behavior of different types of mouth actions in three sign languages. *Sign Language & Linguistics*, 11:45–67, 12 2008.

[62] O. Crasborn and I. Zwitserlood. The Corpus NGT: an online corpus for professionals and laymen. In *Proceedings of the Workshop on the Representation and Processing of Sign Languages*, 01 2008.

[63] O. Crasborn, I. Zwitserlood, E. van der Kooij, and R. Bank. A corpus-based lexical database for Sign Language of the Netherlands, 2017. Radboud University, Centre for Language Studies.

[64] O. A. Crasborn, J. Mesch, D. Waters, A. Nonhebel, E. Van der Kooij, B. Woll, and B. Bergman. Sharing sign language data online: Experiences from the echo project. *International Journal of Corpus Linguistics (IJCL)*, 12(4):535–562, 2007.

[65] A. Curiel and C. Collet. Implementation of an automatic sign language lexical annotation framework based on propositional dynamic logic. In *Proceedings of the International Conference on Language Resources and Evaluation (LREC)*, 2014.

[66] D. Das and A. T. A. Martins. A survey automatic text summarization, 2007. Literature Survey for the Language and Statistics II course at CMU.

[67] A. de Brébisson, É. Simon, A. Auvolat, P. Vincent, and Y. Bengio. Artificial neural networks applied to taxi destination prediction. In *Proceedings of the International Conference on ECML PKDD Discovery Challenge (ECMLPKDDDC)*, 2015.

[68] Deaf Support Voluntary Organisation. Level 3 British Sign Language 2017 handout resources, 2017.

[69] S. Deena, M. Hasan, M. Doulaty, O. Saz, and T. Hain. Combining feature and model-based adaptation of RNNLMs for multi-genre broadcast speech recognition. In *Proceedings of the Conference of the International Speech Communication Association (INTERSPEECH)*, pages 2343–2347, 2016.

[70] B. Demircioglu, G. Bülbül, and H. Kose. Recognition of sign language hand shape primitives with Leap motion. In *Proceedings of the Workshop on the Representation and Processing of Sign Languages*, 2016.

[71] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A large-scale hierarchical image database. In *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 248–255, 2009.

[72] L. Desblache. *How is Music Translated? Mapping the Landscape of Music Translation*, pages 219–264. Palgrave Macmillan UK, London, 2019.

[73] L. F. D'Haro, R. San-Segundo, R. d. Cordoba, J. Bungeroth, D. Stein, and H. Ney. Language model adaptation for a speech to sign language translation system using web frequencies and a map framework. In *Proceedings of the Conference of the International Speech Communication Association (INTERSPEECH)*, 2008.

[74] M. Dilsizian, P. Yanovich, S. Wang, C. Neidle, and D. N. Metaxas. A new framework for sign language recognition based on 3D handshape identification and linguistic modeling. In *Proceedings of the Workshop on the Representation and Processing of Sign Languages*, 2014.

[75] P. Dreuw, J. Forster, T. Deselaers, and H. Ney. Efficient approximations to model-based joint tracking and recognition of continuous sign language. In *Proceedings of the International Conference on Automatic Face and Gesture Recognition (FG)*, 2008.

[76] P. Dreuw, C. Neidle, V. Athitsos, S. Sclaroff, and H. Ney. Benchmark databases for video-based Automatic Sign Language recognition. In *Proceedings of the International Conference on Language Resources and Evaluation (LREC)*, 2008.

[77] P. Dreuw and H. Ney. Visual modeling and feature adaptation in sign language recognition. In *Proceedings of the Conference on Voice Communication*, pages 1–4, 2008.

[78] P. Dreuw, H. Ney, G. Martinez, O. A. Crasborn, J. Piater, J. Miguel Moya, and M. Wheatley. The signspeak project-bridging the gap between signers and speakers. In *Proceedings of the International Conference on Language Resources and Evaluation (LREC)*, 2010.

[79] P. Dreuw, D. Rybach, T. Deselaers, M. Zahedi, and H. Ney. Speech recognition techniques for a sign language recognition system. In *Proceedings of the Conference of the International Speech Communication Association (INTERSPEECH)*, volume 1, pages 2513–2516, 01 2007.

[80] R. Dubot and C. Collet. A hybrid formalism to parse sign languages. In *Proceedings of the International Conference on Language Resources and Evaluation (LREC)*, 2014.

[81] S. Ebling, N. C. Camgöz, P. B. Braem, K. Tissi, S. Sidler-Miserez, S. Stoll, S. Hadfield, T. Haug, R. Bowden, S. Tornay, et al. SMILE Swiss-German Sign Language dataset. In *Proceedings of the International Conference on Language Resources and Evaluation (LREC)*, 2018.

[82] S. Ebling, K. Tissi, and M. Volk. Semi-automatic annotation of semantic relations in a Swiss-German Sign Language lexicon. In *Proceedings of the Workshop on the Representation and Processing of Sign Languages*, 2012.

[83] M. Ebrahim Al-Ahdal and M. T. Nooritawati. Review in sign language recognition systems. In *Proceedings of the Symposium on Computers Informatics (ISCI)*, pages 52–57, 2012.

[84] E. Efthimiou, K. Vasilaki, S.-E. Fotinea, A. Vacalopoulou, T. Goulas, and A.-L. Dimou. The POLYTROPON parallel corpus. In *Proceedings of the International Conference on Language Resources and Evaluation (LREC)*, may 2018.

[85] A. Elons, M. Ahmed, H. Shedid, and M. Tolba. Arabic Sign Language recognition using Leap motion sensor. In *Proceedings of the International Conference on Computer Engineering & Systems (ICCES)*, pages 368–373, 2014.

[86] M. Fagiani, E. Principi, S. Squartini, and F. Piazza. A new Italian Sign Language database. In *Proceedings of the International Conference on Brain Inspired Cognitive Systems*, pages 164–173, 2012.

[87] M. Fagiani, E. Principi, S. Squartini, and F. Piazza. Signer independent isolated Italian sign recognition based on Hidden Markov Models. *Pattern Analysis and Applications*, 18(2):385–402, 2015.

[88] B. Fang, J. Co, and M. Zhang. DeepASL: Enabling ubiquitous and non-intrusive word and sentence-level sign language translation. In *Proceedings of the Conference on Embedded Network Sensor Systems*, pages 1–13, 2017.

[89] G. Fang, W. Gao, and D. Zhao. Large-vocabulary continuous sign language recognition based on transition-movement models. *IEEE Transactions on Systems, Man, and Cybernetics – Part A: Systems and Humans*, 37(1):1–9, 2007.

[90] R. Fatmi, S. Rashad, R. Integlia, and G. Hutchison. American Sign Language recognition using Hidden Markov Models and wearable motion sensors. *Machine Learning and Data Mining (MLDM)*, 10(2):41–55, 2017.

[91] J. Fenlon, K. Cormier, R. Rentelis, A. Schembri, K. Rowley, R. Adam, and B. Woll. BSL signbank: A lexical database and dictionary of british sign language, 2014. Deafness Cognition and Language Research Centre, University College London.

[92] J. Fenlon, A. Schembri, R. Rentelis, D. Vinson, and K. Cormier. Using conversational data to determine lexical frequency in British Sign Language: The influence of text type. *Lingua*, 143:187–202, 2014.

[93] D. Filimonov. *Decision tree-based syntactic language modeling*. PhD thesis, University of Maryland, 2011.

[94] J. Forster, C. Schmidt, T. Hoyoux, O. Koller, U. Zelle, J. H. Piater, and H. 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.

[95] J. Forster, C. Schmidt, O. Koller, M. Bellgardt, and H. 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)*, pages 1911–1916, May 2014.

[96] M. Fragkiadakis, V. Nyst, and P. van der Putten. Signing as input for a dictionary query: Matching signs based on joint positions of the dominant hand. In *Proceedings of the Workshop on the Representation and Processing of Sign Languages*, 2020.

[97] Y. Freund and R. E. Schapire. A desicion-theoretic generalization of on-line learning and an application to boosting. In *Proceedings of the European Conference on Computational Learning Theory*, pages 23–37. Springer, 1995.

[98] Gaolin Fang, Wen Gao, and Debin Zhao. Large vocabulary sign language recognition based on fuzzy decision trees. *IEEE Transactions on Systems, Man, and Cybernetics – Part A: Systems and Humans*, 34(3):305–314, May 2004.

[99] B. Garcia and S. A. Viesca. Real-time American Sign Language recognition with convolutional neural networks. *Machine Learning Research*, pages 225–232, 2016.

[100] S. Gattupalli, A. Ghaderi, and V. Athitsos. Evaluation of deep learning based pose estimation for sign language. In *Proceedings of the International Conference on Pervasive Technologies Related to Assistive Environments (PETRA)*, 2016.

[101] U. Gaur, Y. Zhu, B. Song, and A. Roy-Chowdhury. A "string of feature graphs" model for recognition of complex activities in natural videos. In *Proceedings of the International Conference on Computer Vision (ICCV)*, pages 2595–2602, 2011.

[102] D. M. Gavrila, L. S. Davis, et al. Towards 3-D model-based tracking and recognition of human movement: a multi-view approach. In *Proceedings of the International Conference on Automatic Face and Gesture Recognition (FG)*, 1995.

[103] S. Gibet, F. Lefebvre-Albaret, L. Hamon, R. Brun, and A. Turki. Interactive editing in french sign language dedicated to virtual signers: requirements and challenges. *Universal Access in the Information Society (UAIS)*, 15(4):525–539, 2016.

[104] A. Gineke, M. J. Reinders, E. A. Hendriks, H. de Ridder, and A. J. van Doorn. Influence of handshape information on automatic sign language recognition. In *International Gesture Workshop: Gesture in Embodied Communication and Human-Computer Interaction*, pages 301–312, 2009.

[105] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014.

[106] T. Glasmachers. Limits of end-to-end learning. In *Proceedings of the Asian Conference on Machine Learning*, 2017.

[107] J. Gu, H. Hassan, J. Devlin, and V. O. Li. Universal neural machine translation for extremely low resource languages. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics*, 2018.

[108] D. Guthrie, B. Allison, W. Liu, L. Guthrie, and Y. Wilks. A closer look at skip-gram modelling. In *Proceedings of the International Conference on Language Resources and Evaluation (LREC)*, 2006.

[109] J. F. Hair, W. C. Black, B. J. Babin, R. E. Anderson, R. L. Tatham, et al. *Multivariate data analysis*. Prentice Hall Upper Saddle River, NJ, 1998.

[110] J. Han, G. Awad, and A. Sutherland. Modelling and segmenting subunits for sign language recognition based on hand motion analysis. *Pattern Recognition Letters*, 30(6):623–633, 2009.

[111] M. Handouyahia, D. Ziou, and S. Wang. Sign language recognition using moment-based size functions. In *Proceedings of the International Conference on Vision Interface*, 1999.

[112] T. Hanke. HamNoSys-representing sign language data in language resources and language processing contexts. In *Proceedings of the Workshop on the Representation and Processing of Sign Languages*, 2004.

[113] T. Hanke, S. Matthes, A. Regen, and S. Worseck. Where does a sign start and end? segmentation of continuous signing. In *Proceedings of the Workshop on the Representation and Processing of Sign Languages*, 2012.

[114] K. He, X. Zhang, S. Ren, and J. Sun. Spatial pyramid pooling in deep convolutional networks for visual recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 37(9):1904–1916, 2015.

[115] M. D. Heath, S. Sarkar, T. Sanocki, and K. W. Bowyer. A robust visual method for assessing the relative performance of edge-detection algorithms. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 19(12):1338–1359, 1997.

[116] D. Held, D. Guillory, B. Rebsamen, S. Thrun, and S. Savarese. A probabilistic framework for real-time 3D segmentation using spatial, temporal, and semantic cues. In *Proceedings of the Robotics Science and Systems Conference (RSS)*, 2016.

[117] J. A. Hochgesang, O. Crasborn, and D. Lillo-Martin. Building the ASL Signbank: Lemmatization Principles for ASL. In *Proceedings of the International Conference on Language Resources and Evaluation (LREC)*, may 2018.

[118] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural Computation*, 9(8):1735–1780, 1997.

[119] S.-E. Hong, S. Won, I. Heo, and H. Lee. Development of an "integrative system for Korean Sign Language resources". In *Proceedings of the International Conference on Language Resources and Evaluation (LREC)*, may 2018.

[120] K. Irie, Z. Tuske, T. Alkhouli, R. Schluter, and H. Ney. LSTM, GRU, highway and a bit of attention: an empirical overview for language modeling in speech recognition. Technical report, RWTH Aachen University Aachen Germany, 2016.

[121] A. Isard. Approaches to the anonymisation of sign language corpora. In *Proceedings of the Workshop on the Representation and Processing of Sign Languages*, 2020.

[122] E. Jahn, R. Konrad, G. Langer, S. Wagner, and T. Hanke. Publishing DGS corpus data: Different formats for different needs. In *Proceedings of the International Conference on Language Resources and Evaluation (LREC)*, may 2018.

[123] T. Jantunen, B. Burger, D. De Weerdt, I. Seilola, and T. Wainio. Experiences from collecting motion capture data on continuous signing. In *Proceedings of the International Conference on Language Resources and Evaluation (LREC)*, 2012.

[124] T. Johnston. From archive to corpus: Transcription and annotation in the creation of signed language corpora. *International Journal of Corpus Linguistics (IJCL)*, 15(1):106–131, 2010.

[125] T. Johnston and A. Schembri. *Australian Sign Language (Auslan): An introduction to sign language linguistics*. Cambridge University Press, 2007.

[126] H. R. V. Joze and O. Koller. MS-ASL: A large-scale data set and benchmark for understanding American Sign Language. In *Proceedings of the British Machine Vision Conference*, 2019.

[127] H. Kacorri, A. Syed, M. Huenerfauth, and C. Neidle. Centroid-based exemplar selection of ASL non-manual expressions using multidimensional dynamic time warping and MPEG4 features. In *Proceedings of the Workshop on the Representation and Processing of Sign Languages*, 05 2016.

[128] T. Kadir, R. Bowden, E.-J. Ong, and A. Zisserman. Minimal training, large lexicon, unconstrained sign language recognition. In *Proceedings of the British Machine Vision Conference*, 2004.

[129] A. Kaivapalu and M. Martin. Morphology in Transition: Plural Inflection of Finnish nouns by Estonian and Russian Learners. *Acta Linguistica Hungarica*, 54(2):129–156, 2007.

[130] M. U. Kakde, M. G. Nakrani, and A. M. Rawate. A review paper on sign language recognition system for deaf and dumb people using image processing. *International Journal of Engineering Research & Technology (IJERT)*, 5(03), 2016.

[131] N. Kalchbrenner and P. Blunsom. Recurrent continuous translation models. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1700–1709, 2013.

[132] J. K. Karthick Arya. Convolutional neural networks based sign language recognition. *International Journal of Innovative Research in Computer and Communication Engineering*, 5(10), oct 2017.

[133] D. Kauchak. Improving text simplification language modeling using unsimplified text data. In *Proceedings of the Meeting of the Association for Computational Linguistics (ACL)*, pages 1537–1546, 08 2013.

[134] E. Keating, T. Edwards, and G. Mirus. Cybersign and new proximities: Impacts of new communication technologies on space and language. *Journal of Pragmatics*, 40(6):1067–1081, 2008.

[135] E. Keogh and C. A. Ratanamahatana. Exact indexing of dynamic time warping. *Knowledge and Information Systems (KAIS)*, 7(3):358–386, 2005.

[136] J. Keränen, H. Syrjälä, J. Salonen, and R. Takkinen. The usability of the annotation. In *Proceedings of the Workshop on the Representation and Processing of Sign Languages*, 2016.

[137] C. Keskin, F. Kıraç, Y. E. Kara, and L. Akarun. Real time hand pose estimation using depth sensors. In *Consumer depth cameras for computer vision*, pages 119–137. Springer, 2013.

[138] T. Kim, J. Keane, W. Wang, H. Tang, J. Riggle, G. Shakhnarovich, D. Brentari, and K. Livescu. Lexicon-free fingerspelling recognition from video: Data, models, and signer adaptation. *Computer Speech & Language*, 46:209–232, 2017.

[139] S.-K. Ko, C. J. Kim, H. Jung, and C. Cho. Neural sign language translation based on human keypoint estimation. *Applied Sciences*, 9(13):2683, 2019.

[140] T. Kohonen. Exploration of very large databases by self-organizing maps. In *Proceedings of International Conference on Neural Networks (ICNN)*, volume 1, pages PL1–PL6, 1997.

[141] O. Koller, C. Camgoz, H. Ney, and R. Bowden. Weakly supervised learning with multi-stream CNN-LSTM-HMMs to discover sequential parallelism in sign language videos. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, pages 1–1, 2019.

[142] O. Koller, H. Ney, and R. 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 Conference on Computer Vision and Pattern Recognition (CVPR), pages 3793–3802, June 2016.

[143] O. Koller, H. Ney, et al. Weakly supervised automatic transcription of mouthings for gloss-based sign language corpora. In *Proceedings of the International Conference on Language Resources and Evaluation (LREC)*, 2014.

[144] O. Koller, O. Zargaran, H. Ney, and R. Bowden. Deep sign: Hybrid CNN-HMM for continuous sign language recognition. In *Proceedings of the British Machine Vision Conference*, 2016.

[145] O. Koller, S. Zargaran, and H. Ney. Re-sign: Re-aligned end-to-end sequence modelling with deep recurrent CNN-HMMs. In *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4297–4305, 2017.

[146] R. Konrad. DGS corpus annotation guidelines. In *Proceedings of the Digging into Signs Workshop: Developing Annotation Standards for Sign Language Corpora*, 2015.

[147] O. Kostakis and P. Papapetrou. On searching and indexing sequences of temporal intervals. *Data Mining and Knowledge Discovery*, 31(3):809–850, 2017.

[148] O. K. Kostakis and A. G. Gionis. Subsequence search in event-interval sequences. In *Proceedings of the International Conference on Research and Development in Information Retrieval (SIGIR)*, pages 851–854, 2015.

[149] H. Kulhandjian, P. Sharma, M. Kulhandjian, and C. D'Amours. Sign language gesture recognition using doppler radar and deep learning. In *IEEE Globecom Workshops (GC Wkshps)*, 2019.

[150] P. Kumar, H. Gauba, P. P. Roy, and D. P. Dogra. Coupled HMM-based multi-sensor data fusion for sign language recognition. *Pattern Recognition Letters*, 86:1–8, 2017.

[151] T. Kuroda, Y. Tabata, A. Goto, H. Ikuta, M. Murakami, et al. Consumer price data-glove for sign language recognition. In *Proceedings of the International Conference on Disability, Virtual Reality and Associated Technologies (ICDVRAT)*, 2004.

[152] A. Kuznetsova, L. Leal-Taixe, and B. Rosenhahn. Real-time sign language recognition using a consumer depth camera. In *Proceedings of the International Conference on Computer Vision (ICCV)*, June 2013.

[153] H. M. Lakany and G. M. Hayes. An algorithm for recognising walkers. In *Proceedings of the International Conference on Audio-and Video-Based Biometric Person Authentication (AVBPA)*, pages 111–118, 1997.

[154] H.-S. Le, I. Oparin, A. Allauzen, J.-L. Gauvain, and F. Yvon. Structured output layer neural network language models for speech recognition. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 21(1):197–206, 2013.

[155] Y. LeCun, B. E. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. E. Hubbard, and L. D. Jackel. Handwritten digit recognition with a back-propagation network. In *Proceedings of the Advances in Neural Information Processing Systems (NeurIPS)*, pages 396–404, 1990.

[156] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, Nov 1998.

[157] K. Lee, D. Palsetia, R. Narayanan, M. M. A. Patwary, A. Agrawal, and A. Choudhary. Twitter trending topic classification. In *Proceedings of the International Conference on Data Mining Workshops (ICDMW)*, pages 251–258, 2011.

[158] J. Leskovec, A. Rajaraman, and J. D. Ullman. *Mining of massive data sets*. Cambridge university press, 2020.

[159] J. Lewis, M. Ackerman, and V. de Sa. Human cluster evaluation and formal quality measures: A comparative study. In *Proceedings of the Annual Meeting of the Cognitive Science Society (CogSci)*, 2012.

[160] D. Li, C. R. Opazo, X. Yu, and H. Li. Word-level deep sign language recognition from video: A new large-scale dataset and methods comparison. In *Proceedings of the Winter Conference on Applications of Computer Vision (WACV)*, 2019.

[161] A. Liaw, M. Wiener, et al. Classification and regression by random forest. *R News*, 2002.

[162] J. F. Lichtenauer, E. A. Hendriks, and M. J. Reinders. Sign language recognition by combining statistical DTW and independent classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 30(11):2040–2046, 2008.

[163] S. K. Liddell and R. E. Johnson. American Sign Language: The phonological base. *Sign Language Studies*, 64(1):195–277, 1989.

[164] K. M. Lim, A. W. C. Tan, C. P. Lee, and S. C. Tan. Isolated sign language recognition using convolutional neural network hand modelling and hand energy image. *Multimedia Tools and Applications*, 78(14):19917–19944, 2019.

[165] R. Lionnie, I. K. Timotius, and I. Setyawan. Performance comparison of several preprocessing methods in a hand gesture recognition system based on nearest neighbor for different background conditions. *Journal of ICT Research and Applications*, 6(3):183–194, 2012.

[166] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg. SSD: Single shot multibox detector. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 21–37. Springer, 2016.

[167] V. López-Ludeña, R. San-Segundo, C. G. Morcillo, J. C. López, and J. M. P. Muñoz. Increasing adaptability of a speech into sign language translation system. *Expert Systems with Applications*, 40(4):1312–1322, 2013.

[168] C. Lucas, R. Bayley, M. Rose, and A. Wulf. Location variation in American Sign Language. *Sign Language Studies*, pages 407–440, 2002.

[169] C. Lucas, G. Mirus, J. L. Palmer, N. J. Roessler, and A. Frost. The effect of new technologies on sign language research. *Sign Language Studies*, 13(4):541–564, 2013.

[170] M. Ma, M. Nirschl, F. Biadsy, and S. Kumar. Approaches for neural-network language model adaptation. In *Proceedings of the Conference of the International Speech Communication Association (INTERSPEECH)*, pages 259–263, 2017.

[171] B. MacCartney. NLP lunch tutorial: Smoothing, 2005. Stanford University.

[172] I. Marshall and É. Sáfár. A prototype text to British sign language (BSL) translation system. In *Proceedings of the Meeting of the Association for Computational Linguistics (ACL)*, 2003.

[173] I. Marshall and É. Sáfár. Sign language generation using hpsg. In *Proceedings of the International Conference on Head-Driven Phrase Structure Grammar*, 2004.

[174] G. Massó and T. Badia. Dealing with sign language morphemes in statistical machine translation. In *Proceedings of the Workshop on the Representation and Processing of Sign Languages*, 2010.

[175] S. Merity, N. S. Keskar, and R. Socher. An Analysis of Neural Language Modeling at Multiple Scales. *pre-print*, 2018. https://arxiv.org/abs/1803.08240.

[176] S. Merity, N. S. Keskar, and R. Socher. Regularizing and Optimizing LSTM Language Models. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2018.

[177] J. Mesch and K. Schönström. From design and collection to annotation of a learner corpus of sign language. In *Proceedings of the International Conference on Language Resources and Evaluation (LREC)*, may 2018.

[178] J. Mesch and L. Wallin. Gloss annotations in the Swedish Sign Language corpus. *International Journal of Corpus Linguistics (IJCL)*, 20, 01 2015.

[179] L. Meurant, A. Cleve, and O. Crasborn. Using sign language corpora as bilingual corpora for data mining. contrastive linguistics and computer-assisted annotation. In *Proceedings of the Workshop on the Representation and Processing of Sign Languages*, 05 2016.

[180] T. Mikolov, M. Karafiát, L. Burget, J. Černockỳ, and S. Khudanpur. Recurrent neural network based language model. In *Proceedings of the Conference of the International Speech Communication Association (INTERSPEECH)*, 2010.

[181] T. Mikolov, J. Kopecky, L. Burget, O. Glembek, and J. ?Cernocky. Neural network based language models for highly inflective languages. In *Proceedings of the International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4725–4728, April 2009.

[182] G. R. Mirus. *The linguistic repertoire of deaf cuers: an ethnographic query on practice*. PhD thesis, University of Texas, 2008.

[183] B. Mocialov, G. Turner, and H. Hastie. Towards large-scale data mining for data-driven analysis of sign languages. In *Proceedings of the Workshop on the Representation and Processing of Sign Languages*, 2020.

[184] B. Mocialov, G. Turner, K. Lohan, and H. Hastie. Towards continuous sign language recognition with deep learning. In *Proceedings of the Workshop on the Creating Meaning With Robot Assistants: The Gap Left by Smart Devices*, 2017.

[185] B. Mocialov, P. A. Vargas, and M. S. Couceiro. Towards the evolution of indirect communication for social robots. In *Proceedings of the Symposium Series on Computational Intelligence (SSCI)*, pages 1–8, Dec 2016.

[186] D. C. Montgomery, E. A. Peck, and G. G. Vining. *Introduction to linear regression analysis*, volume 821. John Wiley & Sons, 2012.

[187] D. G. R. Muskan Dhiman. Sign language recognition. Technical report, National Institute of Technology, Hamirpur (H.P.), 2017.

[188] L. Naert, C. Reverdy, C. Larboulette, and S. Gibet. Per channel automatic annotation of sign language motion capture data. In *Proceedings of the International Conference on Language Resources and Evaluation (LREC)*, may 2018.

[189] S. Nayak, K. Duncan, S. Sarkar, and B. Loeding. Finding recurrent patterns from continuous sign language sentences for automated extraction of signs. *Machine Learning Research*, 13(Sep):2589–2615, 2012.

[190] C. Neidle, A. Thangali, and S. Sclaroff. Challenges in development of the American Sign Language lexicon video dataset (ASLLVD) corpus. In *Proceedings of the Workshop on the Representation and Processing of Sign Languages*, 2012.

[191] C. Neidle and C. Vogler. A new web interface to facilitate access to corpora: Development of the ASLLRP data access interface (DAI). In *Proceedings of the Workshop on the Representation and Processing of Sign Languages*, 2012.

[192] P. Nerurkar, A. Shirke, M. Chandane, and S. Bhirud. Empirical analysis of data clustering algorithms. *Procedia Computer Science*, 125:770–779, 2018.

[193] S. Ojala, T. Salakoski, and O. Aaltonen. Coarticulation in sign and speech. In *Proceedings of the Workshop on Multimodal Communication*, page 21, 2009.

[194] E.-J. Ong, O. Koller, N. Pugeault, and R. Bowden. Sign spotting using hierarchical sequential patterns with temporal intervals. In *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1923–1930, 2014.

[195] S. C. Ong, S. Ranganath, et al. Automatic sign language analysis: A survey and the future beyond lexical meaning. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 27(6):873–891, 2005.

[196] E. Orfanidou, R. Adam, J. M. McQueen, and G. Morgan. Making sense of nonsense in british sign language (BSL): The contribution of different phonological parameters to sign recognition. *Memory & cognition*, 37(3):302–315, 2009.

[197] R. Östling, C. Börstell, and S. Courtaux. Visual iconicity across sign languages: Large-scale automated video analysis of iconic articulators and locations. *Frontiers in Psychology*, 9:725, 2018.

[198] M. Oszust and M. Wysocki. Determining subunits for sign language recognition by evolutionary cluster-based segmentation of time series. In *Proceedings of the International Conference on Artificial Intelligence and Soft Computing (ICAISC)*, pages 189–196, 2010.

[199] M. Oszust and M. Wysocki. Polish Sign Language words recognition with Kinect. In *Proceedings of the International Conference on Human System Interactions (HSI)*, pages 219–226, 2013.

[200] A. Othman and M. Jemni. A probabilistic model for sign language translation memory. In A. Abraham and S. M. Thampi, editors, *Proceedings of the International Symposium on Intelligent Informatics*, pages 317–324, 2012.

[201] A. Oviedo and C. Ramírez Valerio. The LESCO corpus. data for the description of Costa Rican Sign Language. In *Proceedings of the International Conference on Language Resources and Evaluation (LREC)*, may 2018.

[202] S. J. Pan and Q. Yang. A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering (TKDE)*, 22(10):1345–1359, 2009.

[203] K. Papadimitriou and G. Potamianos. End-to-end convolutional sequence learning for ASL fingerspelling recognition. In *Proceedings of the Conference of the International Speech Communication Association (INTERSPEECH)*, pages 2315–2319, 2019.

[204] P. Papapetrou, G. Kollios, S. Sclaroff, and D. Gunopulos. Mining frequent arrangements of temporal intervals. *Knowledge and Information Systems (KAIS)*, 21(2):133, 2009.

[205] Y. Park, S. Patwardhan, K. Visweswariah, and S. C. Gates. An empirical analysis of word error rate and keyword error rate. In *Proceedings of the Conference of the International Speech Communication Association (INTERSPEECH)*, 2008.

[206] M. P. Paulraj, S. Yaacob, M. S. bin Zanar Azalan, and R. Palaniappan. A phoneme based sign language recognition system using skin color segmentation. In *Proceedings of the International Colloquium on Signal Processing & its Applications (CSPA)*, pages 1–5, 2010.

[207] N. Pugeault and R. Bowden. Spelling it out: Real-time ASL fingerspelling recognition. In *Workshop on Consumers Depth Cameras for Computer Vision*, 2011.

[208] L. Rabiner and B. Juang. An introduction to Hidden Markov Models. *IEEE ASSP Magazine*, 3(1):4–16, 1986.

[209] C. Rajah. *Chereme-based recognition of isolated, dynamic gestures from South African Sign Language with Hidden Markov Models*. PhD thesis, University of the Western Cape, 2006.

[210] N. Rajalingam and K. Ranjini. Hierarchical clustering algorithm-a comparative study. *International Journal of Computer Applications (IJCA)*, 19(3):42–46, 2011.

[211] P. J. Roach. Report on the 1989 kiel convention: International phonetic association. *Journal of the International Phonetic Association*, 19(2):67–80, 1989.

[212] J. Romero, H. Kjellström, and D. Kragic. Hands in action: real-time 3D reconstruction of hands in interaction with objects. In *Proceedings of the International Conference on Robotics and Automation (ICRA)*, pages 458–463, 2010.

[213] R. Rosenfeld. Two decades of statistical language modeling: Where do we go from here? *Proceedings of the IEEE*, 88(8):1270–1278, 2000.

[214] P. J. Rousseeuw. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics*, 20:53–65, 1987.

[215] A. Roussos, S. Theodorakis, V. Pitsikalis, and P. Maragos. Hand tracking and affine shape-appearance handshape sub-units in continuous sign language recognition. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 258–272, 2010.

[216] S. S Kumar, T. Wangyal, V. Saboo, and R. Srinath. Time series neural networks for real time sign language translation. In *Proceedings of the International Conference on Machine Learning and Applications (ICMLA)*, pages 243–248, 2018.

[217] R. San-Segundo, J. M. Pardo, J. Ferreiros, V. Sama, R. Barra-Chicote, J. M. Lucas, D. Sánchez, and A. García. Spoken Spanish generation from sign language. *Interacting with Computers*, 22(2):123–139, 2009.

[218] R. San Segundo Hernández, V. Lopez Ludeña, R. Martin Maganto, D. Sánchez, and A. García. Language resources for Spanish-Spanish Sign Language (LSE) translation. In *Proceedings of the International Conference on Language Resources and Evaluation (LREC)*, 2010.

[219] A. C. Schembri, J. Fenlon, R. Rentelis, S. A. Reynolds, and K. Cormier. Building the British Sign Language corpus. *Language Documentation and Conservation (LD&C)*, 2013.

[220] B. Schölkopf, A. J. Smola, F. Bach, et al. *Learning with kernels: support vector machines, regularization, optimization, and beyond.* MIT Press, 2002.

[221] E. Schubert, J. Sander, M. Ester, H. P. Kriegel, and X. Xu. Dbscan revisited, revisited: why and how you should (still) use dbscan. *ACM Transactions on Database Systems (TODS)*, 42(3):1–21, 2017.

[222] S. K. Sedeeq Al-khazraji and M. Huenerfauth. Modeling and predicting the location of pauses for the generation of animations of American Sign Language. In *Proceedings of the International Conference on Language Resources and Evaluation (LREC)*, may 2018.

[223] J. Segouat. A study of sign language coarticulation. *ACM SIGACCESS Accessibility and Computing*, pages 31–38, 2009.

[224] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. LeCun. Overfeat: Integrated recognition, localization and detection using convolutional networks. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2014.

[225] B. Shi, A. M. D. Rio, J. Keane, D. Brentari, G. Shakhnarovich, and K. Livescu. Fingerspelling recognition in the wild with iterative visual attention. In *Proceedings of the International Conference on Computer Vision (ICCV)*, pages 5400–5409, 2019.

[226] C. Shorten and T. M. Khoshgoftaar. A survey on Image Data Augmentation for Deep Learning. *Journal of Big Data*, 2019.

[227] T. Simon, H. Joo, I. Matthews, and Y. Sheikh. Hand keypoint detection in single images using multiview bootstrapping. In *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1145–1153, 2017.

[228] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2015.

[229] C. Solomon and T. Breckon. *Fundamentals of Digital Image Processing: A practical approach with examples in Matlab*. John Wiley & Sons, 2011.

[230] T. Starner and A. Pentland. Real-time American Sign Language recognition from video using Hidden Markov Models. In *Motion-based recognition*, pages 227–243. Springer, 1997.

[231] D. Stein. Morpho-syntax based statistical methods for sign language translation. In *Proceedings of the Conference of the European Association for Machine Translation (EAMT)*, 2006.

[232] D. Stein, P. Dreuw, H. Ney, S. Morrissey, and A. Way. Hand in hand: automatic sign language to English translation. In *Proceedings of the Conference on Theoretical and Methodological Issues in Machine Translation (TMI)*, 2007.

[233] D. Stein, P. Dreuw, H. Ney, S. Morrissey, and A. Way. Hand in hand: Automatic sign language to English translation. In *Proceedings of the Conference on Theoretical and Methodological Issues in Machine Translation (TMI)*, 04 2012.

[234] D. Stein, J. Forster, U. Zelle, P. Dreuw, and H. Ney. Analysis of the german sign language weather forecast corpus. In *Proceedings of the Workshop on the Representation and Processing of Sign Languages*, 2010.

[235] W. Stokoe, D. Casterline, and C. Croneberg. *A Dictionary of American Sign Language on Linguistic Principles*. Linstok Press, 1976.

[236] W. C. Stokoe Jr. Sign language structure: An outline of the visual communication systems of the American deaf. *Journal of Deaf Studies and Deaf Education*, 10(1):3–37, 2005.

[237] A. Stolcke. SRILM-an extensible language modeling toolkit. In *Proceedings of the Seventh international conference on spoken language processing*, 2002.

[238] I. Sutskever, O. Vinyals, and Q. V. Le. Sequence to sequence learning with neural networks. In *Proceedings of the Advances in Neural Information Processing Systems (NeurIPS)*, 2014.

[239] V. Sutton. SignWriting for sign languages, 1974. SignWriting.

[240] R. Sutton-Spence and B. Woll. *The linguistics of British Sign Language: an introduction*. Cambridge University Press, 1999.

[241] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[242] The World Association of Sign Language Interpreters (WASLI). WFD and WASLI statement on use of signing avatars, 2018.

[243] D. Tome, C. Russell, and L. Agapito. Lifting from the deep: Convolutional 3D pose estimation from a single image. In *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[244] S. Tornay, M. Razavi, N. C. Camgoz, R. Bowden, and M. M. Doss. HMM-based approaches to model multichannel information in sign language inspired from articulatory features-based speech processing. In *Proceedings of the International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2817–2821, 2019.

[245] T. Troelsgård and J. H. Kristoffersen. An electronic dictionary of Danish Sign Language. In *Proceedings of the International Conference on Language Resources and Evaluation (LREC)*, may 2018.

[246] S. Tsumoto and S. Hirano. Detection of risk factors using trajectory mining. *Journal of Intelligent Information Systems (JIIS)*, 36(3):403–425, 2011.

[247] A. Utsumi and J. Ohya. Multiple-hand-gesture tracking using multiple cameras. In *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 1, pages 473–478, 1999.

[248] V. Vapnik. The support vector method of function estimation. In *Nonlinear Modeling*, pages 55–85. Springer, 1998.

[249] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin. Attention is all you need. In *Proceedings of the Advances in Neural Information Processing Systems (NeurIPS)*, 2017.

[250] P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. In *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 1, pages I–I, 2001.

[251] M. Vlachos, M. Hadjieleftheriou, D. Gunopulos, and E. Keogh. Indexing multi-dimensional time-series with support for multiple distance measures. In *Proceedings of the International Conference on Knowledge Discovery and Data Mining (KDD)*, page 216–225, 2003.

[252] C. Vogler and D. Metaxas. Asl recognition based on a coupling between HMMs and 3D motion analysis. In *Proceedings of the International Conference on Computer Vision (ICCV)*, page 363, 1998.

[253] U. Von Agris, M. Knorr, and K.-F. Kraiss. The significance of facial features for automatic sign language recognition. In *Proceedings of the International Conference on Automatic Face and Gesture Recognition (FG)*, pages 1–6, 2008.

[254] U. Von Agris and K.-F. Kraiss. Towards a video corpus for signer-independent continuous sign language recognition. In *In Proceedings of the International Workshop on Gesture in Human-Computer Interaction and Simulation (GW)*, 2007.

[255] J. Wang and L. Perez. The Effectiveness of Data Augmentation in Image Classification using Deep Learning. Technical report, Stanford University, 2017.

[256] T. Wang, Y. Li, J. Hu, A. Khan, L. Liu, C. Li, A. Hashmi, and M. Ran. A survey on vision-based hand gesture recognition. In *Proceedings of the International Conference on Smart Multimedia*, pages 219–231, 2018.

[257] S.-E. Wei, V. Ramakrishna, T. Kanade, and Y. Sheikh. Convolutional pose machines. In *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[258] E. W. D. Whittaker. *Statistical language modelling for automatic speech recognition of Russian and English*. PhD thesis, University of Cambridge, 2000.

[259] I. H. Witten and E. Frank. Data mining: practical machine learning tools and techniques with Java implementations. *ACM SIGMOD Record*, 31(1):76–77, 2002.

[260] C.-H. Wu, Y.-H. Chiu, and C.-S. Guo. Text generation from Taiwanese Sign Language using a PST-based language model for augmentative communication. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 12(4):441–454, 2004.

[261] Y. Xue, Z. Ju, K. Xiang, J. Chen, and H. Liu. Multimodal human hand motion sensing and analysis—a review. *IEEE Transactions on Cognitive and Developmental Systems (TCDS)*, 11(2):162–175, 2018.

[262] H.-D. Yang. Sign language recognition with the Kinect sensor based on conditional random fields. *Sensors*, 15(1):135–147, 2014.

[263] R. Yang, S. Sarkar, and B. Loeding. Handling movement epenthesis and hand segmentation ambiguities in continuous sign language recognition using nested dynamic programming. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 32(3):462–477, 2010.

[264] S.-H. Yang and J.-Z. Gan. An interactive Taiwan Sign Language learning system based on depth and color images. In *Proceedings of the International Conference on Consumer Electronics-Taiwan (ICCE-TW)*, pages 112–113, 2015.

[265] Y. Yasugahira, Y. Horiuchi, and S. Kuroiwa. Analysis of hand movement variation related to speed in Japanese Sign Language. In *Proceedings of the International Universal Communication Symposium (IUCS)*, pages 331–334, 2009.

[266] F. Yin, X. Chai, and X. Chen. Iterative reference driven metric learning for signer independent isolated sign language recognition. In B. Leibe, J. Matas, N. Sebe, and M. Welling, editors, *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 434–450, 2016.

[267] F. Yin, X. Chai, Y. Zhou, and X. Chen. Semantics constrained dictionary learning for signer-independent sign language recognition. In *Proceedings of the International Conference on Image Processing (ICIP)*, pages 3310–3314, 09 2015.

[268] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson. How transferable are features in deep neural networks? In *Proceedings of the Advances in Neural Information Processing Systems (NeurIPS)*, pages 3320–3328, 2014.

[269] M. M. Zaki and S. I. Shaheen. Sign language recognition using a combination of new vision based features. *Pattern Recognition Letters*, 32(4):572–577, 2011.

[270] U. Zeshan. Sign language of the world. In *Encyclopedia of language and linguistics (vol. 11)*, pages 358–365. Elsevier, 2006.

[271] L. Zhao, K. Kipper, W. Schuler, C. Vogler, N. Badler, and M. Palmer. A machine translation system from english to american sign language. In *Proceedings of the Conference of the Association for Machine Translation in the Americas*, 2000.

[272] A. Zheng and A. Casari. *Feature engineering for machine learning: principles and techniques for data scientists*. "O'Reilly Media, Inc.", 2018.

Appendix A

Towards the Evolution of Indirect Communication for Social Robots

Boris Mocialov
Robotics Lab at
School of Mathematical and Computer Sciences and
School of Engineering & Physical Sciences at
Heriot-Watt University and
Edinburgh Centre for Robotics
Edinburgh, UK
Email: bm4@hw.ac.uk

Patricia A. Vargas
Robotics Lab at
School of Mathematical and Computer Sciences at
Heriot-Watt University and
Edinburgh Centre for Robotics
Edinburgh, UK
Email: P.A.Vargas@hw.ac.uk

Micael S. Couceiro
Ingeniarius Ltd.
and University of Coimbra
Mealhada, Portugal
Email: micael@ingeniarius.pt

***Abstract*—This paper presents preliminary investigations on the evolution of indirect communication between two agents. In the future, behaviours of robots in the RoboCup[1] competition should resemble the behaviours of the human players. One common trait of this behaviour is the indirect communication. Within the human–robot–interaction, indirect communication can either be the principal or supporting method for information exchange. This paper summarises previous work on the topic and presents the design of a self–organised system for gesture recognition. Although, preliminary results show that the proposed system requires further feature extraction improvements and evaluations on various public datasets, the system is capable of performing classification of gestures. Further research is required to fully investigate potential extensions to the system that would be able to support real indirect communication in human-robot interaction scenarios.**

I. Introduction

Tasks that include human subjects, such as surveillance, medical diagnosis, human–machine interaction, and sport analysis, require action and behaviour awareness [1]. For example, in sports, collective behaviour is emerged from individual behaviours within the team. These behaviours include which part of the team is attacking and who is defending [2]. There are only few attempts to modelling perception of the game [3], [4], whereas majority of the research apply pattern recognition to understand human movement in sports [5]. Due to the complexity of rules and concepts in the sports context, most of the robotic football policies in the RoboCup competition [6] are either hard–coded, present simplistic behavioural frameworks that do not represent behaviours of the real football players, or require extensive calibrations prior execution and still lack the autonomy while exhibiting very restricted human-like behaviour [4], [7], [8]. This paper presents a system that is the first step to bridging the gap between robotic football and human football. Long–term aim of this study is to pursue the goal of developing an autonomous team of robots that will defeat human players [6]. The initial step in this direction is to copy the way players understand actions and (re-)act accordingly.

Reliability of the communication via direct communication devices, present on the boards of autonomous robotic agents, can be compromised by a range of internal or external factors. While communication link failure between two agents results in temporal communication impairment that can be recovered by communication managing software, physical damage of communication devices usually leads to permanent communication loss [9].

The results of this study led to development of gesture recognition system using evolutionary approach [10], [11]. The system is intended to be uploaded to a robot and updated in real time as the robot learns new gestures from a coach or a teacher. The system had been tested on a PC using standard web camera, first with a human subject, second, with a publicly available reduced ChaLearn dataset[2], and third, with a NAO torso, performing hitting and hugging gestures. PC-based implementation had been used at this stage for convenient testing of key functionality on video data instead of intricate application directly on board of a robotic platform. However, the functionality is believed to be platform–independent. Ultimately, the system is expected to be used on a robotic platform with a standard camera, which will pose additional challenges for the system, such as change of orientation, varying illumination, motion blur, etc. This work represents the first steps towards the creation of a truly self-organised system that applies evolution to facilitate indirect communication between agents.

This paper is organised as follows. Section II reviews the related literature on gesture recognition and feature extraction, Section III describes the developed system, its layers and their functionality, Section IV describes the backbone of the system, its sub-components and their interactions, Section V presents conducted experiments' setup and preparations, while Section VI shows the results obtained from the experiments, Section VII discusses the results obtained and their implications; the project is summarised in Section VIII and the future work is proposed.

[1] RoboCup competition http://www.robocup.org/

[2] ChaLearn Gesture Dataset (CGD 2011), ChaLearn, California, 2011 http://gesture.chalearn.org/data/cgd2011

978-1-5090-4240-1/16/$31.00 ©2016 IEEE

II. Related Work

In the field of human–computer interaction, two main methods are used for data collection in interaction through indirect communication. These are identified as glove–based and vision–based methods [12]. Previously, LaViola distinguished another hybrid approach that used sensor fusion of the two approaches [13]. On one hand, glove–based devices for interaction data collection generate coherent data, but make the interaction experience cumbersome for the user. On the other hand, vision–based approaches free the user, but tend to introduce additional challenges for the recognition and classification tasks. These challenges, among others, include the variation in light, camera movements, and lack of depth awareness that impacts robustness of the interaction recognition algorithms.

Any gesture recognition system should include (i) data acquisition and pre-processing, (ii) data representation and feature extraction, and (iii) classification or decision-making. These steps form a vision–based framework for the RoboCup scenario in [14]. The important distinction should be made between static and dynamic gestures in the early modelling stages as approaches for feature extraction differ as dynamic gesture recognition requires additional segmentation and tracking modules.

Most distinguished approaches to action representation include Hidden Markov Models as it is done in [15] or straightforward sequence of frames chaining [16]. These approaches consider sequences of frames as action modelling cannot be done without temporal information. Another technique, 'String of feature graphs' (SFGs) [17] represents every frame as a graph of kinematic features. This technique encodes an action by combining the sequence of graphs from every frame. As a result, a sequence of features graphs represents spatio-temporal features of an action.

Different studies on representation and recognition of gestures explored the use of different features. Used features can be classified as either global or local, where local features are more specific and detailed and global features are general and noisy. Global features can be represented as Cartesian distances between centroids of blobs that represent hands on every frame [18]. Less specific classification is done with the clouds of interest points that can represent either shape, speed, density, or all together [19]. Classification of body parts may not be necessary as it is showed by representing gestures by any arbitrary change that happens between the subsequent frames [20]. Local features, such as kinematic points, are less noisy than mentioned above global features and require less post-extraction processing and data cleaning as opposed to global features, for which the amount of noise is proportionate to the amount of data collected [21]. Local features extraction process, nevertheless, requires more precise algorithms.

III. System Design

SFGs approach is used for gesture representation [17]. Feature Graphs (FGs) capture information about kinematic features. Graph data structure allows dynamic addition of new nodes. This serves as an advantage in the gesture representation context as it is not known in advance which gesture is being represented.

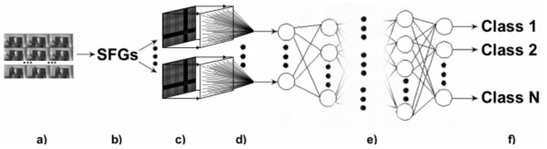


Fig. 1. Overall view of the system
a) extracted features from sequence of frames in a video stream b)) SFG for the video stream c) affinity matrix for the SFG d) feature detectors evolved neural networks e) classifier artificial neural network f) resulting classification of the video stream in a)

A. System Design: Feature Extraction

Feature extraction corresponds to layer a) from Figure 1. The implementation uses OpenCV[3] library due to its useful matrix processing functionality.

Prior to region of interest (ROI) detection, every frame is pre-processed by performing 1) Background-foreground subtraction 2) Illumination reduction and 3) Foreground edges enhancement.

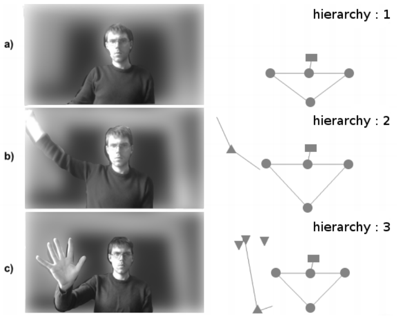


Fig. 2. ROI Detection (Segmentation)
a) Hierarchy 1 (body frame) b) Hierarchy 2 (body frame and limbs) c) Hierarchy 3 (body frame, limbs, and limb details)

1) ROI Detection: ROI consists of the following regions: face, upper body and a moving object. Face and upper body are detected using standard OpenCV Haar feature-based cascade classifier, while movement detection is the result of background subtraction and comparison of the consecutive frames' foregrounds. The movement is classified as a part of the overall body only if it originates from the upper body.

Potential limbs are analysed by looking at hull and convex defects to find break points (elbows) and smaller details (e.g. fingers). The detection is hierarchical and is performed in sequence (e.g. fingers will not be considered until this detail is needed for classification of signs in sign language and the arm has been detected).

[3] OpenCV library http://opencv.org/

2) Feature Extraction: Extracted features are joint positions in space. Following features had been chosen to represent a gesture:

- face and hand with face-hand distance
- first and second hand with hand-hand distance
- first and second shoulder with shoulder-shoulder distance
- first and second elbow with elbow-elbow distance

B. System Design: Feature Encoding

In layer b) from Figure 1, extracted features are encoded as nodes in 2D space and their relations are the Euclidean distances between the nodes encoded as edges in a undirected FG. To describe a video, all FGs are concatenated into a list to make up one SFG.

C. Affinity Matrix Calculation

$$M(a,a) = \begin{cases} \tau_1 \text{ - } d(\imath_1,\, \imath_2) & \text{if } d(\imath_1,\, \imath_2) \leq\tau_1 \\ 0 & \text{otherwise} \end{cases}$$

$$M(a,b) = \begin{cases} \tau_2 \text{ - } d(\imath_1\jmath_1,\, \imath_2\jmath_2) & \text{if } d(\imath_1\jmath_1,\, \imath_2\jmath_2) \leq\tau_2 \\ 0 & \text{otherwise} \end{cases}$$

Fig. 3. Affinity matrix definition

,where

M	: affinity matrix
a, b	: matrix indices
τ_1, τ_2	: threshold values, where τ_1 is the maximum allowed Euclidean distance between two nodes and τ_2 is the maximum allowed deviation between edges inclinations
$\imath_1, \jmath_1, \imath_2, \jmath_2$	: nodes of SFGs or edge between nodes $(\imath_1, \jmath_1)$ and $(\imath_2, \jmath_2)$
$d(\imath_1, \imath_2)$	: distance between two nodes that belong to different FGs
$d(\imath_1\jmath_1, \imath_2\jmath_2)$	: inclination between edges that belong to different FGs

In layer c) from Figure 1, SFGs are transformed into affinity matrices that hold similarity information between all frames in a single matrix. Figure 3 formally describes that the diagonal holds information about similarity between nodes, while the rest of the matrix represents similarity between edges [17].

D. System Design: Detectors

This implementation, as shown in layer d) from Figure 1, uses artificial neural networks as an alternative to spectral clustering, performed on the resulting affinity matrices as it is done in [17]. By using neural networks, the system is able to classify gestures directly from video stream without the need to compare every learned template gesture to the stream.

Kocmánek in [22] presents a method for handwritten digit recognition with HyperNEAT [23] algorithm. The algorithm evolves novel detectors that extract unique features from images. Similar approach is used in this paper with only difference in that the system is operating on the spatio-temporal data, encoded as affinity matrices.

Neural network processing (hnn[4]) package together with Python-based implementation of the HyperNEAT algorithm (peas[5]), developed in [24], are used to evolve distinct detectors for SFG gesture representations.

For all experiments 50 detectors with 100 inputs, no hidden layers, and a single output are evolved using novelty search technique. This leads to different detectors focusing on different sections of the affinity matrices.

In this implementation, the HyperNEAT algorithm is restricted to produce detectors of certain topology as described in [22]. For more complex detectors, future evolutions of detectors could be more elaborate, evolving the size and the activation functions of the detectors.

TABLE I
HYPERNEAT PARAMETERS FOR DETECTORS' EVOLUTION

Substrate	Inputs 10×10 Outputs 0×1
Generations	depending on the experiment
Population	50
Inputs per individual	100
Outputs per individual	1
Maximum depth	3
Weights range	(-3.0, 3.0)
P(new connection)	0.3
P(new node)	0.1
P(weight mutation)	0.8
P(weight reset)	0.1
P(disable connection)	0.01
P(re-enable connection)	0.01
Node types range	tanh
Evaluation function	argmax($\sum_1^k$Manhattan(k-NN($output_{detector}$)))
Minimum allowed fitness	0.05

Every detector is evolved by a separate instance of peas algorithm using parameters, given in Table I.

Substrate consists of two fully connected layers. Input layer has at most 10×10 nodes and output layer has at most 0×1 nodes. 'P' is the probability of adding new connections, adding new nodes, etc. Evaluation function objective is to maximise the Manhattan distance between all the detectors. The aim is to evolve novel detectors.

In k-NN, the k-nearest neighbour, $k = 50$ (all other detectors are considered). The problem of maximising the distance between the evolved detectors is reduced to finding the maximum Manhattan distance between a set of arrays.

A single vector is associated with every detector with as many items as there are gestures to be learned by the system. The vector is used to accumulate activations of the output neuron for every gesture. The vector describes how many times the detector detected something in affinity matrix.

E. System Design: Classifier

As can be seen in layer e) from Figure 1, the classifier has same amount of inputs as there are detectors in the system, with every detector feeding its output into the classifier's dedicated input. In this setup, the classifier has 50 inputs, 2 hidden layers with 300 neurons in each and certain amount of outputs, depending on the experiment, with every output

[4] A reasonably fast and simple neural network library https://hackage.haskell.org/package/hnn

[5] Python Evolutionary Algorithms https://github.com/noio/peas/

representing a probability of the gesture class, associated with that output.

The library[6] uses resilient backpropagation (Rprop) [25] as network training method.

IV. Sub-Systems Interaction

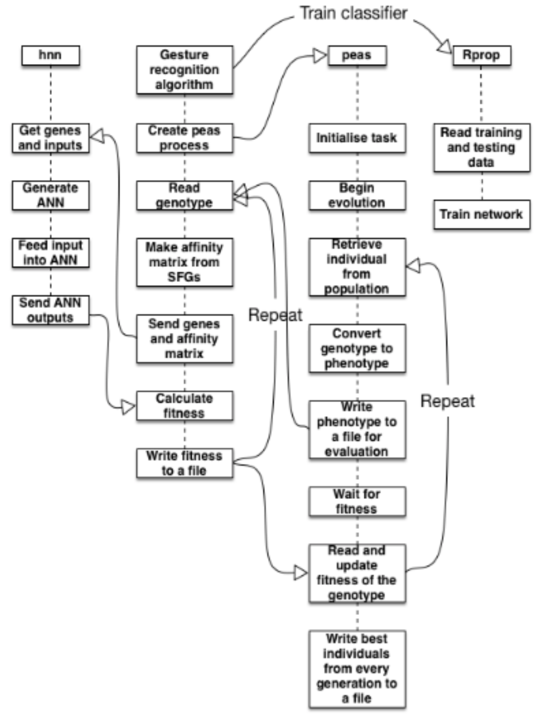

Train classifier
hnn
Gesture recognition algorithm
peas
Rprop
Get genes and inputs
Create peas process
Initialise task
Read training and testing data
Generate ANN
Read genotype
Begin evolution
Train network
Feed input into ANN
Make affinity matrix from SFGs
Retrieve individual from population
Send ANN outputs
Repeat
Send genes and affinity matrix
Convert genotype to phenotype
Calculate fitness
Write phenotype to a file for evaluation
Repeat
Write fitness to a file
Wait for fitness
Read and update fitness of the genotype
Write best individuals from every generation to a file

Fig. 4. Sub-Systems Execution and Communication Sequence Diagram
a) hnn (Haskell neural network library) b) C/C++ gesture recognition algorithm c) peas (Python HyperNEAT algorithm) d) Resilient Backpropagation implemented in Matlab

The system consists of 4 parts and is presented in Figure 4. Data exchange between peas and gesture recognition algorithms is done through the file system.

Gesture recognition algorithm launches the HyperNEAT [23] instances to begin the evolution of detectors. Once the instance is launched, the gesture recognition algorithm waits for generated neural networks (genotypes) from the instance. When genotype is generated, it is written to a file and the HyperNEAT is paused until the evaluation results are written to another file. Both files are used to exchange data between the two algorithms. When the genotype is received, gesture recognition algorithm evaluates it on training data, calculates the fitness, writes the fitness to the file, and pauses until the next genotype becomes available. At this time, HyperNEAT algorithm continues, reads the fitness of the genotype, and writes it for further evaluations of the population. When a genotype is available and affinity matrix is ready to be evaluated, hnn is invoked.

When the evolution of detectors has finished, the gesture recognition algorithm launches Rprop algorithm that trains classifier neural network using detectors' outputs as inputs into the classifier.

V. Experiment Setup

The system has been tested on three gesture datasets. First, single subject, self-made 4 different gestures (left hand wave, right hand wave, both hands wave simultaneously, and no hands waving). Second experiment used single subject's 10 more complicated signaling gestures from ChaLearn dataset. Third, a self-made NAO gesturing dataset was created with 2 gestures (raise one arm up as trying to hit and spread both arms apart as trying to hug).

Before the training of the model, the raw video data for all datasets were transformed into affinity matrices.

A. Experiment Preparation with Self-Made Gestures Dataset

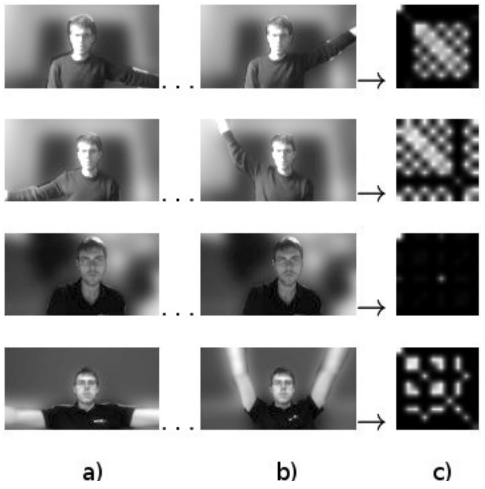


Fig. 5. Affinity matrices generation from gestures in video data from self-made gestures dataset (single gesture approx. 1-2 seconds)
a) first frame b) last frame c) generated affinity matrix

Figure 5 shows that extracted kinematic features from videos are being used to generate affinity matrices, which are different for every gesture. Showed results correspond to steps A, B, and C from Section III.

B. Experiment Preparation with ChaLearn Signaling Gestures Dataset

Figure 6 shows same kinematics features extracted from videos of another dataset without a single change to the feature extraction algorithm. Extracted features are then encoded in corresponding affinity matrices.

[6] Rprop training for Artificial Neural Networks
http://uk.mathworks.com/matlabcentral/fileexchange/32445-rprop

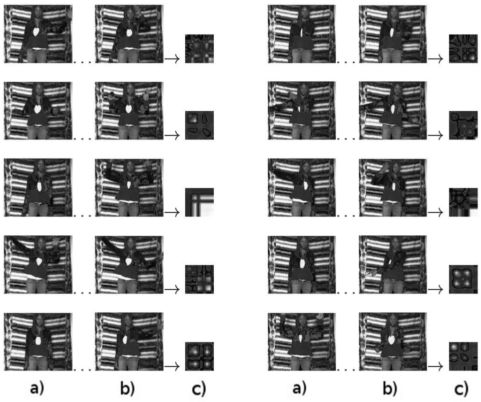


Fig. 6. Affinity matrices generation from gestures in video data from partial ChaLearn signaling gestures dataset (single gesture approx. 2-5 seconds)
a) first frame b) last frame c) generated affinity matrix

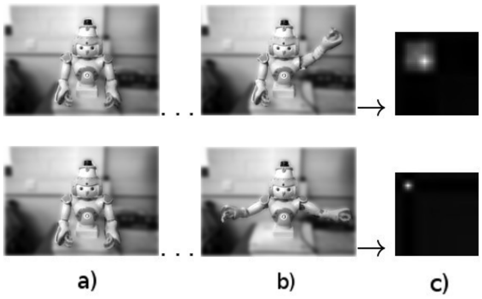


Fig. 7. Affinity matrices generation from gestures in video data from self-made NAO gestures dataset (single gesture approx. 50 seconds)
a) first frame b) last frame c) generated affinity matrix

C. Experiment Preparation with Self-Made NAO Gestures Dataset

Figure 7 shows the same feature extraction algorithm used on an artificial subject. Separate Haar feature-based cascade classifier was trained to detect face and torso the artificial subject.

VI. Experiments Results

All experiments had been conducted using leave-one-out strategy, testing on a single testing instance for every gesture class.

Neither feature extraction, nor feature encoding, nor affinity matrix algorithms have been edited between experiments, except for use of different Haar feature-based cascade classifier when capturing features of an artificial subject.

Images of the evolved detectors represent neural network weights as heatmaps. The black colour means positive weights and grey colour represents negative weights, while the white colour means the weight is zero. There is no identification or order in the presented heatmaps. Distinctiveness of the detectors for individual experiment has importance on the accuracy of the system.

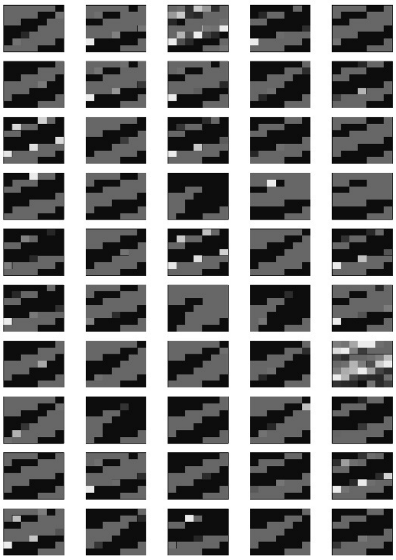


Fig. 8. Heatmaps of 50 detectors, evolved for 130 generations for one experiment, for the self-made gestures dataset

Fitness score is superficial as the evolution would never be able to reach 100% fitness. The maximum is taken as a case when all the values of affinity matrices are uniformly distributed, which is never the case with affinity matrices for gestures.

A. Experiment Preparation with Self-Made Gestures Dataset

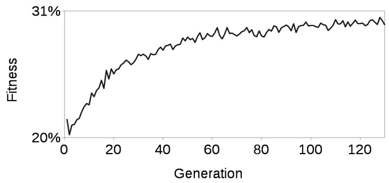


Fig. 9. Average fitness development of detectors evolved for 130 generations for self-made gestures dataset

1) Fitness Evolution: The fitness evolution appears to be steady and continues until 130_{th} generation where it becomes apparent that the fitness tends to converge at around 31% fitness. The evolution is terminated manually after 130 generations.

2) Evolved Detectors: Evolved detectors after 130 generations are presented in Figure 8. Most of the detectors on the figure are slightly different from each other. This shows that the evolutionary algorithm attempted to make detectors distinct, but more evolutions were required to see bigger differences.

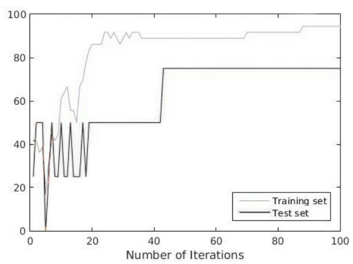
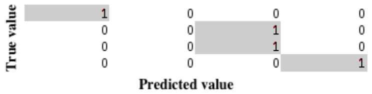


Fig. 10. Classifier training with Rprop for self-made gestures dataset

3) Classifier Training: Figure 10 shows training of the classifier for 100 generations. The training accuracy lies between 90% and 95%, while the testing accuracy achieved 75%.



1	0	0	0
0	0	1	0
0	0	1	0
0	0	0	1

True value

Predicted value

Fig. 11. Confusion matrix for self-made gestures dataset with leave-one-out strategy using one example for classifier, described in Section III-E. Average accuracy: 75%

4) Confusion Matrix: The recognition algorithm confused the raise one arm up gesture with the not raising arms up gesture. This may have happened due to the poor feature extraction on that particular case, where the arm may not have been detected by the algorithm.

B. Experiment Preparation with ChaLearn Signaling Gestures Dataset

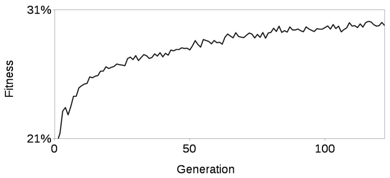


Fig. 12. Average fitness development of detectors evolved for 120 generati for ChaLearn gestures dataset

1) Fitness Evolution: The fitness development, shown Figure 12, becomes unstable after approximately 60 generations and tends to converge at around 31% fitness. The evolution is terminated after 120 generations.

2) Evolved Detectors: Figure 13 presents evolved detectors for ChaLearn dataset after 120 generations. Although the pattern is very similar for most of the detectors, definite variety can be noticed.

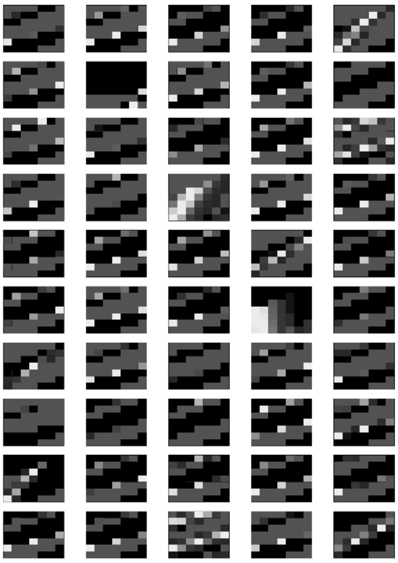


Fig. 13. Heatmaps of 50 detectors, evolved for 120 generations for one experiment, for ChaLearn gestures dataset

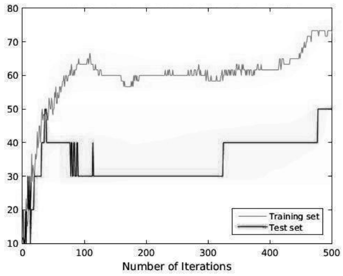
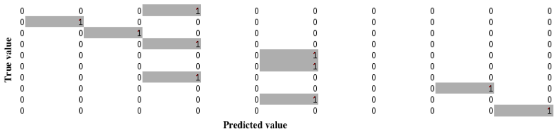


Fig. 14. Classifier training with Rprop for ChaLearn gestures datase

3) Classifier Training: Figure 14 shows training of the classifier. Accuracy of the training set lies around 70%, while the test dataset accuracy is no greater than 50%.



0	0	0	1	0	0	0	0	0	0
0	1	0	0	0	0	0	0	0	0
0	0	1	0	0	0	0	0	0	0
0	0	0	1	0	0	0	0	0	0
0	0	0	0	0	1	0	0	0	0
0	0	0	0	0	1	0	0	0	0
0	0	0	1	0	0	0	0	0	0
0	0	0	0	0	0	0	0	1	0
0	0	0	0	0	1	0	0	0	0
0	0	0	0	0	0	0	0	0	1

True value

Predicted value

Fig. 15. Confusion matrix for ChaLearn gestures dataset with leave-one-out strategy using one example for classifier, described in Section III-E. Average accuracy: 50%

4) Confusion Matrix: Figure 15 shows confusion matrix for the ChaLearn dataset. It is apparent that the algorithm has many misclassifications. In particular, the classifier labels the first gesture as the fourth one. The gestures are indeed very similar with the only difference in another hand active during the gesturing of the third gesture. This can be explained by the poor feature extraction. Same holds for gesture number 7 being mixed with gesture number 4.

C. Experiment Preparation with Self-Made NAO Gestures Dataset

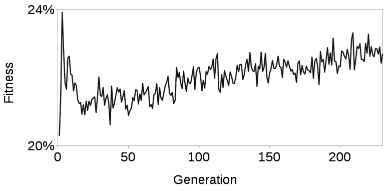


Fig. 16. Average fitness development of detectors evolved for 230 generations for self-made NAO gestures dataset

1) Fitness Evolution: Figure 16 presents fitness evolution of detectors, applied on encoded and transformed NAO gestures. The evolution is very unstable, but slowly improving. There is a spike of fitness in the first generations, for which there is no definite explanation. This may have been a feature of evolved detectors that performed very well on the training data, but that feature was lost in the next generations. The evolution is terminated after 230 generations.

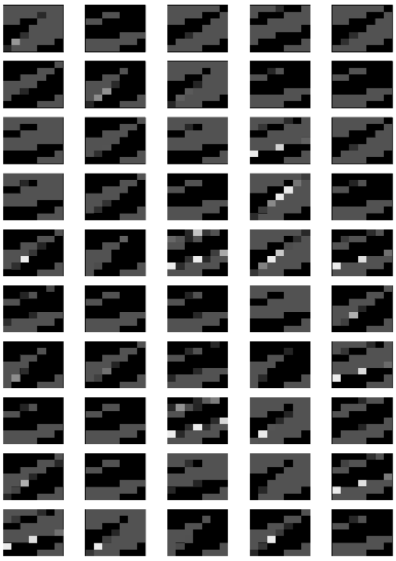


Fig. 17. Heatmaps of 50 detectors, evolved for 230 generations for one experiment, for self-made NAO gestures dataset

2) Evolved Detectors: Figure 17 shows evolved detectors after 230 generations. Some detectors are similar, but overall some variety is noticeable.

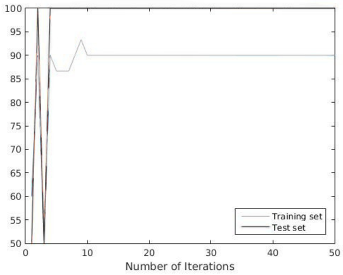
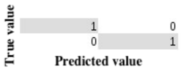


Fig. 18. Classifier training with Resilient Backpropagation for self-made NAO gestures dataset

3) Classifier Training: Figure 18 shows the training of the classifier. Although the accuracy on training dataset is around 90%, the test dataset scores 100% in just few iterations. This can be explained by the fact that only two gestures are classified.


True value
Predicted value

1	0
0	1

Fig. 19. Confusion matrix for self-made NAO gestures dataset with leave-one-out strategy using one example for classifier, described in Section III-E. Average accuracy: 100%

4) Confusion Matrix: Since the accuracy on the training set is 100%, confusion matrix shows that the predicted class for two gestures is always correct.

VII. Discussion

It can be seen from the affinity matrices, presented in Figures 11, 15, and 19, that the classification accuracy drops down as the number of gesture classes increases (2 gestures - 100% accuracy, 4 gestures - 75% accuracy, 10 gestures - 50% accuracy). Complexity of the gestures was not expected to be a major factor in the recognition accuracy. The major factor that affects the accuracy, on the other hand, is the feature extraction, which currently is very simple and does not account for such gesture details as seen in ChaLearn dataset. Currently, the feature extraction looks only at the face, upper body and the limbs of the subject.

During the experiments it had been noticed that the feature extraction should be tailored to every dataset due to the variations in camera positioning with respect to the subject, illumination, and others. Nevertheless, the system is robust enough, considering that nothing had been changed between the different experiments.

With different datasets, which may include more gesture details (e.g. sign languages), the system would have to be improved by extending the feature extraction.

Evolved detectors had very little variation in all conducted experiments. This may be due to few evolution generations or the incorrectness of the fitness function. Evolution of detectors has to be studies separately to investigate how many distinct detectors can be evolved.

VIII. Conclusion

This research has planned the initial steps to research on indirect communication between two agents. As a result of the study, a real-time gesture recognition system had been produced that is partly developed with the use of the evolutionary techniques.

Preliminary results of this project show that the gesture recognition using the proposed system is possible and can be refined by improving the accuracy of the feature extraction algorithm. This work should be seen as the first step towards the creation of real self-organised systems based on evolution that can be applied to social robots and thus facilitate human-robot-interaction.

Future work lies in further testing of the system on public datasets. Further on, potential extensions to the system may include additional feature extraction to accommodate the algorithm for the sign language recognition and processing. Segmentation is another possible extension that would make the system even more robust and complete.

Acknowledgment

This work has been supported by the Heriot-Watt University School of Engineering & Physical Sciences James Watt Scholarship through Edinburgh Centre for Robotics.

References

[1] J. Aggarwal and S. Park, "Human motion: modeling and recognition of actions and interactions," in *3D Data Processing, Visualization and Transmission, 2004. 3DPVT 2004. Proceedings. 2nd International Symposium on*, Sept 2004, pp. 640–647.

[2] K. Davids, D. Araújo, and R. Shuttleworth, "Applications of dynamical systems theory to football," *Science and football V*, pp. 537–550, 2005.

[3] F. Dylla, A. Ferrein, G. Lakemeyer, J. Murray, and O. Obst, "Approaching a formal soccer theory from behaviour specifications in robotic soccer ."

[4] A. Bogdanovych, C. Stanton, X. Wang, and M.-A. Williams, *RoboCup 2011: Robot Soccer World Cup XV*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, ch. Real-Time Human-Robot Interactive Coaching System with Full-Body Control Interface, pp. 562–573. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-32060-6$_4$8

[5] Y. Kong, X. Zhang, Q. Wei, W. Hu, and Y. Jia, "Group action recognition in soccer videos," in *Pattern Recognition, 2008. ICPR 2008. 19th International Conference on*, Dec 2008, pp. 1–4.

[6] H. Kitano, M. Asada, Y. Kuniyoshi, I. Noda, E. Osawai, and H. Matsubara, *RoboCup-97: Robot Soccer World Cup I*. Berlin, Heidelberg: Springer Berlin Heidelberg, 1998, ch. RoboCup: A challenge problem for AI and robotics, pp. 1–19. [Online]. Available: http://dx.doi.org/10.1007/3-540-64473-3$_4$6

[7] A. Bezek, M. Gams, and I. Bratko, "Multi-agent strategic modeling in a robotic soccer domain," in *Proceedings of the fifth international joint conference on Autonomous agents and multiagent systems*. ACM, 2006, pp. 457–464.

[8] T. Nakashima, M. Takatani, M. Udo, H. Ishibuchi, and M. Nii, *RoboCup 2005: Robot Soccer World Cup IX*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, ch. Performance Evaluation of an Evolutionary Method for RoboCup Soccer Strategies, pp. 616–623. [Online]. Available: http://dx.doi.org/10.1007/11780519$_6$1

[9] R. N. Parasuraman, K. Kershaw, and M. F. Perez, "Experimental investigation of radio signal propagation in scientific facilities for telerobotic applications," *International Journal of Advanced Robotic Systems*, vol. 10, no. 364, pp. 1–11, July 2013. [Online]. Available: http://oa.upm.es/30850/

[10] S. Nolfi and D. Floreano, *Evolutionary Robotics: The Biology,Intelligence,and Technology*. Cambridge, MA, USA: MIT Press, 2000.

[11] P. Vargas, E. Di Paolo, I. Harvey, and P. Husbands, *The Horizons of Evolutionary Robotics*. MIT Press, 3 2014.

[12] P. Garg, N. Aggarwal, and S. Sofat, "Vision based hand gesture recognition."

[13] J. J. LaViola, Jr., "A survey of hand posture and gesture recognition techniques and technology," Providence, RI, USA, Tech. Rep., 1999.

[14] P. Trigueiros, F. Ribeiro, and L. Reis, "Generic system for human-computer gesture interaction," in *Autonomous Robot Systems and Competitions (ICARSC), 2014 IEEE International Conference on*, May 2014, pp. 175–180.

[15] M. Malgireddy, I. Inwogu, and V. Govindaraju, "A temporal bayesian model for classifying, detecting and localizing activities in video sequences," in *Computer Vision and Pattern Recognition Workshops (CVPRW), 2012 IEEE Computer Society Conference on*, June 2012, pp. 43–48.

[16] D. Faria, C. Premebida, and U. Nunes, "A probabilistic approach for human everyday activities recognition using body motion from rgb-d images," in *Robot and Human Interactive Communication, 2014 RO-MAN: The 23rd IEEE International Symposium on*, Aug 2014, pp. 732–737.

[17] U. Gaur, Y. Zhu, B. Song, and A. Roy-Chowdhury, "A "string of feature graphs" model for recognition of complex activities in natural videos," in *Computer Vision (ICCV), 2011 IEEE International Conference on*, Nov 2011, pp. 2595–2602.

[18] M. Brand, N. Oliver, and A. Pentland, "Coupled hidden markov models for complex action recognition," in *Computer Vision and Pattern Recognition, 1997. Proceedings., 1997 IEEE Computer Society Conference on*, Jun 1997, pp. 994–999.

[19] M. Bregonzio, S. Gong, and T. Xiang, "Recognising action as clouds of space-time interest points," in *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, June 2009, pp. 1948–1955.

[20] G. Willems, T. Tuytelaars, and L. Gool, "An efficient dense and scale-invariant spatio-temporal interest point detector," in *Proceedings of the 10th European Conference on Computer Vision: Part II*, ser. ECCV '08. Berlin, Heidelberg: Springer-Verlag, 2008, pp. 650–663. [Online]. Available: http://dx.doi.org/10.1007/978-3-540-88688-4$_4$8

[21] D. Lisin, M. Mattar, M. Blaschko, E. Learned-Miller, and M. Benfield, "Combining local and global image features for object class recognition," in *Computer Vision and Pattern Recognition - Workshops, 2005. CVPR Workshops. IEEE Computer Society Conference on*, June 2005, pp. 47–47.

[22] T. Kocmánek, "HyperNEAT and Novelty Search for Image Recognition," Master's thesis, Czech Technical University in Prague, 2015.

[23] K. O. Stanley, D. B. D'Ambrosio, and J. Gauci, "A hypercube-based encoding for evolving large-scale neural networks," *Artif. Life*, vol. 15, no. 2, pp. 185–212, Apr. 2009. [Online]. Available: http://dx.doi.org/10.1162/artl.2009.15.2.15202

[24] T. G. van den Berg and S. Whiteson, "Critical factors in the performance of hyperneat," in *Proceedings of the 15th Annual Conference on Genetic and Evolutionary Computation*, ser. GECCO '13. New York, NY, USA: ACM, 2013, pp. 759–766. [Online]. Available: http://doi.acm.org/10.1145/2463372.2463460

[25] M. Riedmiller and H. Braun, "A direct adaptive method for faster backpropagation learning: the rprop algorithm," in *Neural Networks, 1993., IEEE International Conference on*, 1993, pp. 586–591 vol.1.

Appendix B

Towards Continuous Sign Language Recognition with Deep Learning

Boris Mocialov[1], Graham Turner[2], Katrin Lohan[3], Helen Hastie[4]

Abstract— **Humans communicate with each other using abstract signs and symbols. While the cooperation between humans and machines can be a powerful tool for solving complex or difficult tasks, the communication must be at the abstract enough level that is both natural to the humans and understandable to the machines. Our paper focuses on natural language and in particular on sign language recognition. The approach described here combines heuristics for segmentation of the video stream by identifying the epenthesis with stacked LSTMs for automatic classification of the derived segments. This approach segments continuous stream of video data with the accuracy of over 80% and reaches accuracies of over 95% on segmented sign recognition. We compare results in terms of the number of signs being recognised and the utility of various features used for the recognition. We aim to integrate the models into a single continuous sign language recognition system and to learn policies for specific domains that would map perception of a robot to its action. This will improve the accuracy of understanding the common task within the shared activity between a human and a machine. Such understanding, in turn, will foster meaningful cooperation.**

I. INTRODUCTION

Interacting with machines, users are required to use the input devices, such as remote controls, keyboards, or touch interfaces, provided with these machines. The input devices usually eliminate the uncertainty of the user input to show that the machine functions properly and reliably. While the provided input devices are reasonable communication mediums, they are neither intuitive nor natural for a human user, which leads to either an overhead of learning how to use the given input devices or even inability to use a machine, perhaps due to a disability. Such limited communication hinders opportunities for effective collaboration between humans and machines.

To eliminate this limiting factor, the machines should understand interaction that is natural to the user. This natural interaction could be achieved using natural language or gesturing, while the natural language, in turn, could be either spoken languages or sign languages [1]. We focus on the recognition of sign languages from video in this paper. Whilst much research has been done on the partially analogous problem of continuous speech recognition, few researchers have investigated sign language recognition. There is a general misconception that sign languages are simply gestures with simple rules, in fact this is not the case. A single sign, corresponding to a word or concept, is multimodal from the perspective of the producer and can have many variations within a single language. Compounding the problem is the fluid nature of signing where signs are interleaved with transitional motions called *epenthesis*, which themselves are easily confusable with signs. This is combined with synchronous facial recognition making the feature space very large and the problem complex.

In Section II, this paper identifies relevant research for the segmented, continuous, and vocabulary-based sign language recognition and divides the overall problem into three high-level sub-problems, listing the methods that are used by other authors to tackle these sub-problems. Section III describes the dataset that has been used for this paper and presents the methodology in terms of a process pipeline. Sections IV and V present results after experimenting with every component of the pipeline individually. Section VI discusses the results and their meaning in the context of the continuous sign language recognition. Finally, Section VII concludes the paper and outlines the future development of this work that aims to tackle the uncertainty in recognition for continuous signing.

II. RELATED WORK

Much previous work has focused on the recognition of signs in terms of isolated, segmented video snippets with a clear start and end time [2], [3], [4]. Alternatively, continuous sign language recognition focuses on the stream of signs in a sentence with the task to process a signed sentence and produce aligned *glosses*, which is the written form of a signed sentence in words [5], [1].

The final approach is analogous to keyword spotting in automatic speech recognition, where a finite list of signs is spotted in the video [1], [6]. This is the middle ground between the isolated sign and continuous sign language recognition and the approach that we adopt here. Our approach breaks down into 3 sub-problems, which will be discussed here in terms of previous work: 1) feature extraction; 2) detection of the movement epenthesis as a means of segmentation; and 3) classification of segmented signs.

*This work was supported by the Heriot-Watt University School of Engineering & Physical Sciences James Watt Scholarship and Engineering and Physical Sciences Research Council (EPSRC) as part of the CDT in Robotics and Autonomous Systems at Heriot-Watt University and The University of Edinburgh (Grant reference EP/L016834/1) as well as the RAEng/Leverhulme Trust Senior Research Fellowship Scheme (Hastie/LTSRF1617/13/37)

[1] Boris Mocialov is with the Department of Computer Science, the School of Engineering & Physical Sciences, and the Edinburgh Centre of Robotics, Heriot-Watt University, Edinburgh, UK bm4@hw.ac.uk

[2] Prof. Graham Turner is with the Department of Languages & Intercultural Studies, Heriot-Watt University, Edinburgh, UK g.h.turner@hw.ac.uk

[3] Dr. Katrin Lohan is with the Department of Computer Science, Heriot-Watt University, Edinburgh, UK and the Edinburgh Centre of Robotics k.lohan@hw.ac.uk

[4] Prof. Helen Hastie is with the Department of Computer Science, Heriot-Watt University, Edinburgh, UK and the Edinburgh Centre of Robotics h.hastie@hw.ac.uk

Firstly, local feature extraction methods from noisy input data have recently become more precise [7], [8], [9], although, some challenges, such as tackling occlusions, still persist. The majority of the sign languages consist of manual (hands, fingers, posture) and non-manual features (facial expressions), which makes them multimodal from the signer's perspective. The features are used in parallel and tend to complement each other. Specific features, in some cases, may not be required in order to interpret the sign [5]. The common local features used for sign language recognition are body posture (shoulders, neck, waist), hands (elbows, wrists, and phalanges), and facial features (mouth and eyes).

Once the features are chosen, they should be tracked throughout the frames to get all the information that forms a sign [10]. In [11], the author questions whether all parts of the signing features are equally important during signing and how much movement and configuration variations are allowed for the sign to be recognised. In fact, [12] have shown that the index finger is the salient finger during signing and determines the speed and amplitude of signing with other fingers following the motion of the index finger. This theory is supported by [13], who shows that physiologically this should be the case that not all fingers are dominant during signing, which makes it applicable to any sign language. In the work described here, we will examine the utility of a number of different feature sets.

Secondly, regarding segmentation by means of motion epenthesis modelling, this is directed towards explicit detection of the motion between the intended signs during signing. The detection of the motion epenthesis can be achieved with dynamic programming [6], which is advantageous, because it does not require training as with machine learning approaches [1].

Thirdly, isolated signs have been previously modelled to incorporate both spatial and temporal information, such as sequential pattern mining that fuses multimodal signals [14]. The same paper uses regression, SVM and LSTM for comparisons and concludes that the models that incorporate spatial and temporal features are superior. More recent work on networks allow the network to be trained on videos of different lengths [15], which is useful because the same signs may be of different lengths due to signing speed. Most promising results are achieved with deep learning techniques, such as CNN with temporal convolution and pooling for spatio-temporal representations or RNN with long short-term memory (LSTM) to learn the mapping of feature sequences to sequences of glosses [16].

Our approach requires a large amount of quality data. In recent years, the situation regarding sign language data has improved with more readily available larger datasets that are realistic rather than simulated, and involve more complex interactions for specific tasks, such as explaining directions or story retelling [17], [18], [19].

III. Methodology

Figure 1 shows the processing pipeline for the continuous sign language recognition with the raw video data input and the recognised individual signs as output. Further, the dataset used for the training and testing of the system will be introduced. Finally, the parts of the pipeline will be discussed in detail.

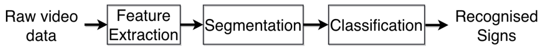


Fig. 1: Data processing pipeline for the continuous sign language recognition

A. Dataset

Due to the annotation quality, a portion of the NGT[1] corpus has been used for this project. The corpus contains approximately 100 participants telling stories, or having discussions with other Dutch sign language users.

We have chosen a part of the corpus where participants retell the Canary Row cartoon of Tweety & Sylvester by the Warner Brothers Pictures. Details about the recording setup for the corpus can be found in [17].

TABLE I: Chosen classes for training (glosses translated from Dutch with Google Translate)

Classes	Glosses	Maximum signers/class
0-10	ape, building, electricity, handwriting, look, poet, rain, run, shake, tram	7
10-20	ball, binoculars, bird, birdcage, inside, not, ready, rope, same, search	4
20-30	and, apartment, climb, corner, how, hurry, line, old, pipe, thinking	6
30-40	window, clothes, box, suitcase, contact, aunt, draw, music, funny, tighten	4

Table I lists the glosses that the models were trained on. The choice of the glosses was guided by the amount of the available instances of that particular class. The more example videos of the sign there were present in the dataset, the more likely the sign had been chosen for the training.

The mean length of a sign is 6.75 frames where one frame length is approximately 40 milliseconds. The average amount of examples per sign is approximately 11 videos and was unfortunately not enough to train our models. Therefore, for every selected class, additional data was generated using extracted features from the original data. For every video example of the real data, 200 more examples were synthesised by adding perturbation along both x and y axes to the extracted features from the original examples. For the first 100 synthesised examples, the same perturbation has been added to every extracted feature, while for the second 100 synthesised examples, different perturbations were added to every extracted features along the x and y axes. This was done to synthesise examples of a sign, where, for example, the hand is moved further from the body or the face of the signer than in the original example.

[1] Sign Language of the Netherlands - NGT (Nederlandse Gebarentaal), is the language of the deaf community in the Netherlands

B. Feature Extraction

Used features resemble the features provided by the commercial sensors, such as Microsoft Kinect. Instead of using an additional high-cost sensor such as Kinect, a standard camera is used and features are extracted with the help of the deep learning techniques, provided by the openpose library[2] [7], [8], [9].

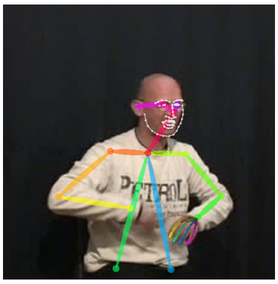


Fig. 2: NGT dataset feature extraction example with openpose. The lines are drawn between identified body features, such as shoulder, neck, phalanges, etc.

Figure 2 shows an arbitrary frame from the NGT corpus after the openpose feature extraction algorithm is applied. The algorithm provides information about the body pose, hands, and facial features. The limitation of the openpose algorithm is that it does not recover the features when occlusions are present, which is very common during signing as the hands occlude each other and the face.

C. Segmentation

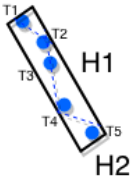


Fig. 3: Example of hand trajectory during signing that is used to decide whether the motion is epenthesis or a part of a sign. T1-T5 correspond to centroids of hand contour, acquired during feature extraction; H1 and H2 are height and width of the minimum bounding box for the T1-T5

The main assumption for the segmentation is that the hands move slower during the signing than during the motion epenthesis. Motion epentheses are identified by looking at the distance travelled by each hand an interval. In this particular experiment, 5 frames are chosen for this interval for detection of the motion epenthesis as was reported in [20]. Using the extracted features from the hands, as can be seen in Figure 2, the centroids of all the hand points are calculated and accumulated for the period of 5 frames (T1-T5 on Figure 3). Later, the minimum bounding box is calculated for the hand trajectory over 5 frames (black rectangle on the figure). At the end, the longest side of the minimum bounding box (either H1 or H2 from the figure) is taken to decide whether the segment is motion epenthesis or a part of the sign. Both H1 and H2 are considered, because the hand may travel in any direction during signing. Using similar techniques as in [20], the segment is labelled as epenthesis if the longest side of the minimum bounding box is between 18 and 60 pixels.

D. Classification

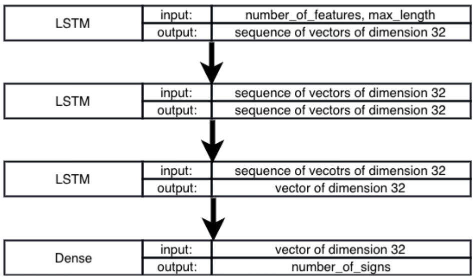

LSTM
input:
number_of_features, max_length
output:
sequence of vectors of dimension 32
LSTM
input:
sequence of vectors of dimension 32
output:
sequence of vectors of dimension 32
LSTM
input:
sequence of vecotrs of dimension 32
output:
vector of dimension 32
Dense
input:
vector of dimension 32
output:
number_of_signs

Fig. 4: Model architecture for the TensorFlow library, consisting of stacked LSTM layers and one Dense layer that outputs the sign class. Inputs and outputs specify the type of inputs and outputs for a particular layer of the network.

With the video segmented, isolated sign language recognition is done by training deep learning models using TensorFlow[3] and openpose libraries. The architecture, shown in Figure 4, is composed of three stacked LSTM layers with the first two layers producing a sequence of vectors with 32 dimensions and the last LSTM layer producing a single vector, composed of 32 dimensions. At the output of the network, the dense layer outputs the likelihood of every sign. The first layer accepts a sequence of inputs (chunks) of length equals to the number of extracted features per one frame. The maximum number of chunks is set to be the longest sequence of frames for a sign and all other sequences are padded at the end with zeros. The network is trained offline with the objective function set to categorical cross entropy and the optimizer set to resilient backpropagation with the adaptive learning rate, which is a good choice for the recurrent neural networks.

Extracted features, such as posture, finger, and facial information are combined together by stacking feature vectors together for the isolated video of a single sign.

The dataset is split into training, validation, and testing sets. The training data consists of 80% of the overall dataset, validation and testing sets consists of 10% of the overall dataset each. All the dataset is shuffled before performing the split into training, validation, and testing sets. This means

[2] https://github.com/CMU-Perceptual-Computing-Lab/openpose/

[3] https://www.tensorflow.org/

that the method is not signer independent as the testing set is likely to contain some variation of the same signer from the training set. The future experiments will test the signer independent condition for a more robust solution.

IV. Results: Segmentation Accuracy

One continuous single-signer video has been used for testing the accuracy of the segmentation. The ground truth was annotated by considering the time between every gloss in the annotation file to be the epenthesis motion, with 206 motion epenthesis occurrences annotated.

The epenthesis detection returns start and end times of the epenthesis interval. To calculate the accuracy in terms of F-measure, the returned epenthesis interval is compared to the ground truth, extracted from the annotated video. As a result, the algorithm identified 201 True Positives TP that lied within the ground truth ($Predicted \in GT$). Some of the identified intervals are repeated, due to the fact that both hands are tracked and analysed for the epenthesis identification. The algorithm identified 39 False Positives FP that did not match epintheses in the ground truth ($Predicted \not\subseteq GT$). All the intervals that were not included in the predicted TP are assumed to be True Negatives TN ($Predicted \in \neg GT$). The algorithm identified 210 TN intervals. The intervals that were considered and were not in the ground truth were assumed to be False Negatives FN ($Predicted \not\subseteq \neg GT$). The algorithm identified 46 FN intervals.

$$F - measure =$$

$$(2 * Precision * Recall)/(Precision + Recall) = \mathbf{0.825}, \text{ where}$$

$$Precision = TP/(TP + FP) = \mathbf{0.837} \quad \text{and}$$

$$Recall = TP/(TP + FN) = \mathbf{0.813}$$

V. Results:Classification

Figure 6 shows the training progress of the model, trained for classifying 10-40 classes of individual signs from the NGT corpus. The figures suggest that the training can produce effective model for the recognition of the signs. However, the training is not stable, the accuracy fluctuates between the epochs and occasionally drops down to the random choice accuracy level. When the model is trained with facial features, the performance degrades, because the input feature vector is increased in size, which makes it more difficult for the model to generalise. When the number of features is reduced from full facial to reduced facial information, the accuracy increases, but does not surpass the accuracy of the model without the facial features. Generally, the more classes the model is trained to distinguish, the more challenging the recognition task.

Table II shows the accuracies, achieved on the testing data for models, trained for 100 epochs on different amount of classes. It is worth noting that not the best, but the last trained model has been used on the training data.

The table shows that the best accuracy is achieved with the lowest number of classes and that the accuracy degrades with addition of more extracted features. This result could arise due to the amount of features used for the recognition, some of which could be perceived only as noise during the training and the recognition, as they do not convey any meaning for the chosen signs.

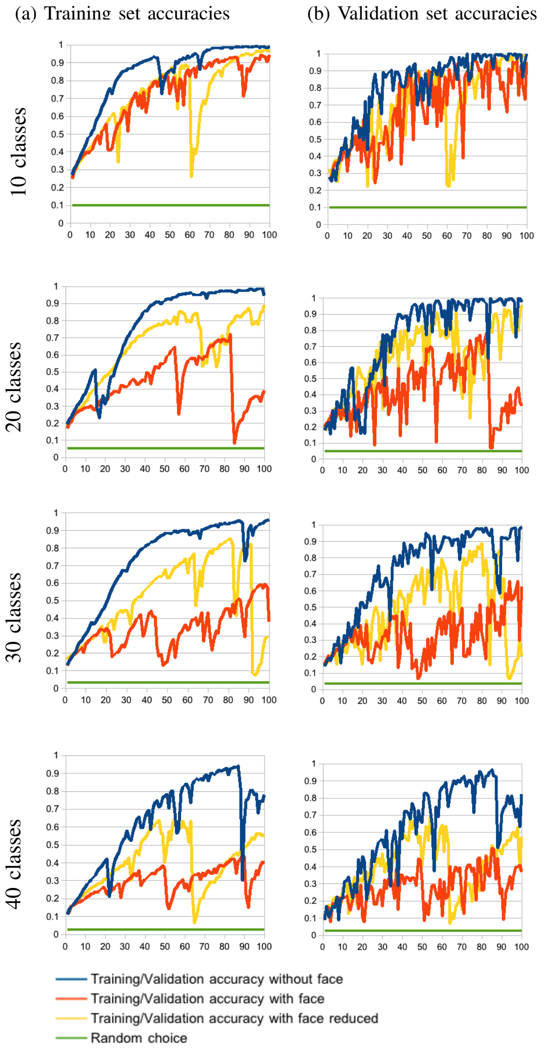


Fig. 6: Individual sign language classification model training for 10, 20, 30, and 40 classes. The graphs in the right column correspond to the training accuracy and the graphs in the left column correspond to the validation accuracy. X-axes correspond to the number of epochs the model was trained on, while the Y-axes correspond to the accuracy of the model on the validation set.

VI. Discussion

Segmentation accuracy indicates that the approach that uses heuristics to detect epenthesis can achieve sufficient results. Varying parameters of the segmentation may yield

TABLE II: Testing accuracies

	Without face information	**With face reduced features**	**With face full features**
10 classes	0.999	0.992	0.955
20 classes	0.972	0.951	0.344
30 classes	0.983	0.207	0.625
40 classes	0.807	0.572	0.378

different results by manipulating the threshold of the H1 and/or H2 and simultaneously changing the number of frames for which H1 and H2 are computed. By allowing more frames, it would be more likely for the H1 or H2 to increase, because the epenthesis will become a part of the segment. Therefore, it is important to use the information about the average sign length and choose the number of frames to be fewer than the average number of frames per sign.

The classification results suggest that for the selected signs, listed in Section III-A, the inclusion of the facial features degrades the classification accuracy, whether all the features are chosen or the reduced amount. More experiments will be required to identify whether these results are consistent for the different signs, even those that are heavily dependent on the facial features. Additional consultation with a linguist will be needed to identify which signs are heavily dependent on the facial features and which are not in the NGT dataset. Obtained results support the claim that not all extracted features are necessary for the successful classification of signs.

VII. Conclusion and Future Work

The paper presented the continuous sign language recognition pipeline that uses heuristic approach for epenthesis detection and deep learning for isolated signs recognition in a continuous stream of video data. The methods show adequate results when tested individually, while more resources need to be invested for an integrated continuous sign language recognition system. The paper investigated the utility of the extracted features for the sign language recognition model. The results suggest that, for the selected signs from the NGT dataset and the chosen stacked LSTM model, not all the features are necessary to perform relatively accurate sign language recognition. Our primary goal is to support continuous natural interaction between the user and the machine as we focus on sign languages as means for communication. Sole segmentation and recognition of the perceived signs is not enough to achieve the understanding of sign language between the human and the machine in terms of dialogue. To cope with the occasional misclassifications, we propose to learn policies for the specific domains (i.e. navigation domain) that map perception to action and reduce the classification confusion, as the choices of actions available to the machine will be restricted by the current state.

References

[1] G. Fang, W. Gao, and D. Zhao, "Large-vocabulary continuous sign language recognition based on transition-movement models," *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans*, vol. 37, no. 1, pp. 1–9, Jan 2007.

[2] H. Wang, X. Chai, X. Hong, G. Zhao, and X. Chen, "Isolated sign language recognition with grassmann covariance matrices," *ACM Transactions on Accessible Computing (TACCESS)*, vol. 8, no. 4, p. 14, 2016.

[3] Y. Jiang, J. Tao, W. Ye, W. Wang, and Z. Ye, "An isolated sign language recognition system using rgb-d sensor with sparse coding," in *2014 IEEE 17th International Conference on Computational Science and Engineering*, Dec 2014, pp. 21–26.

[4] K. M. Lim, A. W. Tan, and S. C. Tan, "A feature covariance matrix with serial particle filter for isolated sign language recognition," *Expert Systems with Applications*, vol. 54, pp. 208–218, 2016.

[5] O. Koller, J. Forster, and H. Ney, "Continuous sign language recognition: Towards large vocabulary statistical recognition systems handling multiple signers," *Computer Vision and Image Understanding*, vol. 141, no. Supplement C, pp. 108 – 125, 2015, pose & Gesture. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1077314215002088

[6] R. Yang, S. Sarkar, and B. Loeding, "Handling movement epenthesis and hand segmentation ambiguities in continuous sign language recognition using nested dynamic programming," *IEEE transactions on pattern analysis and machine intelligence*, vol. 32, no. 3, pp. 462–477, 2010.

[7] Z. Cao, T. Simon, S.-E. Wei, and Y. Sheikh, "Realtime multi-person 2d pose estimation using part affinity fields," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[8] T. Simon, H. Joo, I. Matthews, and Y. Sheikh, "Hand keypoint detection in single images using multiview bootstrapping," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[9] S.-E. Wei, V. Ramakrishna, T. Kanade, and Y. Sheikh, "Convolutional pose machines," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[10] H. Cooper, B. Holt, and R. Bowden, *Sign Language Recognition*. London: Springer London, 2011, pp. 539–562.

[11] A. Gineke, M. J. Reinders, E. A. Hendriks, H. de Ridder, and A. J. van Doorn, "Influence of handshape information on automatic sign language recognition," in *International Gesture Workshop*. Springer, 2009, pp. 301–312.

[12] S. Ojala, T. Salakoski, and O. Aaltonen, "Coarticulation in sign and speech," in *workshop Multimodal Communication, from Human Behaviour to Computational Models*, 2009.

[13] J. Ann, "On the relation between ease of articulation and frequency of occurrence of handshapes in two sign languages," *Lingua*, vol. 98, no. 1, pp. 19 – 41, 1996, sign Linguistics Phonetics, Phonology and Morpho-syntax. [Online]. Available: http://www.sciencedirect.com/science/article/pii/0024384195000313

[14] H. Cate, F. Dalvi, and Z. Hussain, "Sign language recognition using temporal classification," *CoRR*, vol. abs/1701.01875, 2017. [Online]. Available: http://arxiv.org/abs/1701.01875

[15] N. C. Camgöz, S. Hadfield, O. Koller, and R. Bowden, "Subunets: End-to-end hand shape and continuous sign language recognition," *ICCV 2017 Proceedings*, 2017.

[16] R. Cui, H. Liu, and C. Zhang, "Recurrent convolutional neural networks for continuous sign language recognition by staged optimization," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[17] O. A. Crasborn and I. Zwitserlood, "The corpus NGT: an online corpus for professionals and laymen," 2008.

[18] R. Nishio, S.-E. Hong, S. König, R. Konrad, G. Langer, T. Hanke, and C. Rathmann, "Elicitation methods in the DGS (german sign language) corpus project," in *Poster presented at the 4th Workshop on the Representation and Processing of Sign Languages: Corpora and Sign Language Technologies, following the 2010 LREC Conference in Malta, May 22.-23., 2010.*

[19] G. Quinn, A. Merrison, B. Davies, K. Pollitt, and G. Turner, "Task-oriented discourse between british sign language (BSL) users," in *British Association for Applied Linguistics 41st Annual Meeting*, 2008.

[20] A. Choudhury, A. K. Talukdar, M. K. Bhuyan, and K. K. Sarma, "Movement epenthesis detection for continuous sign language recognition," *Journal of Intelligent Systems*, 2017.