NAZARBAYEV UNIVERSITY

Towards Large-Vocabulary Kazakh Sign Language Processing: Corpus collection, Semi-automatic annotation, Recognition, and Translation

by

Medet Mukushev

Submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy in Science Engineering and Technology

Date of Completion
August, 2025

Towards Large-Vocabulary Kazakh Sign Language Processing: Corpus collection, Semi-automatic annotation, Recognition, and Translation

by
Medet Mukushev

Submitted in partial fulfillment of the requirements for the degree of
Doctor of Philosophy in Science Engineering and Technology

School of Engineering and Digital Sciences
School of Sciences and Humanities
Nazarbayev University

August, 2025

Supervised by
Prof. Anara Sandygulova
Prof. Matteo Rubagotti
Prof. Deniz Başkent

Declaration

I, Medet Mukushev, declare that the research contained in this thesis, unless otherwise formally indicated within the text, is the author's original work. The thesis has not been previously submitted to this or any other university for a degree and does not incorporate any material already submitted for a degree.

Signature: 

Date: 21.08.2025

BLANK

Abstract

Sign language (SL) is the primary communication mode for Deaf communities globally, yet automatic Sign Language Processing (SLP) technologies lag significantly behind those for spoken languages, particularly for under-resourced languages like Kazakh Sign Language (KSL). Progress is hindered by critical challenges: severe scarcity of large-scale datasets capturing diverse signers and continuous, natural signing; the difficulty in computationally representing both manual and crucial non-manual linguistic features (e.g., facial expressions, mouthing); and the laborious, time-consuming nature of manual data annotation. This thesis directly confronts these obstacles by developing foundational resources and methodologies specifically tailored for large-vocabulary, continuous KSL processing.

We address data scarcity by introducing two novel, large-scale KSL datasets: FluentSigners-50, collected via community crowdsourcing to maximize signer and environmental diversity, and KSL-OnlineSchool, leveraging extensive online interpreted educational content to achieve a large vocabulary. Together, these provide over 900 hours of video data, forming an unprecedented resource for KSL. To tackle representation challenges, we propose and evaluate a framework encompassing both manual components, including an extensive study on automatic handshape classification, and non-manual components like head movements, facial expressions, and mouthing. Addressing the annotation bottleneck, we developed SLAN-tool, an open-source, web-based platform employing machine learning models for semi-automatic signing segmentation and handshape classification, designed to accelerate corpus creation.

Finally, the utility of these resources is demonstrated by establishing baseline performance metrics for state-of-the-art Sign Language Recognition (SLR) and Translation (SLT) models evaluated on challenging, purpose-built splits of the FluentSigners-50 dataset. The primary contributions: the creation and release of the first large-scale continuous KSL datasets, the proposed sign representation framework, and the open-source semi-automatic annotation tool, collectively provide essential infrastructure to catalyze future research and development in KSL processing and related low-resource SLP tasks.

BLANK

Acknowledgments

First and foremost, I would like to express my deepest gratitude to my supervisor, Professor Anara Sandygulova. Her continuous support, insightful guidance, and expertise in world-class research were invaluable throughout this research journey. Her encouragement and meticulous feedback were instrumental in shaping this thesis and bringing it to completion. I am also profoundly thankful to my co-supervisors, Professor Matteo Rubagotti and Professor Deniz Başkent. Their combined supervision provided a rich and supportive environment for this interdisciplinary work.

I wish to extend my sincere appreciation to Nazarbayev University, particularly the School of Engineering and Digital Sciences, for providing the opportunity and resources to pursue this doctoral research. The stimulating academic environment has been crucial for my development.

This research would not have been possible without the contributions and collaboration of many individuals. I am particularly grateful to my colleagues and collaborators whose names appear alongside mine on the publications stemming from this work: Vadim Kimmelman, Alfarabi Imashev, Arman Sabyrov, Kenesary Koishybay, Madina Sultanova, and Aidyn Ubingazhibov. Their collaboration on various aspects of this project, including dataset design, linguistic analysis, annotation tool development, and experimental evaluation, has been invaluable. I also thank the broader research group for stimulating discussions.

I am deeply indebted to the members of the Deaf community in Kazakhstan who participated in the collection of the KSL: FluentSigners-50 dataset. Their willingness to share their language and time was fundamental to this research. I also thank the professional KSL interpreters who assisted in the initial stages and the dedicated annotators who worked on both the FluentSigners-50 and KSL-OnlineSchool datasets, including those involved with the Surdobot platform. My gratitude extends to the El-arna TV channel and the interpreters whose work made the OnlineSchool data collection possible.

I would also like to acknowledge the undergraduate students at NU who have assisted with various project stages, such as data collection or preliminary experiments.

Furthermore, I am thankful for the support and feedback received from faculty members at NU during various stages of my research. I would like to thank Professors Luis Ramon Rojas-Solórzano and Konstantinos Kostas for their valuable feedback during research seminars.

Finally, and most importantly, I want to express my heartfelt gratitude to my family. Their unwavering belief, constant encouragement, patience, and sacrifices throughout my lengthy academic journey have been my greatest source of strength and inspiration. This accomplishment would not have been possible without them.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

The development of effective systems for automatic Sign Language Processing (SLP), especially for languages with fewer resources like Kazakh Sign Language (KSL), is held back by major challenges. A primary obstacle, which this thesis addresses directly, is the critical lack of large, suitable datasets needed for training modern recognition and translation models. Processing sign languages, which are visual and complex, presents unique difficulties not found in speech recognition, making data even more crucial. This thesis works to overcome these key challenges for KSL, aiming to improve communication access for its users.

Sign language (SL) is the main way Deaf communities communicate worldwide. As visual languages, sign languages use hand gestures, facial expressions, and body movements to convey meaning. It's important to know that sign language is not universal; each country often has its own distinct sign language, with over 300 different sign languages used globally by about 70 million people [World Health Organization, 2021]. Like spoken languages, these are natural languages with their own linguistic structures, including phonology, morphology, syntax, and semantics [Sandler & Lillo-Martin, 2006]. Signs are made up of manual features (like hand shapes, movements, and locations) and non-manual features (such as facial expressions, eye gaze, and mouth movements), which work together to express meaning.

Two key tasks in Sign Language Processing are Recognition (SLR) and Translation (SLT). SLR aims to automatically classify sign language videos into text, which is useful for education, accessibility, and human-computer interaction. It is a difficult task because models must understand subtle human motions. This is different from simpler action recognition (like identifying walking or jumping) which deals with a small, fixed set of actions; sign language has a very large and open vocabulary, similar to spoken languages. SLT takes this further, aiming to translate sign language videos directly into written or spoken language sentences. This helps Deaf or hard-of-hearing individuals communicate with non-signers and also aids linguists in studying sign languages. However, SLT is also very challenging due to sign language's visual nature, complex grammar, and the many ways signs can change based on context.

1.1 Main Problems for KSL Processing

Based on the current situation in SLP and specific needs for KSL, this research focuses on three main problems:

Insufficient Data: The primary concern is the lack of large KSL datasets. Modern machine learning methods, especially deep learning, require a lot of diverse data. For KSL, there is not much data, especially for continuous signing. Many existing SL datasets use isolated signs, have few signers, or are recorded in labs, which lacks realism. To build good KSL systems, we first need more and better data. A major challenge for Sign Language Processing (SLP), as noted by Bragg et al.[Bragg et al., 2019b], is that current public sign language datasets have serious limitations. These problems weaken the recognition systems trained on them and make them less useful in different situations. One issue is often the small vocabulary size, partly because recording and annotating datasets costs a lot of time and money. Another significant problem is that many datasets, like MS-ASL[Joze & Koller, 2018] and Devisign[Chai et al., 2014b], mostly contain videos of single, isolated signs. This isn't suitable for real-world use, which needs systems trained on natural, continuous signing found in full sentences and longer conversations.

Limited Sign Representation: To recognize signs well, models must capture the important parts of signing. Many current methods focus too much on hands and ignore important information from non-manual parts like facial expressions, mouth movements, and head/body motion. These non-manuals carry important grammar and meaning in sign language. Ignoring them limits how accurate recognition systems can be. Non-manual components, such as facial expressions and head movements, are recognized by many researchers as very important parts of sign languages [Chatzis et al., 2020]. These non-manual signals can change the meaning of a sign or even distinguish between different types of sentences. Despite their importance, research shows [Koller, 2020a], that these non-manual features are often missing from sign language recognition systems, especially those designed to understand many different signs (medium or large vocabulary systems).

Difficult Annotation: Adding notes (like glosses, details about handshapes) to sign language videos is very slow and hard work when done manually. This "annotation bottleneck" makes it difficult to create the labeled data needed to train models. Faster, easier ways to annotate data are needed. Unlike speech recognition, there are currently no effective computer tools available to help automatically or semi-automatically annotate sign language videos. Because of this lack of tools, annotating sign language data becomes a very slow and difficult manual task [Kopf et al., 2021].

Taken together, these significant challenges create major barriers for developing effective Sign Language Processing systems for Kazakh Sign Language and clearly motivate the need for the new resources and methods presented in this thesis.

1.2 Goals and Research questions

Because of the problems discussed earlier (lack of data, limited representation, difficult annotation), the main goal of this thesis is to build basic resources and methods for processing Kazakh Sign Language, focusing on large vocabulary and continuous signing. Our specific goals are:

- First, to tackle the data problem, we aim to create the first large KSL datasets suitable for continuous Sign Language Recognition and Translation, using different ways to collect data to ensure variety.
- Second, to find good ways to represent KSL videos, including both manual (hand) and important non-manual parts.
- Third, to help overcome the annotation problem, we aim to make annotation easier by creating and testing a semi-automatic annotation tool..
- Finally, to validate these solutions, we will test how well current SLR/SLT methods work on the new KSL data we created, establishing baseline performance..

These specific research questions break down our main goals into focused areas of investigation. The research presented throughout this thesis is designed to systematically find answers to these questions.

- RQ1: What are good ways to collect large KSL datasets with many different signers and a big vocabulary? This question relates to our first goal and is primarily addressed in Chapters 3 and 4.
- RQ2: How can we represent KSL manual and non-manual parts for models, and how important are non-manuals for recognition accuracy? This relates to our second goal and is discussed in Chapter 5.
- RQ3: What features does a semi-automatic annotation tool need to help with KSL data annotation? This question connects to our third goal and the tool presented in Chapter 6.
- RQ4: How well do modern SLR/SLT models perform on our new KSL datasets, and can methods like transfer learning help improve translation? This relates to our fourth goal, with results presented in Chapter 7.

  These goals and research questions set the clear direction and focus for the research undertaken in this thesis. The chapters that follow will describe the work done to achieve these aims and answer these specific questions.

1.3 Contributions

This PhD research led to several outcomes that address the goals and research questions outlined above. These include academic publications, new datasets created for KSL, and software tools.

- Novel KSL Datasets: We created the first large-scale Kazakh Sign Language datasets designed for continuous signing analysis: KSL: FluentSigners-50 (Chapter 3) and KSL: OnlineSchool (Chapter 4). These datasets feature high signer variability and large vocabularies, collected using innovative crowdsourcing and online resource strategies.
- Sign Representation Framework: We developed and evaluated methods for representing both manual (handshape classification) and measuring importance of non-manual components in KSL videos (Chapter 5).
- Semi-Automatic Annotation Tool: We designed, implemented, and evaluated SLAN-tool, an open-source, web-based tool to assist and accelerate the annotation of sign language videos (Chapter 6).
- Benchmark Results: We established the first baseline performance results for state-of-the-art SLR and SLT models on the new KSL datasets, providing a crucial reference point for future research (Chapter 7)

These contributions have been partly disseminated through peer-reviewed publications during the course of this PhD.

1.3.1 Publications and Joint Work

The work done during this PhD led to the following publications:

1.3.1.1 Journal papers:

- Mukushev, M., Ubingazhibov, A., Kydyrbekova, A., Imashev, A., Kimmelman, V., Sandygulova, A. (2022). FluentSigners-50: A signer independent benchmark dataset for sign language processing. PloS one, 17(9), e0273649. (Chapter 3)

1.3.1.2 Conference papers:

- Mukushev, M., Kydyrbekova, A., Imashev, A., Kimmelman, V., Sandygulova, A. (2022, June). Crowdsourcing Kazakh-Russian Sign Language: FluentSigners-50. In Proceedings of the Thirteenth Language Resources and Evaluation Conference (pp. 2541-2547). (Chapter 3)

- Mukushev, M., Sabyrov, A., Imashev, A., Koishibay, K., Kimmelman, V., Sandygulova, A. (2020). Evaluation of manual and non-manual components for sign language recognition. In Proceedings of The 12th Language Resources and Evaluation Conference. European Language Resources Association (ELRA). (Chapter 5)

1.3.1.3 Workshop papers:

- Mukushev, M., Kydyrbekova, A., Kimmelman, V., Sandygulova, A. (2022, June). Towards Large Vocabulary Kazakh-Russian Sign Language Dataset: KRSL-OnlineSchool. In Proceedings of the LREC2022 10th Workshop on the Representation and Processing of Sign Languages: Multilingual Sign Language Resources (pp. 154-158). (Chapter 4)

- Mukushev, M., Imashev, A., Kimmelman, V., Sandygulova, A. (2020). Automatic classification of handshapes in Russian Sign Language. In Proceedings of the LREC 2020 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. European Language Resources Association (ELRA). (Chapter 5)

- Sabyrov, A., Mukushev, M., Kimmelman, V., Sandygulova, A. (2019, June). Towards Real-time Sign Language Interpreting Robot: Evaluation of Non-manual Components on Recognition Accuracy. In CVPR Workshops. (Chapter 5)

- Mukushev, M., Sabyrov, A., Sultanova, M., Kimmelman, V., Sandygulova, A. (2022, June). Towards Semi-automatic Sign Language Annotation Tool: SLAN-tool. In sign-lang@ LREC 2022 (pp. 159-164). European Language Resources Association (ELRA). (Chapter 6)

The joint work done during this PhD led to the following publications not included to this thesis:

- Kimmelman, V., Imashev, A., Mukushev, M., Sandygulova, A. (2020). Eyebrow position in grammatical and emotional expressions in Kazakh-Russian Sign Language: A quantitative study. PloS one, 15(6), e0233731.

- Kuznetsova, A., Imashev, A., Mukushev, M., Sandygulova, A., Kimmelman, V. (2021, August). Using computer vision to analyze non-manual marking of questions in KRSL. In Proceedings of the 1st International Workshop on Automatic Translation for Signed and Spoken Languages (AT4SSL) (pp. 49-59).

- Koishybay, K., Mukushev, M., Sandygulova, A. (2021, January). Continuous Sign Language Recognition with Iterative Spatiotemporal Fine-tuning. In 2020 25th International Conference on Pattern Recognition (ICPR) (pp. 10211-10218). IEEE.
- Abutalipov, A., Janaliyeva, A., Mukushev, M., Cerone, A., Sandygulova, A. (2020, October). Handshape Classification in a Reverse Dictionary of Sign Languages for the Deaf. In International Symposium: From Data to Models and Back (pp. 217-226). Springer, Cham.
- Imashev, A., Mukushev, M., Kimmelman, V., Sandygulova, A. (2020, November). A dataset for linguistic understanding, visual evaluation, and recognition of sign languages: The k-rsl. In Proceedings of the 24th Conference on Computational Natural Language Learning (pp. 631-640). (Chapter 3)

1.4 Thesis Structure

This thesis is organized to present the research in a logical flow, addressing the problems and research questions step-by-step:

- Chapter 2 (Related Work): Discusses previous work in SLP, highlighting the gaps this thesis fills.
- Part I: Creating Datasets (Chapters 3 and 4): Details the collection and characteristics of the new KSL datasets (FluentSigners-50 and OnlineSchool), addressing RQ1.
- Part II: Methods and Tools (Chapters 5-7): Presents the technical contributions.
  - Chapter 5 (Sign Language Representation): Explores methods for representing manual and non-manual features, addressing RQ2.
  - Chapter 6 (Automatic Sign Language Annotation): Describes the SLAN-tool developed to ease annotation, addressing RQ3.
  - Chapter 7 (Sign Language Recognition and Translation): Reports the baseline experimental results on the KSL datasets using modern models, addressing RQ4.
- Chapter 8 (Conclusion): Summarizes the key findings and contributions, discusses limitations, and suggests directions for future work.

Chapter 2

Literature review

This chapter reviews previous research relevant to the work presented in this thesis. We will discuss studies on recognizing isolated signs (ISLR), recognizing continious sign sentences (CSLR), translating sign language (SLT), classifying handshapes, and tools used for sign language annotation. In each section, we examine existing methods and datasets, pointing out the limitations and research gaps that motivate the contributions of this thesis.

2.1 Isolated Sign Language Recognition

ISLR focuses on identifying individual signs performed in isolation, typically within short video clips. This task is usually approached as a classification problem.

2.1.1 Datasets for ISLR

Several specialized datasets have been created for ISLR. Examples include ASLLVD ([Neidle et al., 2012], DEVISIGN-L (includes RGB-D data)[Chai et al., 2014a], MS-ASL (collected from online videos)[Joze & Koller, 2018], Slovo (online sources)[Kapitanov et al., 2023] and WL-ASL (increased vocabulary and signers number)[Li et al., 2020a]. These datasets differ in language, vocabulary size, number of signers, and recording modalities. Table 2.1 provides an overview of several commonly used datasets appropriate for CSLR.

When annotating videos of single, isolated signs, researchers assign a label called a "gloss" to the segment containing the sign. This gloss represents the meaning of that specific sign. A gloss can be a word from the sign language itself or a written label describing the sign.

Several datasets are commonly used for training Isolated Sign Language Recognition (ISLR) systems, each with different characteristics. For example, CSL-500[Pu et al., 2016] is a Chinese Sign Language (CSL) dataset featuring 500 distinct sign glosses performed by 50 signers. It is often used as a starting point for feature learning before fine-tuning on larger CSL data. Another CSL dataset, DEVISIGN-L [Chai et al., 2014a], contains 2000 words recorded from 8 signers, resulting in 24,000 instances. The sample video and recording setup is shown on Figure 2.1. It was recorded using an RGB-Depth setup with a Microsoft Kinect capturing RGB video at 1280x720

Table 2.1: Datasets used for ISLR.

Datasets	Language	Signers	Vocabulary	Samples
ASLLVD-L (2012) [Neidle et al., 2012]	American	6	2284	8,585
CSL-500 (2014) [Pu et al., 2016]	Chinese	50	500	125,000
DEVISIGN-L (2014) [Chai et al., 2014a]	Chinese	8	2000	24,000
MS-ASL (2018) [Joze & Koller, 2018]	American	222	1000	25,513
WL-ASL (2020) [Li et al., 2020a]	American	119	2000	21,013
SLOVO (2023) [Kapitanov et al., 2023]	Russian	194	1000	20,000

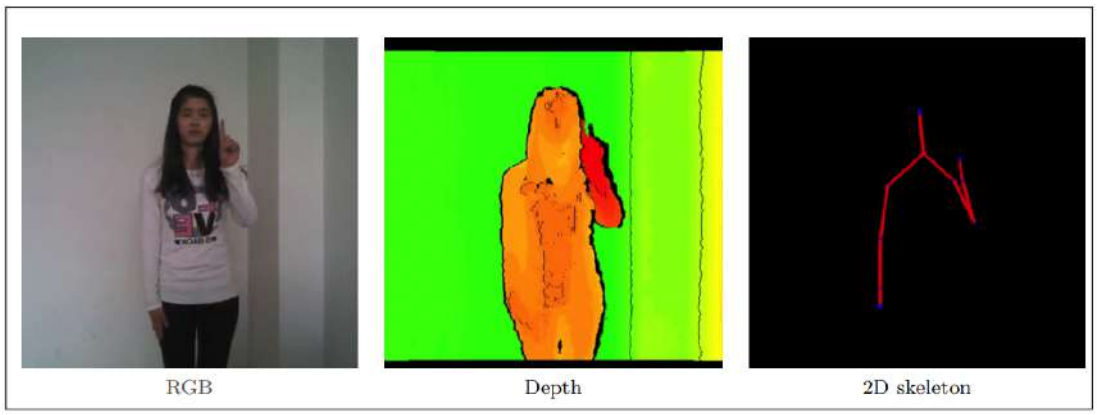


Figure 2.1: DEVISIGN dataset screenshots [Chai et al., 2014a]

resolution, depth information at 512x424, and 2D upper-body skeleton data, all at 30 frames per second. Smaller subsets named DEVISIGN-G and DEVISIGN-D are also available.

For American Sign Language (ASL), the ASLLVD [Neidle et al., 2012] dataset includes 2,284 signs performed by one to six native signers each, totaling 8,585 samples. Its recording setup included multiple camera views (two front views at different resolutions/frame rates, plus side and head close-up views), and annotations provide sign start/end times and handshape information. MS-ASL[Joze & Koller, 2018] is another significant ASL dataset, offering 1,000 distinct glosses from 222 signers. Sourced from YouTube videos, it features a wide variety of background settings, making it valuable for training models robust to real-world conditions. MS-ASL is also divided into four subsets (ASL1000, ASL500, ASL200, ASL100) with varying numbers of classes, signers, and instances, and provides signer-independent training, validation, and test splits. Lastly, WL-ASL[Li et al., 2020a] is an ASL dataset containing 2,000 unique glosses from 119 signers.

Created using crowdsourcing platforms, Slovo [Kapitanov et al., 2023] contains 20,000 samples of 1,000 different isolated RSL signs performed by 194 signers making it a large and varied resource for the language. Alongside the dataset, the authors detail their entire creation pipeline – including video collection using templates, multi-worker validation, and time-interval annotation – as a methodology contribution.

Figure 2.2: Sample video screenshots from SLOVO dataset [Kapitanov et al., 2023]

2.1.2 ISLR Methods

ISLR task aims to accurately detect specific signs in videos. This task is often handled similarly to action or gesture recognition, requiring methods that can find and learn distinctive features from the video.

One common approach in ISLR research is to focus on specific areas like the hands and mouth in the video frames. The goal is often to remove distracting background information that could interfere with classification. For example, Liao et al.[Liao et al., 2019] proposed a method that extracts the hand area and then uses 3D ResNet and BLSTM networks for classification. Similarly, Aly et al. [Aly & Aly, 2020] developed a method using the DeepLabv3+ algorithm to segment hand regions, then used other neural networks (Convolutional Self-Organizing Map and BLSTM layers) to classify the signs based on those regions.

Another strategy researchers use to improve ISLR accuracy and robustness is to combine different types of visual information, like standard video frames (RGB), optical flow (motion information), or skeleton joint data (body posture). These "multi-stream" approaches usually require more computation but can be better at handling confusing situations where one type of feature isn't enough. For instance, Sarhan et al. [Sarhan & Frintrop, 2020] used a two-stream network with both RGB and optical flow input, processed by I3D networks, to achieve reliable recognition. Rastgoo et al. [Rastgoo et al., 2020] also used a multi-stream approach, combining hand images, hand heatmaps, and skeleton data, feeding these combined features into 3D-CNN and LSTM networks for sign recognition.

2.1.3 Limitations

In summary, while various methods exist for recognizing isolated signs, their main limitation is the focus on single signs performed out of context. This approach does not fully address the challenge of understanding natural, continuous sign language as used in everyday communication, which leads us to the more complex task of Continuous Sign Language Recognition.

Table 2.2: Datasets used for Continuous Sign Language Recognition.

Datasets	Language	Signers	Vocabulary	Samples
SIGNUM [Von Agris & Kraiss, 2007]	German	25	780	780
RWTH-BOSTON-400 [Dreuw et al., 2008]	American	4	483	843
RWTH-PHOENIX [Cihan Camgoz et al., 2018]	German	9	2887	8257
Video-Based CSL [Huang et al., 2018a]	Chinese	50	178	25000
BSL-1K [Albanie et al., 2020a]	British	40	1064	273000
How2Sign [Duarte et al., 2020]	American	11	16 000	35000

2.2 Continuous Sign Language Recognition

Continuous Sign Language Recognition (CSLR) involves the task of recognizing sequences of signs, such as those forming sentences or phrases, within longer video recordings. This task is generally treated as a sequence-to-sequence problem in machine learning and is considered significantly more challenging than ISLR due to factors like co-articulation between signs and the need for temporal segmentation.

2.2.1 Datasets

This subsection reviews various sign language datasets suitable for CSLR research. The availability of high-quality datasets is critically important for advancing both SLR and SLT tasks. Datasets for sign language research can utilize different recording modalities: 1) motion-capture data obtained from sensors placed on the body (e.g., [Benchiheub et al., 2016, Lu & Huenerfauth, 2010, Jedlička et al., 2022]); 2) RGB-D data captured using depth cameras like Microsoft Kinect (e.g., [Oszust & Wysocki, 2013, Cooper et al., 2012]); and 3) standard RGB video data, which is increasingly preferred due to its direct applicability to real-world scenarios using readily available cameras. Datasets may contain either videos of isolated signs or continuous signing sequences. Table 2.2 provides an overview of several commonly used datasets appropriate for CSLR.

Continuous sign language datasets can be further categorized by their annotation level: (1) datasets with sentence-level annotations provide transcriptions or translations for entire signed utterances; (2) datasets with gloss-level annotations provide sequences of labels (glosses) corresponding to individual signs within the utterance. A gloss is typically a written word (often from a spoken language) used to represent a specific sign. Datasets featuring continuous signing provide more realistic data reflecting natural communication patterns but are generally more complex and resource-intensive to create and annotate compared to isolated sign datasets. The choice between sentence-level and gloss-level annotations depends on the specific research goal, with glosses being essential for training CSLR models directly, while sentence translations are needed for SLT.

Developing high-performance deep learning models for CSLR and SLT requires

large datasets, often containing thousands of examples. However, as Bragg et al. [Bragg et al., 2019b] emphasized, the field currently suffers from a scarcity of publicly available, large-scale sign language corpora. They identified several major concerns regarding existing datasets, including often limited vocabulary sizes, the lack of spontaneous (real-life) signing data, the frequent use of novice signers or interpreters rather than fluent native users, and insufficient diversity among signers. Considering the importance of fluency and naturalness, distinguishing between datasets based on contributor expertise (e.g., native signers vs. learners or interpreters) is crucial. Many datasets utilize professional interpreters, who may not be native signers (e.g., hearing CODAs); while fluent, their signing might differ from native signing due to the interpretation process itself (e.g., using calques or literal translations). Furthermore, Bragg et al. [Bragg et al., 2019b] stress the importance of differentiating between datasets containing elicited or prompted content versus those capturing more naturalistic, "real-life" signing, and also considering whether data was collected *in the wild* (varied settings/devices) or under controlled laboratory conditions. An early CSLR benchmark, RWTH-Boston-400 [Dreuw et al., 2008] for ASL, exemplifies some limitations, notably featuring only four signers.

A widely used benchmark for more recent CSLR research is the RWTH-Phoenix-Weather-2014 dataset [Koller et al., 2015]. This German Sign Language (DGS) corpus consists of weather forecast translations performed by nine different signers, recorded from television broadcasts. Sample video frames are shown in Figure 2.3.

Figure 2.3: Sample video screenshots for RWTH-Phoenix-Weather-2014 corpus [Koller et al., 2015].

In contrast to datasets with few signers, the Video-Based CSL dataset

[Huang et al., 2018a] included a large number of participants (n=50). However, it was noted that all recordings took place in identical settings, and many participants appeared unfamiliar with sign language, resulting in slow, artificial signing often lacking natural facial expressions.

The SIGNUM dataset [Von Agris & Kraiss, 2007] for DGS is signer-independent and features fluent participants who are deaf or hard of hearing. Nevertheless, a limitation is its highly controlled recording environment: a single RGB camera, consistent lighting, and a uniform blue background (Figure 2.4). These types of concerns regarding limited variability and naturalness in existing datasets can restrict the robustness and real-world applicability of SLR models developed using them.

More recent datasets have aimed to address some of these challenges. BSL-1K [Albanie et al., 2020a] provides a very large volume of annotated British Sign Language (BSL) data (273,000 samples, 40 signers) sourced from news programs, utilizing automatically generated annotations from subtitles. Using broadcast data provides realism but introduces challenges related to automatic annotation accuracy and potential editing and framing variations. How2Sign [Duarte et al., 2020] offers a large ASL vocabulary (16,000 words) recorded by 11 native signers (35,000 samples).

Figure 2.4: Sample video screenshots for SIGNUM dataset [Von Agris & Kraiss, 2007].

2.2.2 Experiments

Methodologies for CSLR typically rely on extracting informative features from image and video data. Common image feature representations leverage deep learning models like Residual Networks (ResNet) [He et al., 2016a] or architectures with dense connections [Huang et al., 2017, Tan & Le, 2019]. An alternative approach utilizes human keypoint coordinates obtained via pose estimation algorithms. For instance, Dong et al. [Dong et al., 2015a] employed depth cameras to extract body keypoints for ASL recognition training. Gattupalli et al. [Gattupalli et al., 2016] specifically assessed the effectiveness of pose estimation features for SLR.

Video representation learning techniques, often adapted from action recognition [Hernandez Ruiz et al., 2017, Ji et al., 2012, Qiu et al., 2017] and video captioning [Chen et al., 2017], are fundamental to CSLR. Many studies utilize 3D Convolutional Neural Networks (3D-CNNs), first explored in this area by Ji et al. [Ji et al., 2012]. Qiu et al. [Qiu et al., 2017] investigated pseudo-3D residual networks for spatiotemporal feature learning. Tran et al. [Tran et al., 2018a] introduced the ResNet (2+1)D architecture, which outperformed the I3D model [Carreira & Zisserman, 2017a] on action recognition benchmarks. The choice between keypoint-based features and direct video representations often involves trade-offs between computational cost, sensitivity to pose estimation errors, and the ability to capture fine-grained appearance details.

As previously mentioned, CSLR is inherently more complex than ISLR because it must model long-range dependencies over time within sign sequences. Consequently, significant research effort has focused on methods for sequence modeling and optimizing the alignment between video features and the corresponding sequence of sign glosses. Deep neural networks, including Recurrent Neural Networks (RNNs) and Transformers, as well as techniques from reinforcement learning, have been widely explored. Model evaluation in CSLR predominantly uses the RWTH-PHOENIX-Weather 2014 [Koller et al., 2015] and RWTH-PHOENIX-Weather 2014T [Cihan Camgoz et al., 2018] datasets as standard community benchmarks [Bragg et al., 2019b, Koller, 2020b]. Standardized benchmarks like these are essential for comparing the performance of different proposed methods reliably.

Several influential models and techniques have emerged. Zhang et al. [Zhang et al., 2019] were among the first to apply the Transformer architecture [Vaswani et al., 2017] to CSLR, using an encoder-decoder structure within a reinforcement learning setup. They highlighted the effectiveness of the Transformer's attention mechanism for focusing on relevant signing features, achieving a Word Error Rate (WER) of 38.3

To address the challenge of explicitly segmenting signs in time, Huang et al. [Huang et al., 2018b] proposed the Hierarchical Attention Network with Latent Space (LS-HAN). This framework aimed to eliminate error-prone temporal segmentation

pre-processing steps, achieving competitive results (0.617 accuracy metric) by using hierarchical attention over learned latent representations.

Zhou et al. [Zhou et al., 2019] proposed an iterative optimization approach combining I3D features with a temporal expectation-maximization (TEM) module and a Connectionist Temporal Classification (CTC) loss (I3D-TEM-CTC). Their work included a dynamic pseudo-label decoding method using dynamic programming to generate improved frame-to-gloss alignments compared to simpler greedy or probabilistic approaches, leading to a reduced WER of 34.5%.

Significant performance gains have also been demonstrated by incorporating multiple modalities. Koller et al. [Koller et al., 2019] achieved state-of-the-art results (WER 26.0%) with a model that learned features for sign language, mouth shape, and hand shape classification in parallel. This clearly showed the benefit of including non-manual features like mouth movements. This highlights the limitations of methods focusing solely on manual articulation and motivates the exploration of multi-modal representations, a theme relevant to the work in Chapter 5.

Stochastic CSLR [Niu & Mak, 2020] represents another state-of-the-art end-to-end model based on a Transformer encoder and a CTC [Graves et al., 2006] decoder. Its novelty includes representing each sign gloss with a variable number of stochastic states and employing stochastic frame dropping and gradient stopping mechanisms to combat overfitting and improve training efficiency. This model achieved a WER of 25.3% on the RWTH-PHOENIX-Weather 2014 dataset [Koller et al., 2015], outperforming the previous result by Koller et al. [Koller et al., 2019].

2.2.3 Limitations

The reviewed literature reveals significant limitations hindering progress in CSLR. Key challenges relate to datasets, including the general scarcity of large-scale corpora, limited vocabulary sizes, lack of signer and environmental diversity, and a frequent absence of spontaneous, natural signing critical for real-world applicability [Bragg et al., 2019b, Huang et al., 2018a, Von Agris & Kraiss, 2007]. Reliance on existing benchmarks like RWTH-PHOENIX [Koller et al., 2015, Cihan Camgoz et al., 2018] may also not fully capture all real-world complexities. These combined limitations highlight the critical need for more diverse datasets and robust, comprehensive recognition methods, motivating the work presented in this thesis.

The availability of multiple, diverse benchmark datasets such as FluentSigners-50 alongside established corpora is crucial for enabling comprehensive evaluation of future SLR and SLT models, thereby fostering the development of more robust and reliable systems applicable to real-world scenarios.

2.3 Sign Language Translation

Sign Language Translation (SLT) refers to the task of automatically translating sign language videos directly into spoken language, presented either as text or speech. This process involves transforming the visual sign language recordings into a target spoken language output, requiring the system to model not only the sequence of signs (glosses) but also the linguistic structure and grammar necessary for a correct translation [Cihan Camgoz et al., 2018]. SLT represents a critical research direction aimed at improving communication accessibility between Deaf individuals and the hearing/speaking population.

The challenge presented by SLT is generally considered greater than that of CSLR. This increased difficulty stems largely from the need to handle linguistic constraints, including grammatical differences between the visual sign language and the target spoken language, as well as accurately representing the semantics of the spoken language output. Evaluating the quality of SLT systems commonly involves using the Bilingual Evaluation Understudy (BLEU) score [Papineni et al., 2002a]. BLEU is a widely adopted metric in machine translation that measures the similarity between the system's generated translation (candidate text) and one or more human-generated reference translations (ground truth text). Specifically, BLEU-n variants assess the overlap of n-grams (sequences of n words) between the candidate and reference texts. Reporting BLEU scores for n=1 through n=4 is considered good practice for providing a comprehensive view of a translation method's performance. While BLEU is a standard metric, it is important to remember it primarily measures surface-level similarity and may not fully capture translation fluency or adequacy, motivating ongoing research into better evaluation metrics for SLT.

A key distinction between CSLR and SLT is that SLT requires the model to learn not only the sequence of concepts or signs but also the correct word order and grammatical structure of the target spoken language. Figure 2.5 illustrates this difference conceptually, showing CSLR typically outputs a sequence of glosses, while SLT outputs a grammatically complete sentence. A significant contribution enabling SLT research was the introduction by Camgoz et al. [Cihan Camgoz et al., 2018] of the RWTH-PHOENIX-Weather 2014T dataset, which includes spoken German annotations aligned with the DGS videos, providing a benchmark for the task. Their initial work employed attention-based encoder-decoder models, extracting features related to sign glosses from video frames and then applying sequence-to-sequence techniques to translate DGS into German text.

Alternative approaches have also been explored. For example, Ko et al. [Ko et al., 2019] investigated the use of human keypoints extracted using pose estimation techniques [Cao et al., 2019] as input features for SLT. Their rationale was that using higher-level, lower-dimensional features like keypoints could be beneficial.

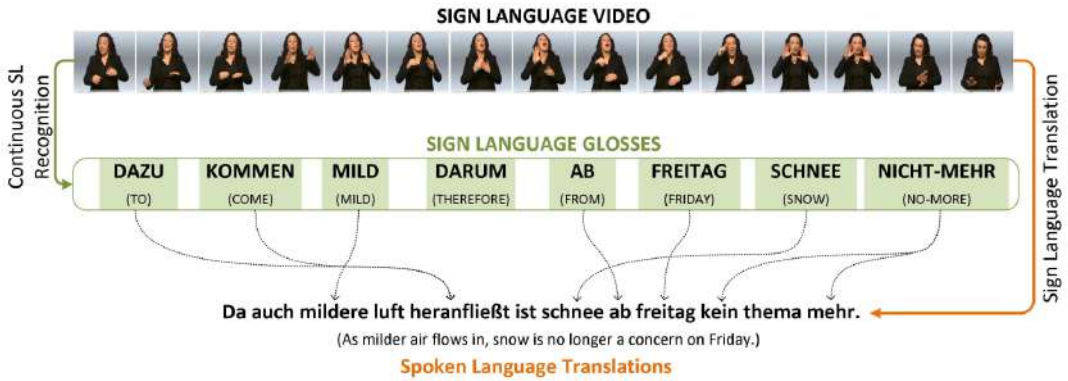


Figure 2.5: Overview of CSLR and SLT tasks. [Cihan Camgoz et al., 2018]

They reported successful training of an SLT system based on OpenPose keypoints [Cao et al., 2019], achieving 55.28% accuracy on their KETI Sign Language Dataset. Seeking to reduce reliance on potentially laborious gloss annotation, Orbay and Akarun [Orbay & Akarun, 2020] proposed a pre-processing approach involving tokenization learned via multi-task learning, demonstrating the potential to achieve higher translation scores without explicit gloss labels. Approaches using keypoints or aiming for gloss-free translation are particularly interesting for potentially simplifying the data requirements for SLT system development.

More recently, Transformer networks, highly successful in text-based machine translation, have shown encouraging results when applied to SLT. Camgoz et al. [Camgoz et al., 2020], for instance, utilized multi-task Transformers to jointly handle sign recognition and translation within a single end-to-end architecture, achieving a BLEU-4 score of 21.32. Their method employed a CTC loss, removing the need for precise ground-truth timing information for glosses while addressing both sequence-to-sequence tasks simultaneously, leading to notable performance gains. This approach encoded video frames using pre-trained spatial embeddings derived from work by Koller et al. [Koller et al., 2019], which were themselves dependent on gloss annotations. The success of multi-task learning suggests potential benefits in leveraging the relationship between recognition and translation, especially in data-constrained scenarios.

2.4 Sign Language Handshape Classification

Research related to classifying handshapes in sign languages has often focused significantly on the sub-problem of fingerspelling recognition. Fingerspelling involves spelling out words letter-by-letter using manual alphabets derived from spoken languages and is considered an important component within broader sign language recognition systems. Progress in this area has largely benefited from advancements in computer vision techniques [Dong et al., 2015b, Mukai et al., 2017, Shi et al., 2018].

Early work leveraging depth-sensing cameras like the Microsoft Kinect demonstrated promising results. For example, Pugeault and Bowden [Pugeault & Bowden, 2011] proposed a real-time system for recognizing ASL fingerspelling based on Kinect data. Their approach focused on detecting the user's hands and extracting handshape features using Gabor filtering on both intensity and depth images, achieving 75% accuracy with a multi-class Random Forest classifier. Dong et al. [Dong et al., 2015b] developed a model recognizing 24 static ASL alphabet signs with 90% accuracy, also using Kinect data; their method extracted hand segments based on depth contrast features to localize hand joints, followed by classification using a Random Forest algorithm. Machine learning techniques combined with classification trees were also applied to Japanese Sign Language by Mukai et al. [Mukai et al., 2017], reporting 86% accuracy for classifying 41 static characters. These initial successes highlighted the utility of depth information for static or controlled fingerspelling recognition but relied on specialized hardware not always available in real-world settings.

Armband wearable sensors represent another modality applied to recognize sign language gestures and fingerspelling. Paudyal et al. [Paudyal et al., 2019] developed the Dynamic Feature Selection and Voting algorithm specifically to detect 26 characters of the ASL alphabet using data from armbands. Their algorithm dynamically selects a list of salient features for each input sample, achieving a best accuracy of 95.28% when using 300 features. These examples illustrate alternative sensor modalities (depth cameras providing 3D structure vs. wearables capturing muscle activity or motion) that complement vision-based approaches, each with its own advantages and limitations regarding setup complexity, intrusiveness, and the type of information captured.

More recent efforts have shifted towards vision-based approaches using standard cameras, aiming for broader applicability. Shi et al. [Shi et al., 2018] made a significant contribution by introducing a large-scale dataset specifically for ASL fingerspelling recognition, importantly including examples of both carefully articulated ("careful") and faster, more natural ("rapid") fingerspelling collected from videos recorded "in the wild" under diverse conditions. This dataset presented a more realistic and challenging scenario compared to previous work relying on studio recordings. They trained baseline recognition models processing sequences of hand-cropped images using either an auto-regressive decoder or Connectionist Temporal Classification (CTC). Their findings indicated that the CTC-based model outperformed the auto-regressive one, yet both achieved relatively low recognition rates, reporting only 35-41% accuracy at the character level. This performance stands in stark contrast to estimated human accuracy (around 82%), highlighting the significant difficulty computers face in recognizing dynamic, natural fingerspelling under unconstrained conditions, even with large datasets.

2.5 Sign Language Annotation Tools

Several software tools are currently available for general video annotation, and many of these are frequently employed for the specific task of annotating sign language data. A useful overview of software that supports time-aligned annotation necessary for sign language research is provided by Perniss [Perniss, 2015].

2.5.1 VIA-SLA: VIA Sign Language Annotator

Woll et al. [Woll et al., 2022] introduced VIA-SLA, a tool designed to automate the segmentation of continuous signing in sign languages. They compared human segmentation with machine-generated segmentation using samples from the British Sign Language (BSL) Corpus. The study demonstrated that VIA-SLA, powered by temporal convolutional networks, significantly outperformed human annotators in speed, with an accuracy of around 78%.

The tool shows great potential in reducing the workload of human annotators and increasing the availability of annotated data for linguistic research and machine learning. However, challenges remain, particularly in improving segmentation accuracy for complex features like fingerspelling. Further testing and refinement are necessary before VIA-SLA can be fully integrated into research workflows.

2.5.2 SLAPE: Sign Language Portable Editor

SLAPE (Sign Language Portable Editor) is a versatile tool designed for the annotation and analysis of sign language video data [Kanis, 2008]. SLAPE's core functionality centers on providing a portable and user-friendly platform that supports detailed sign language transcription, including both manual and non-manual features. The tool facilitates the annotation of hand movements, facial expressions, body posture, and spatial use, which are critical for understanding the full scope of sign language communication.

SLAPE also supports customizable tagging schemes, enabling researchers to adapt the tool for various sign languages and linguistic frameworks. It provides features for time-aligned annotation, allowing users to sync their annotations with specific frames or segments of the video, which is essential for accurate linguistic analysis. Its integration with machine learning models further enhances its utility by enabling semi-automated annotations and facilitating the training of sign language recognition systems.

2.5.3 SLP-AA

Hall et al. [Hall et al., 2022] introduced the Sign Language Phonetic Annotator-Analyzer (SLP-AA), a free and open-source tool for detailed form-based transcription

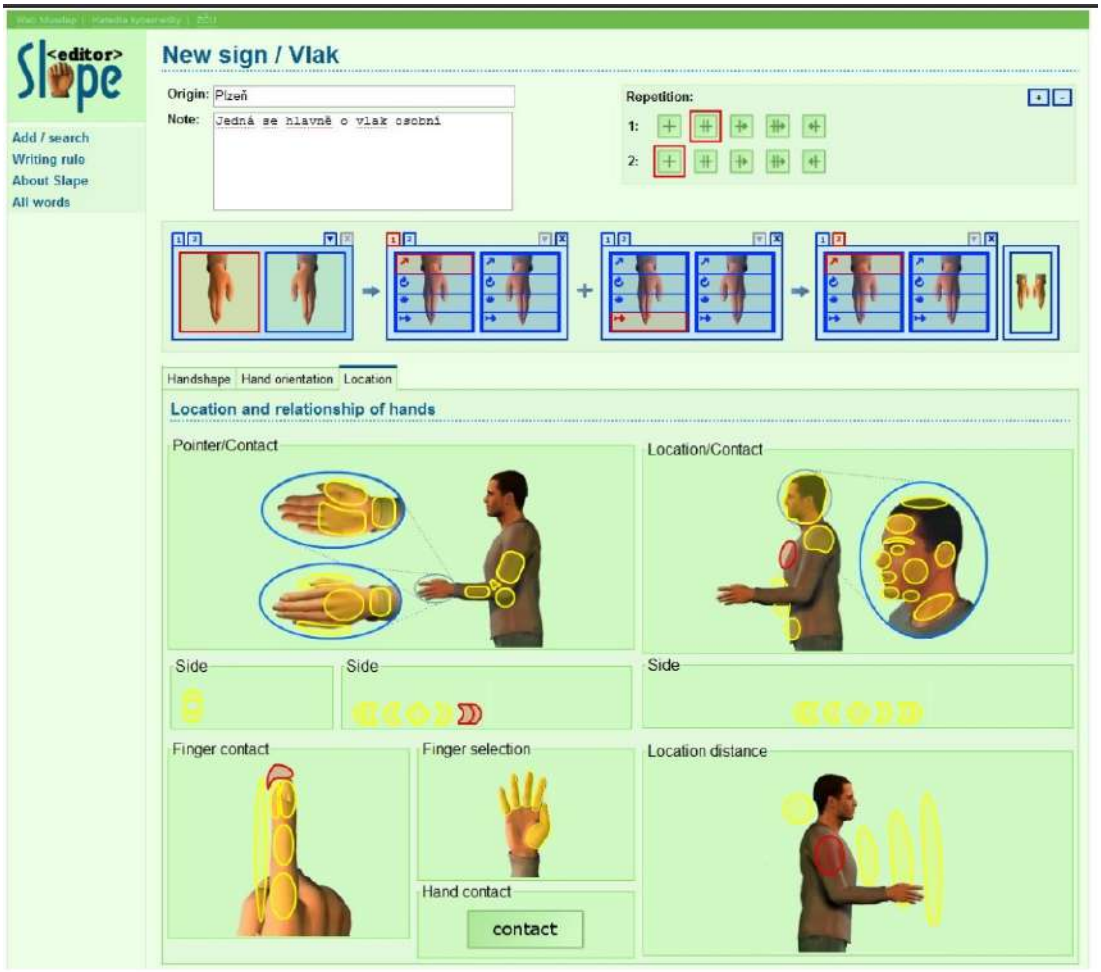


Figure 2.6: The SLAPE editor user interface. [Kanis, 2008]

of sign languages. SLP-AA allows users to transcribe phonetic details, such as movement, location, hand configuration, orientation, and timing relations, without being tied to a specific phonological framework. Its modular design offers flexibility and compatibility with multiple phonological theories, supporting transcription across various sign languages. Although still under development, SLP-AA aims to facilitate phonological analysis and offers additional tools for phonological searches. The authors emphasize that detailed phonetic transcription is crucial for documenting linguistic phenomena and contrasts in sign languages, positioning SLP-AA as a valuable resource for researchers.

2.5.4 SLPAnnotator: Sign Language Phonetic Annotator

Hall et al. [Hall et al., 2017] introduced the SLPAnnotator, a specialized tool designed for the creation of phonetically transcribed corpora of signed languages. The primary focus of SLPAnnotator is on the detailed transcription of hand configurations, which are crucial components in the articulation of signs. The tool allows researchers to document

the precise shapes, orientations, and movements of the hands during signing.

SLPAnnotator stands out due to its emphasis on phonetic precision in the annotation process, ensuring that the physical characteristics of each handshape are accurately captured. The tool is adaptable to different sign languages, allowing for flexibility in transcription based on the specific phonological requirements of the language being studied. It supports the creation of highly detailed and structured annotations, which are essential for linguistic analyses, and enables researchers to build comprehensive, searchable corpora that can serve as the foundation for further phonetic and phonological research.

While primarily focused on hand configuration, SLPAnnotator also offers features that allow for the integration of non-manual markers such as facial expressions and body movements. This holistic approach makes it a valuable resource for building datasets that represent the full complexity of sign language communication.

2.5.5 Non-manual Feature Annotation by Chételat-Pelé et al. (2008)

Chételat-Pelé et al. [Chételat-Pelé & Braffort, 2008] proposed an annotation methodology aimed specifically at annotating non-manual features in sign languages. This methodology emphasizes both precision and simplicity, ensuring that non-manual features are annotated in a way that is both linguistically rigorous and easy to apply. This is particularly important because non-manual components of sign languages can be subtle and complex, involving rapid movements or shifts in facial expressions that occur simultaneously with manual signing. The proposed system offers a framework for consistently capturing these nuances across various datasets.

2.5.6 OpenPose-based Annotation Tool by Fragkiadakis et al. (2021)

Fragkiadakis et al. [Fragkiadakis et al., 2021] introduced an annotation tool that leverages the OpenPose framework [Cao et al., 2019] to automate aspects of the sign language annotation process, particularly for detecting hand use and identifying handshape configurations. This is particularly useful for creating machine-readable datasets, which are essential for training Sign Language Recognition (SLR) models.

Part I

Creating datasets for sign language processing with various approaches

BLANK

Chapter 3

KSL-FluentSigners-50: Addressing Data Scarcity in KSL via Community Crowdsourcing

3.1 Introduction and Motivation

This chapter details the first key contribution of this thesis: tackling the significant data challenges for Kazakh Sign Language (KSL) processing outlined earlier. We describe the creation of a new dataset, FluentSigners-50, through community engagement and discuss its role in enabling research. As established in Chapter 2, progress in SLP is often hindered by limitations in available datasets. Many resources contain primarily isolated signs or lack the vocabulary size and diversity needed for training systems capable of understanding continuous, real-world signing [Koller, 2020a]. Recognizing continuous signing presents considerable difficulties due to co-articulation, depiction, and generalization issues [Bragg et al., 2019a]. Shortcomings in public datasets, such as limited signer variety and constrained recording environments, restrict the performance and applicability of trained models [Bragg et al., 2019b]. Therefore, advancing SLP, especially for less-resourced languages like KSL, requires the development of large, realistic, and diverse datasets. This FluentSigners-50 dataset represents a significant step up in scale and diversity compared to previously available resources for Russian Sign Language and related dialects discussed in Chapter 2, providing a much-needed foundation for KSL research.

To address these needs, this work introduces FluentSigners-50 [Mukushev et al., 2022], a new, large-scale KSL dataset designed as a benchmark for CSLR. It specifically targets three critical limitations often found in existing resources: it features *continuous signing*, includes *signer variety*, and involves *native signers* [Bragg et al., 2019b]. Figure 3.1 illustrates some of the signer variety captured. Acknowledging the importance of signer diversity for model generalization, we employed crowdsourcing methods, collaborating with the local Deaf community in Kazakhstan. While crowdsourcing maximizes

Figure 3.1: K-RSL: FluentSigners-50. Signers showing the sign HI

diversity and ecological validity, it inherently introduces challenges in maintaining consistent data quality across participants, requiring robust validation procedures, as detailed below. A major strength of FluentSigners-50 is the variety among its 50 participants (age 8-57, 18 male and 32 female, diverse hearing status/backgrounds). The use of personal recording devices also resulted in varied settings, lighting, and camera characteristics (Figure 3.3). Since contributors use KSL daily, the dataset reflects substantial linguistic variation, making it suitable for training models for real-life SLR. This inherent variability, while beneficial for robustness, likely necessitates more sophisticated feature extraction methods (explored in Chapter 5) capable of handling noise and diverse visual conditions compared to cleaner lab data.

We establish baseline performance on this dataset using state-of-the-art CSLR (Stochastic CSLR [Niu & Mak, 2020]) and SLT (TSPNet [Li et al., 2020b]) models, allowing comparison with standard benchmarks like RWTH-PHOENIX [Koller et al., 2015]. Establishing these first baseline results for continuous KSL recognition and translation is a crucial step, providing the research community with a reference point for future algorithmic improvements on this new resource. As detailed in Chapter 7, initial baseline results on the signer-independent split (Split 1) achieved competitive Word Error Rates (WER) around 24.9% (+/- 6.2) for CSLR and BLEU-4 scores around 16.0 (+/- 0.8) for SLT, demonstrating the dataset's utility as a challenging benchmark.

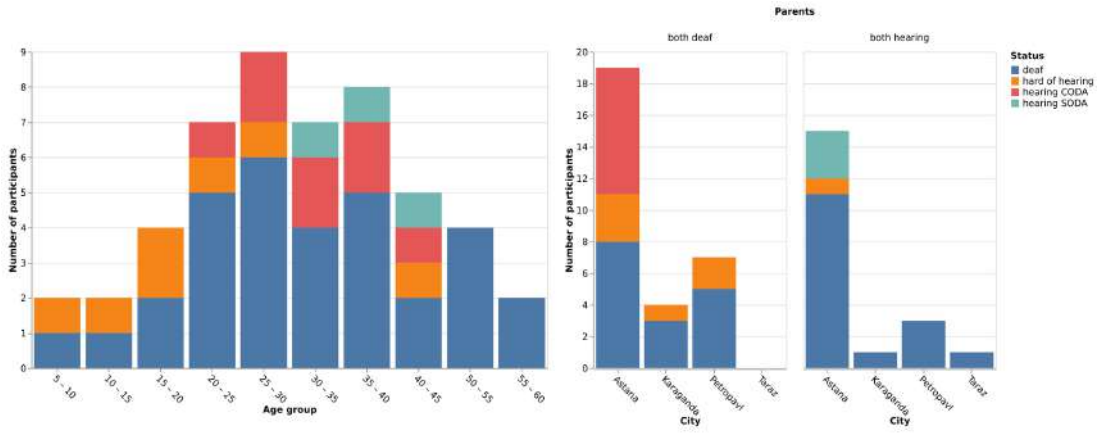
3.2 Dataset Design and Collection Methodology

Addressing the known limitations of prior datasets regarding signer and environmental variability [Koller et al., 2016, Bragg et al., 2019b] was central to the design of FluentSigners-50. The collection methodology aimed specifically to capture diverse, fluent signing in naturalistic settings.

3.2.1 Participant Recruitment and Demographics

A community-based recruitment strategy began with six professional, native KSL interpreters (CODAs). They assisted in compiling a corpus of 173 common KSL phrases and sentences covering various communicative functions and recorded initial template videos using Logitech C920 Pro webcams. Through their networks, 44 additional contributors were recruited, forming a total group of 50 participants. This cohort exhibits significant demographic diversity: age range 8-57 years, 18 males and 32 females, varied hearing statuses (32 deaf, 6 hard of hearing, 9 hearing CODA, 3 hearing SODA), diverse KSL acquisition histories (most from birth [Lu et al., 2016]), and representation from different regions of Kazakhstan (Figure 3.2). The inclusion of participants across such a wide age range, including minors, necessitated careful ethical considerations but provides unique data for studying age-related variation in signing, a factor often missing in SL datasets. Although the notion of "native signer" can be complex [Zorzi et al., 2021], all contributors use sign language daily. Participants provided informed consent following ethical approval (forms translated to KSL), received compensation, and agreed to data sharing (details adapted from Allen (2015) [Allen, 2015]). Ensuring ethical procedures, including accessible consent and fair compensation, is particularly important when working with potentially vulnerable communities. Data anonymization steps were implemented during processing to protect participant privacy.

Figure 3.2: K-RSL: FluentSigners-50. Demographics of participants



3.2.2 Data Acquisition Protocol

An instruction video demonstrated the 173 target sentences. Participants recorded five repetitions of each sentence using their own devices (smartphones/webcams) in their typical environments, after viewing the corresponding example. This distributed, unsupervised method maximized environmental realism but allowed for natural

linguistic variations as participants did not always perfectly replicate the templates. While using predefined sentences based on templates ensures coverage of specific vocabulary and structures, it may capture less spontaneous signing compared to methods analyzing unprompted conversations or narratives; this represents a trade-off in the data collection design. The use of varied personal devices also led to heterogeneity in video quality and format (Figure 3.3). The filming process averaged about 3.5 hours per contributor.

Figure 3.3: K-RSL: FluentSigners-50. Percentage of videos per resolution scale

Table 3.1: K-RSL: FluentSigners-50 dataset statistics

Video resolution	Range
Number of Signers	50
Repetitions	5
Number of sentences	173
Video duration (seconds)	2∼11
Body joints	Upper-body involved
Mean number of signs per sentence/phrase	4
Vocabulary size	278
Total number of videos	43250
Total number of hours	43.9 (∼150 raw)

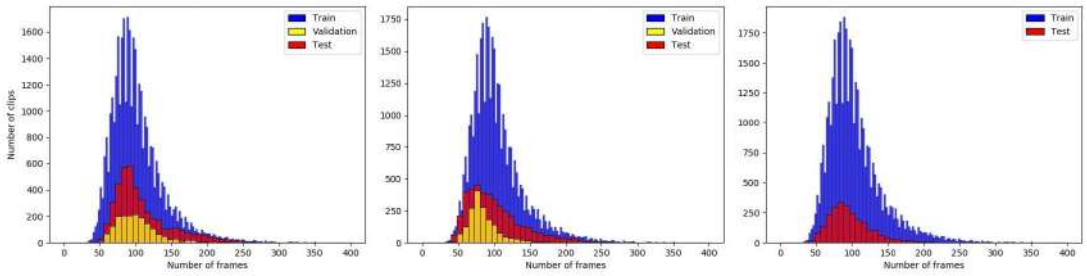
3.2.3 Data Processing and Curation

The initial collection yielded over 150 hours of raw video footage. This large volume required a rigorous process of manual validation and temporal segmentation (trimming) to isolate the relevant sentence performances. This resulted in the final dataset comprising 43,250 labeled video clips totaling 43.9 hours. The sheer volume of data generated through crowdsourcing underscores the critical need for efficient validation and annotation workflows, reinforcing the motivation for the semi-automatic tools developed in Chapter 6. Key statistics summarizing the dataset are provided in Table 3.1.

3.3 Linguistic Context: KSL

The language captured in FluentSigners-50 is Kazakh Sign Language, the primary sign language used within Kazakhstan. KSL is closely related to Russian Sign Language, and observational evidence suggests considerable lexical similarity and likely mutual intelligibility, although formal comparative studies are limited [Kimmelman et al., 2020]. The dataset includes varied sentence types common in daily interaction: statements, questions (polar and wh-), and requests. This mix of sentence types provides valuable data for studying linguistic features.

Figure 3.4: Distribution of the number of frames over sentence-level clips in train, val and test sets for each split: Split 1 (left), Split 2 (middle), Split 3 (right).



3.4 Evaluation Framework: Proposed Data Splits

To promote consistent evaluation and allow for targeted analysis of model generalization across different challenges inherent in real-world SLP, we propose three specific partitioning schemes for dividing the FluentSigners-50 dataset into training, validation, and test sets, rather than using standard random splits. Defining these explicit splits is crucial for benchmark datasets, as it allows researchers to directly compare the performance of different models under identical conditions, fostering reproducible research. These splits are deliberately constructed to assess model robustness against signer variability, age differences, and novel linguistic contexts. The distribution of video lengths (measured in frames) across the sets for each split configuration is illustrated in Figure 3.4. For rigorous evaluation on Splits 1 and 3, we recommend employing a 5-fold cross-validation procedure; this involves training and evaluating the model five times on different subsets of the data, providing more reliable average performance metrics and an indication of result stability. Due to the specific age-based stratification in Split 2, creating five balanced folds might be problematic, thus evaluation based on a single predefined train/validation/test partition may be more appropriate for this specific split.

3.4.1 Split 1: Signer Independence

This split evaluates generalization to unseen signers, crucial for practical systems. It addresses natural variations in appearance and signing style (phonetic, phonological, lexical, syntactic). Training data uses 40 signers; Validation uses 5 different signers; Testing uses 5 further distinct signers. Stratification ensures diverse signer backgrounds (CODA/non-CODA) in evaluation sets. Resulting set sizes: 34,600 (train), 4,325 (val), 4,325 (test).

3.4.2 Split 2: Age Independence

This split assesses generalization from adult to child signers (ages 8-18). Child signers (defined as 9-18 years old for this split) are excluded from training and appear only in validation/testing sets. Testing generalization across age groups is important for developing inclusive SLP systems, although it remains an under-explored area in the literature. This split also incorporates a device mismatch challenge (train: webcam/mobile vs. test: mobile only).

3.4.3 Split 3: Unseen Sentences and Signer Independence

This split tests generalization to novel linguistic contexts. It evaluates recognition of familiar signs within different sequential orders or syntactic structures than seen during training, leveraging dataset variability. This split directly probes the model's ability to handle co-articulation effects and semantic understanding beyond simple sign spotting. 163 unique sentences are used for training, with 10 distinct sentences held out for testing. Test set signers are also unseen during training, combining linguistic context and signer independence challenges.

3.5 Chapter Conclusion

In conclusion, the FluentSigners-50 dataset, generated through a community-centric crowdsourcing methodology, represents a significant contribution to KSL resources. It provides a large-scale corpus of continuous signing with unprecedented signer and environmental diversity, directly addressing key limitations of prior datasets. The carefully designed evaluation splits, detailed above, further enhance its value as a benchmark for developing and rigorously assessing robust SLP models for KSL. This dataset forms the empirical foundation for much of the analysis and evaluation presented in subsequent chapters.

Chapter 4

KSL-OnlineSchool: Leveraging Online Resources for Large-Scale Corpus Creation

4.1 Introduction

This chapter introduces the second large-scale KSL dataset developed within this thesis: KSL-OnlineSchool. While Chapter 3 detailed the community-sourced FluentSigners-50 dataset focused on capturing signer variability in controlled tasks, this chapter presents a complementary approach targeting large vocabulary size and extensive continuous signing data by leveraging existing online resources. The primary objective behind creating KSL-OnlineSchool was to address the significant limitations of small vocabulary sizes and the lack of naturally occurring continuous signing data prevalent in many existing SLP corpora (as discussed in Chapter 2). This dataset, therefore, represents a different strategy for overcoming data scarcity, capitalizing on publicly broadcast interpreted content rather than direct elicitation.

The KSL-OnlineSchool dataset consists primarily of recordings of synchronous sign language interpretation accompanying online school lessons broadcast nationally in Kazakhstan. These lessons span diverse academic subjects and cater to grades 1 through 11. The overall data collection methodology, from broadcast recording to annotation, is depicted in Figure 4.1.

The opportunity for this large-scale data collection emerged during the rapid shift to online schooling in Kazakhstan in 2020 due to the COVID-19 pandemic. The El-arna TV channel broadcast daily lessons with simultaneous KSL interpretation, providing a unique source of extensive, naturally occurring interpreted sign language data across varied topics. Over nine months, these broadcasts were systematically recorded. Initial processing yielded a substantial corpus, which, after cleaning and organization, resulted in 890 hours of video material featuring 7 different KSL interpreters across 4,547 distinct lessons. To manage the annotation of this large dataset, a custom web-based tool, "Surdobot," was developed (Section 4.4). To date, annotation efforts have yielded partial sign language glosses for approximately 325 hours of video (over 39,000 segments) and automatic speech recognition (ASR) transcripts for the spoken content of 4,009 lessons.

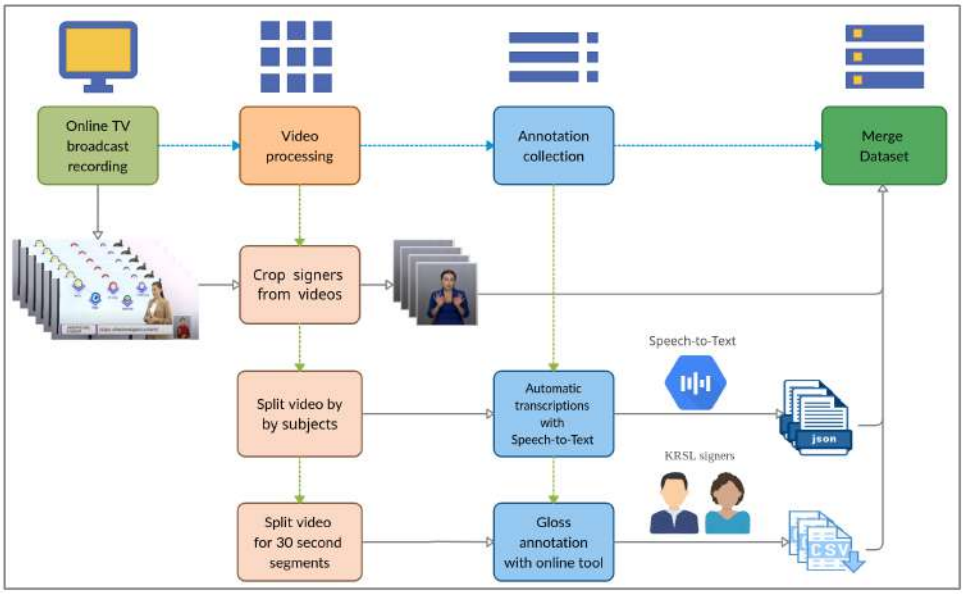


Figure 4.1: KSL: OnlineSchool. Dataset collection methodology

The key contributions associated with the KSL-OnlineSchool dataset are:

- The release of the large-scale KSL video corpus (4,547 lessons, 890 hours, 7 interpreters), categorized by subject and grade. Its scale significantly surpasses most existing sign language corpora, particularly for KSL.
- Spoken language transcripts for 4,009 lessons (approx. 1 million sentences), obtained via ASR.
- Partial sign language gloss annotations for 325 hours (39,000 segments). The availability of parallel video, text transcripts, and partial glosses makes this dataset uniquely suited for research into multimodal SLP, translation models, and semi-supervised learning approaches.

4.2 Dataset Source and Collection Process

The source material consisted of daily video lessons broadcast by the El-arna national TV channel during the September 2020 to May 2021 academic year. These lessons, covering subjects for grades 1-11 (see Table 4.1 for subject list), were aired live with KSL interpretation between 9 AM and 5 PM daily, each lesson averaging 10-12 minutes. Continuous screen recording was employed over the 9-month period to capture these broadcasts.

Post-recording processing involved several steps. Custom scripts using the OpenCV library were used to automatically crop the video region containing the sign language

interpreter (resulting resolution: 230x264 pixels). The continuous recordings were then segmented into individual lessons corresponding to the broadcast schedule and categorized by subject and grade. Lessons originally in English (without KSL interpretation) were excluded. This process yielded the final corpus of 4,547 videos totaling 890 hours. Table 4.2 shows the distribution of lessons across school grades.

	Subject name	**Videos**
1	Literacy education	76
2	Math	602
3	Second language	794
4	Natural science	129
5	World science	91
6	Digital literacy	43
7	History	357
8	Kazakh language	538
9	World history	216
10	Algebra	298
11	Informatics	178
12	Geography	248
13	Chemistry	193
14	Literature	100
15	Geometry	185
16	Physics	263
17	Biology	236
Total		**4547**

Table 4.1: KSL: OnlineSchool. List of subjects in dataset.

Grade	**Videos**	**Transcripts**
1	249	205
2	318	257
3	334	288
4	325	282
5	366	349
6	344	292
7	484	441
8	513	457
9	584	522
10	518	468
11	506	448
Total	**4547**	**4009**

Table 4.2: KSL: OnlineSchool. Number of lessons by grade in dataset.

4.3 Dataset Overview and Characteristics

The complete KSL-OnlineSchool dataset comprises 890 hours of video across 4,547 lessons interpreted by 7 signers. Accompanying annotations include ASR-generated text transcripts for 4,009 lessons and manual sign glosses for 325 hours (39,000 segments). A key characteristic of this dataset is its large vocabulary, estimated at over 20,000 unique words (based on transcript analysis), reflecting the diverse academic subject matter covered.

Figure 4.2 shows the distribution of transcript lengths per lesson, averaging around 1,000 words, with shorter lessons common for lower grades. Figure 4.3 shows the distribution of gloss counts per 30-second annotated clip, averaging around 30 glosses. The annotated subset contains over 1,000 unique gloss types with significant repetition, suitable for training recognition models. Annotation efforts are ongoing to cover the entire dataset.

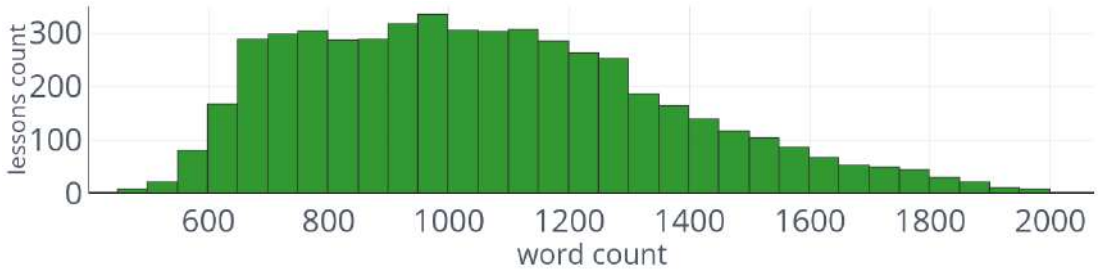


Figure 4.2: KSL: OnlineSchool. Full text transcripts length for each lesson

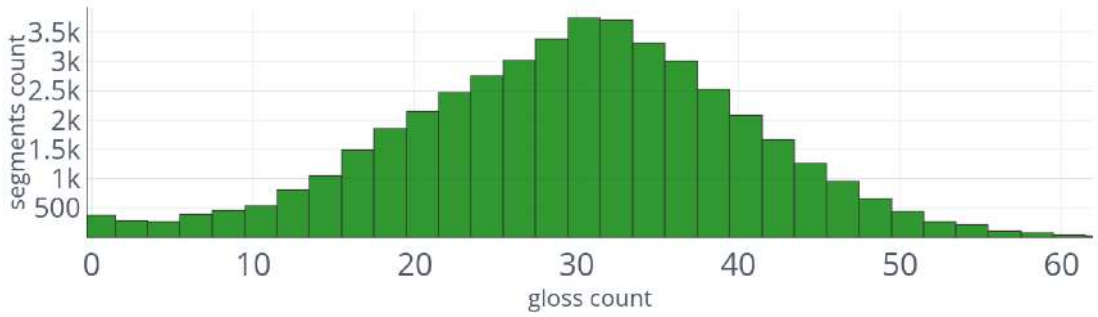


Figure 4.3: KSL: OnlineSchool. Gloss annotation length for each 30 second clip

Analysis of frequent terms reveals semantic overlap between the ASR transcripts and manual gloss annotations (Figures 4.4 and 4.5), with common words like "this," "equal," "correct," etc., appearing in both. Similar, though less extensive, overlap is observed in frequent 2-gram sequences (Figures 4.6 and 4.7), such as "lesson today" and "correct answer." This correlation suggests the potential utility of ASR transcripts as a source of weak supervision or supplementary information for sign language understanding tasks trained on this data.

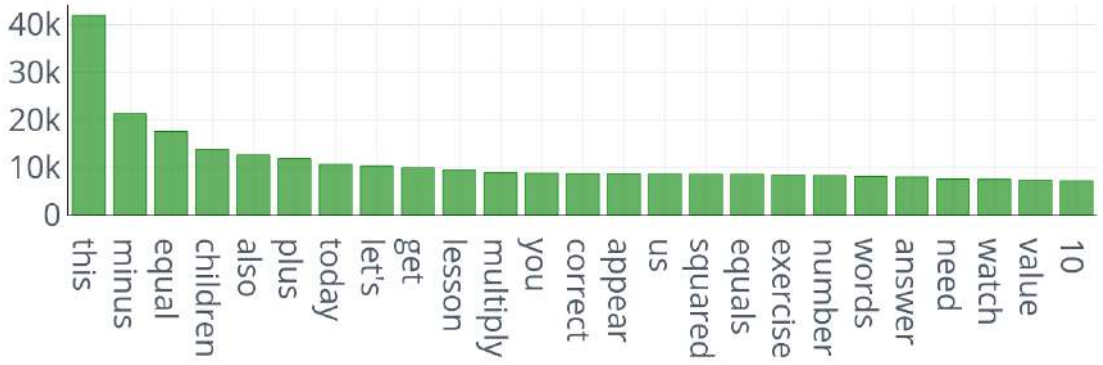


Figure 4.4: KSL: OnlineSchool. Top words in full text transcripts

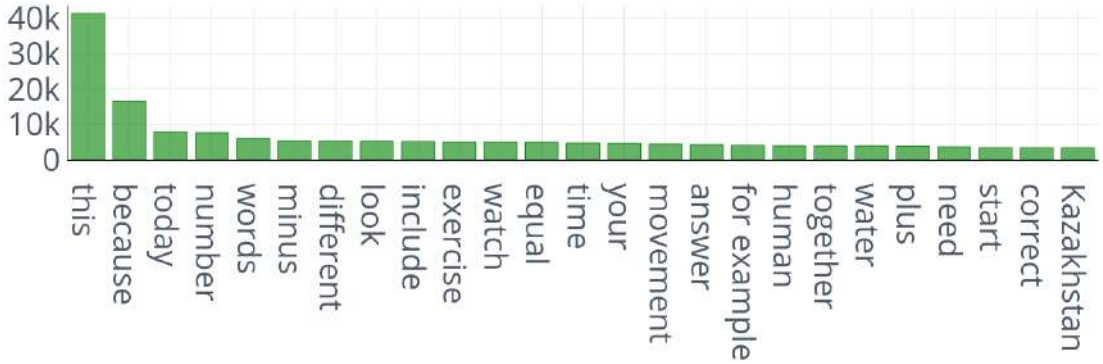


Figure 4.5: KSL: OnlineSchool. Top glosses in gloss annotations

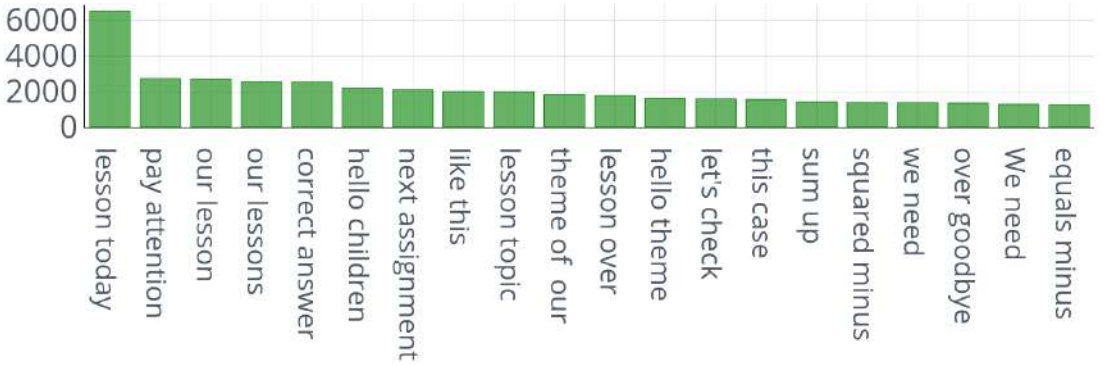


Figure 4.6: KSL: OnlineSchool. Top 2-grams for text transcriptions

4.4 Annotation Methodology

Two main types of annotations were generated: full text transcriptions (in Russian) and sign language glosses (of the KSL interpretation).

For text transcription, initial attempts using the Kaldi [Povey et al., 2011] proved inconvenient due to the required audio extraction and segmentation steps. A more practical method involved uploading the videos to YouTube and utilizing its automatic captioning service. A custom script using the YouTube API facilitated downloading these time-aligned transcriptions for 4,009 lessons. While convenient, the accuracy of

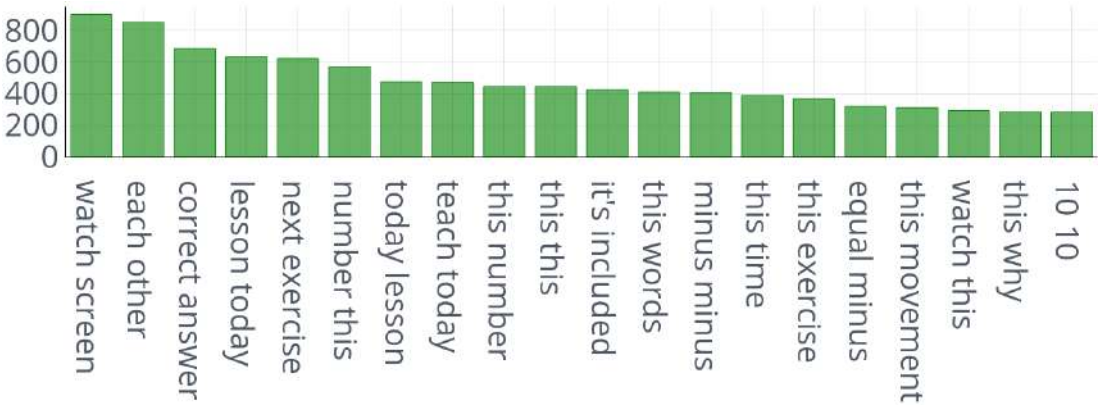


Figure 4.7: KSL: OnlineSchool. Top 2-grams for gloss annotations

these ASR transcripts depends on YouTube's algorithms and may vary, representing a potential source of noise if used directly as ground truth for translation tasks.

For gloss annotation, the videos were segmented into 30-second clips to create manageable tasks for annotators. A dedicated web-based tool, "Surdobot" (https:surdobot.kz, Figure 4.8), was developed to manage this large-scale annotation effort involving 8 annotators (5 deaf, 3 professional KSL translators). The tool presented random clips to annotators, provided playback controls, allowed text input for glosses, tracked progress, and enabled review/editing. This facilitated the annotation of 39,000 clips (325 hours) thus far. The development of Surdobot demonstrates a practical solution for distributed, large-scale sign language annotation projects, related to the more general annotation tool framework discussed in Chapter 6.

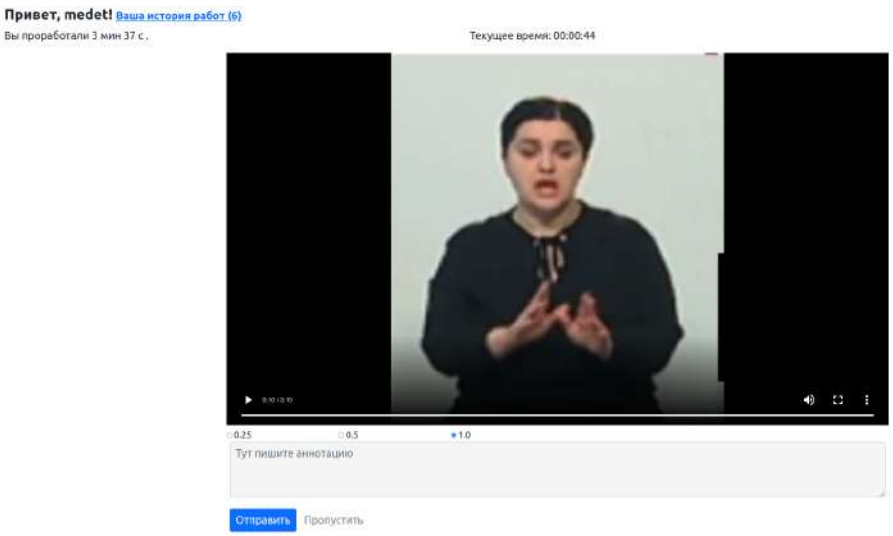


Figure 4.8: Surdobot annotation tool's user interface

4.5 Chapter Conclusion

A crucial characteristic of sign language research is the large disparity in vocabulary size between typical sign language datasets and spoken language corpora. As pointed out rightly by Bragg et al. (2019), one of the main challenges of SLP is related to significant shortcomings of public sign language datasets that limit the power and generalizability of recognition systems trained in them. A standard sign language corpus may include roughly 1,500 distinct signs, whereas a spoken language corpus can encompass 300,000 words. This gap has practical consequences for both data collection and translation models.

During data collection, a smaller common lexicon means that elicitation prompts must be carefully designed. Many spoken sentences contain words without a direct, single-sign equivalent. This narrows the conceptual coverage of controlled collection and can introduce inconsistencies when signers resort to descriptive phrases or fingerspelling for more nuanced terms.

For recognition and translation tasks, a model must learn to map a single sign from a small vocabulary to potentially many different words in a large vocabulary, relying entirely on context to select the correct one. With limited training data, a model trained on limited data will inevitably learn to produce a generic, "safe" translation, failing to capture the specific meaning intended by the signer. This is why large, context-rich resources such as our KSL-OnlineSchool corpus are vital. By providing nearly 900 hours of continuous signing across diverse topics, we provide the contextual variety needed for models to learn how to resolve these challenging mappings between a compact sign lexicon and a much larger spoken vocabulary.

The KSL-OnlineSchool dataset represents a second major contribution towards addressing data scarcity for KSL processing. By leveraging readily available online interpreted educational content, this effort yielded a very large corpus (890 hours) characterized by an extensive vocabulary derived from diverse academic subjects. While the interpretation setting differs from the community crowdsourcing of FluentSigners-50, this dataset provides unique value through its scale and the availability of parallel text transcripts alongside partial gloss annotations. Together, these two datasets offer complementary resources for advancing KSL recognition, translation, and linguistic analysis.

Part II

Methods and tools for sign language processing

BLANK

Chapter 5

Sign language representation

Signs in sign languages are understood to be composed of phonological components combined according to certain linguistic rules [Sandler & Lillo-Martin, 2006]. Initially, sign language linguistics identified three main components: handshape, location on the body, and movement. Later, orientation and non-manual components were recognized as additional essential elements. This chapter discusses various features that can be extracted from sign language videos for computational purposes. Feature extraction serves as the initial component within an SLR system; its goal is to obtain useful information from the input sign language video. This extracted information is subsequently utilized by the system's machine learning component to recognize the sign or signs present in the video. Accordingly, this section focuses on discussing the different features extractable from sign language video data.

Many different approaches exist for feature extraction. Some methods involve using specialized hardware to directly acquire the features. Other techniques rely on applying image processing methods to the video frames. Some approaches utilize 3D models of the human body to estimate relevant features. Often, methods combine techniques, for example, using specialized hardware in conjunction with image processing, or integrating image processing techniques with 3D human body models.

Sign language recognition systems may extract both manual and non-manual features from a sign language video, as these types of features work together to convey meaning. The non-manual features of sign language include elements like facial expressions, eye gaze direction, and mouth movements; the use of these features is an important aspect of sign communication. The manual features primarily include hand shapes, hand movements, and hand positions. Hand movements in sign language can be broadly categorized into: (1) one-handed movements, where the non-dominant hand typically remains stationary while the dominant hand moves; and (2) two-handed movements, where both hands actively participate in conveying meaning. Two-handed movements can be further classified as either symmetric (both hands move similarly) or asymmetric (the hands move in different manners). Hand position refers to the location of the hands relative to the signer's body or within the signing space; hands might remain stationary in some signs, while others involve distinct movement trajectories. Understanding the roles and types of both manual and non-manual features is foundational for developing comprehensive sign language representations.

5.1 Manual components representation

By deploying various computer vision approaches, we aim to investigate the automation of creating a handshape inventory, a task traditionally considered highly time-consuming for linguists. Numerous researchers have previously worked on establishing handshape inventories (refer, e.g., to [Van der Kooij, 2002, Nyst, 2007, Tsay & Myers, 2009, Kubuş, 2008, Klezovich, 2019]). In almost all of this prior work, handshapes were extracted from video data and subsequently annotated through manual effort [Klezovich, 2019].

Klezovich [Klezovich, 2019] proposed the first handshape inventory for RSL utilizing a semi-automatic approach. This involved extracting "hold-stills" (segments where handshape is stable) within sign videos based on an image overlay technique. The rationale is that handshapes are generally articulated most clearly during holds, whereas transitional movements typically do not contain distinct target handshapes. Klezovich suggested that extracting only these hold segments for subsequent manual labeling could significantly expedite the process of creating handshape inventories [Klezovich, 2019]. This highlights a potential pathway for combining computational analysis with linguistic expertise to tackle laborious tasks.

We test an automatic approach for generating a handshape inventory. First, we attempt unsupervised learning and demonstrate that the results are unsatisfactory, as this method cannot distinguish handshape categories separately from variations in orientation and location. Second, we manually label a training dataset based on HamNoSys handshape descriptions [Hanke, 2004] and evaluate the utility of semi-supervised and supervised learning techniques on new data.

5.1.1 Data pre-processing

5.1.1.1 Dataset

The dataset was created by downloading videos from the Spreadthesign online dictionary (www.spreadthesign.com). We downloaded a total of 14,875 RSL videos from this website. The videos typically contain demonstrations of either a single sign or a phrase consisting of several signs. Using a large online dictionary provides access to a wide vocabulary but also introduces variability in recording quality and signer performance inherent in such resources.

In our process, blurry images were removed using the variance of the Laplacian method with an empirically set threshold of 350. Images with variance below this threshold were considered blurry and discarded. Selecting an optimal threshold typically requires some experimentation depending on the specific dataset. This filtering step reduced the total number of initial images from 141,135 down to 18,226 cropped images containing hands.

5.1.1.2 Hand extraction

Hand detection can be viewed as a sub-task of object detection and segmentation. Hands present challenges due to their variability in shape, orientation, and configuration. Standard object detection frameworks like Mask R-CNN [He et al., 2017] and CenterNet [Duan et al., 2019] are potentially applicable.

However, issues like occlusion and motion blur can decrease the accuracy of general models. For these reasons, this work employed a novel CNN architecture named Hand-CNN [Narasimhaswamy et al., 2019]. Based on Mask R-CNN [He et al., 2017], it incorporates an attention module using contextual cues (feature similarity, spatial relationships) to improve detection robustness, particularly for occlusions and motion blur. Hand-CNN outputs segmentation masks, bounding boxes, and orientations for detected hands. We used the predicted bounding boxes to crop hand images with two padding settings: 0-pixel (tight crop) and 20-pixel (including context). Consequently, the first group contains tightly cropped hand images, while the second group includes the hands along with some surrounding area reflecting their position relative to the body.

5.1.1.3 Image pre-processing

For images with 0-pixel padding, we applied Histogram of Oriented Gradients (HOG) descriptors [Dalal & Triggs, 2005]. HOG features are common in computer vision for object detection, based on distributions of intensity gradients or edge directions. An image is divided into small regions (cells); a histogram of gradient directions is calculated for each cell; concatenations of these histograms form the descriptor. We used the scikit-image library's 'feature' module [van der Walt et al., 2014] with parameters: orientations=9, pixels_per_cell=(10,10), cells_per_block=(2,2), and L1 block normalization. Images were first converted to grayscale and resized to 128x128 pixels.

For images with 20-pixel padding, we utilized the AlexNet CNN architecture [Krizhevsky et al., 2012], often used as a baseline. We employed only the first five convolutional layers (96, 256, 384, 384, 256 filters) as a fixed feature extractor, as our goal was clustering, not classification. Images were resized to 224x224 pixels before feature extraction. The resulting CNN features underwent PCA reduction to 256 dimensions prior to clustering. Comparing HOG features (capturing shape) with CNN features (potentially context-aware) helps assess representation suitability for handshape analysis.

5.1.2 Unsupervised Methodology

5.1.2.1 Clustering

We utilized the classical k-means clustering algorithm. An implementation by [Johnson et al., 2019] was applied to ConvNet features, while the scikit-learn [Pedregosa et al., 2011] implementation was used for HOG features. Each training ran for 20 iterations with random initialization.

The number of clusters (k) was determined experimentally. As handshape orientation seemed to influence clustering, we increased k (trying 100, 150, 200, 300, 400) aiming to force differentiation between shape and orientation.

5.1.2.2 Analysis and evaluation

We used two metrics: the Silhouette Coefficient and Normalized Mutual Information (NMI). The Silhouette Coefficient [Rousseeuw, 1987], used when ground truth labels are unknown, assesses how similar an item is to its own cluster versus others (higher positive values are better, range -1 to +1). Maximum observed Silhouette scores (Figure 5.1) were low (just over 0.12 for AlexNet features, k=100, 15 epochs), indicating overlapping clusters. NMI scores (Figure 5.2), measuring agreement between predicted and actual labels, reached high values (0.9) after 15 epochs across different cluster numbers.

The low Silhouette scores might result from high similarity between hand image descriptors, making differentiation difficult. The high NMI suggests stable cluster assignments over epochs relative to some reference. Increasing cluster density likely requires additional image pre-processing.

5.1.2.3 Results

Figure 5.3 provides insights into the unsupervised clustering results. It is clear the algorithm distinguishes classes based not only on handshape but also orientation (for 0-pixel padding images) and localization (for 20-pixel padding images). This is a disadvantage for the linguistic task of creating a phonemic handshape inventory.

Despite shortcomings, the method yields linguistically interesting results. Expected unmarked handshapes (A, 5, 1) appear frequently as labels for multiple clusters. Thus, although the classification isn't purely linguistically relevant, the effect of markedness (frequency and visual salience) is visible.

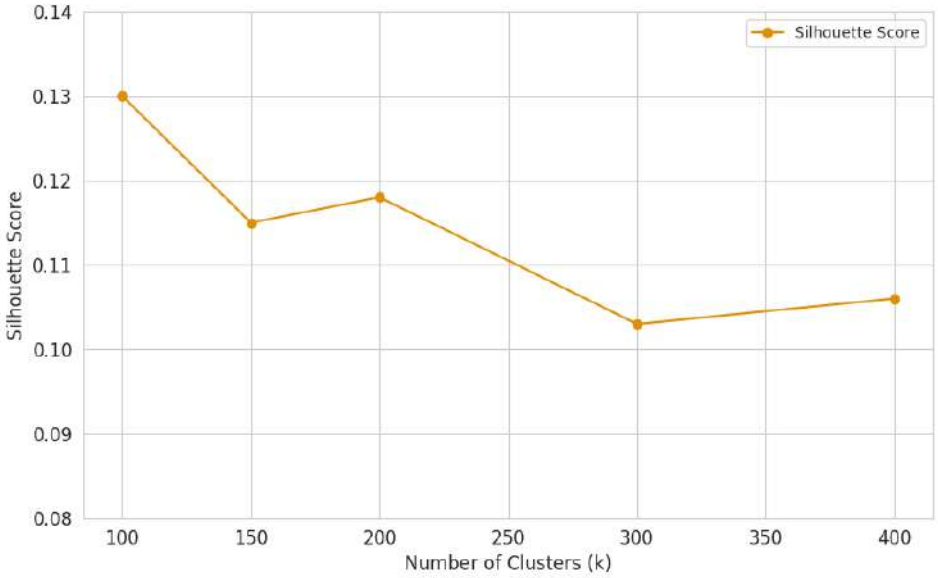


Figure 5.1: The mean Silhouette Coefficient scores for the clustering based on number of clusters for AlexNet features

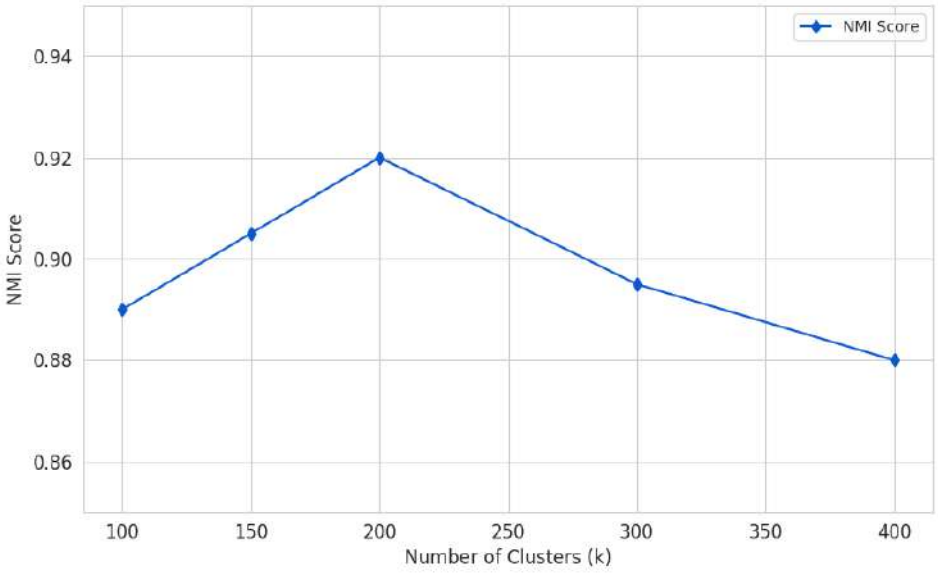


Figure 5.2: The degree of similarity between predicted and actual labels, measured using Normalized Mutual Information (NMI) based on number of clusters

Figure 5.3: Visualization of the 135 most representative clusters derived using HOG features

5.1.3 Supervised Methodology

5.1.3.1 Dataset

Given the limitations of unsupervised approaches, we turned to supervised methods. The initial dataset derived from HOG clustering (140 clusters, 18,226 images) was manually cleaned by four undergraduate students. They visually scanned each cluster, removed incorrectly assigned handshapes, and merged folders representing the same handshape with different orientations. This resulted in 35 handshape classes and a large 'junk' folder, yielding a final dataset of 7,346 cropped images (0-pixel padding) across the 35 classes.

Classes were created based on intuitive visual similarity by linguistically naive annotators. Post-factum analysis suggests the classification is linguistically reasonable as an approximation of a phonological inventory, distinguishing by selected fingers, spreading, and finger position (straight, bent, curved). Thumb position (opposed vs. other) was a distinguishing feature, but non-selected finger position was not considered. This approximates features relevant to inventories in other sign languages and can be used for RSL. A finer phonetic classification (distinguishing exact thumb position, non-selected fingers) might use HamNoSys [Hanke, 2004] but likely requires more data than available here.

The manually labeled subset was divided into training (6,430 images) and validation (916 images). Figure 5.4 shows the class distribution, confirming expected frequency properties: the most frequent handshapes are the expected unmarked ones (A, 5, 1, B),

comprising 48% of classifiable handshapes (excluding the two-handed sign category noted in the footnote).

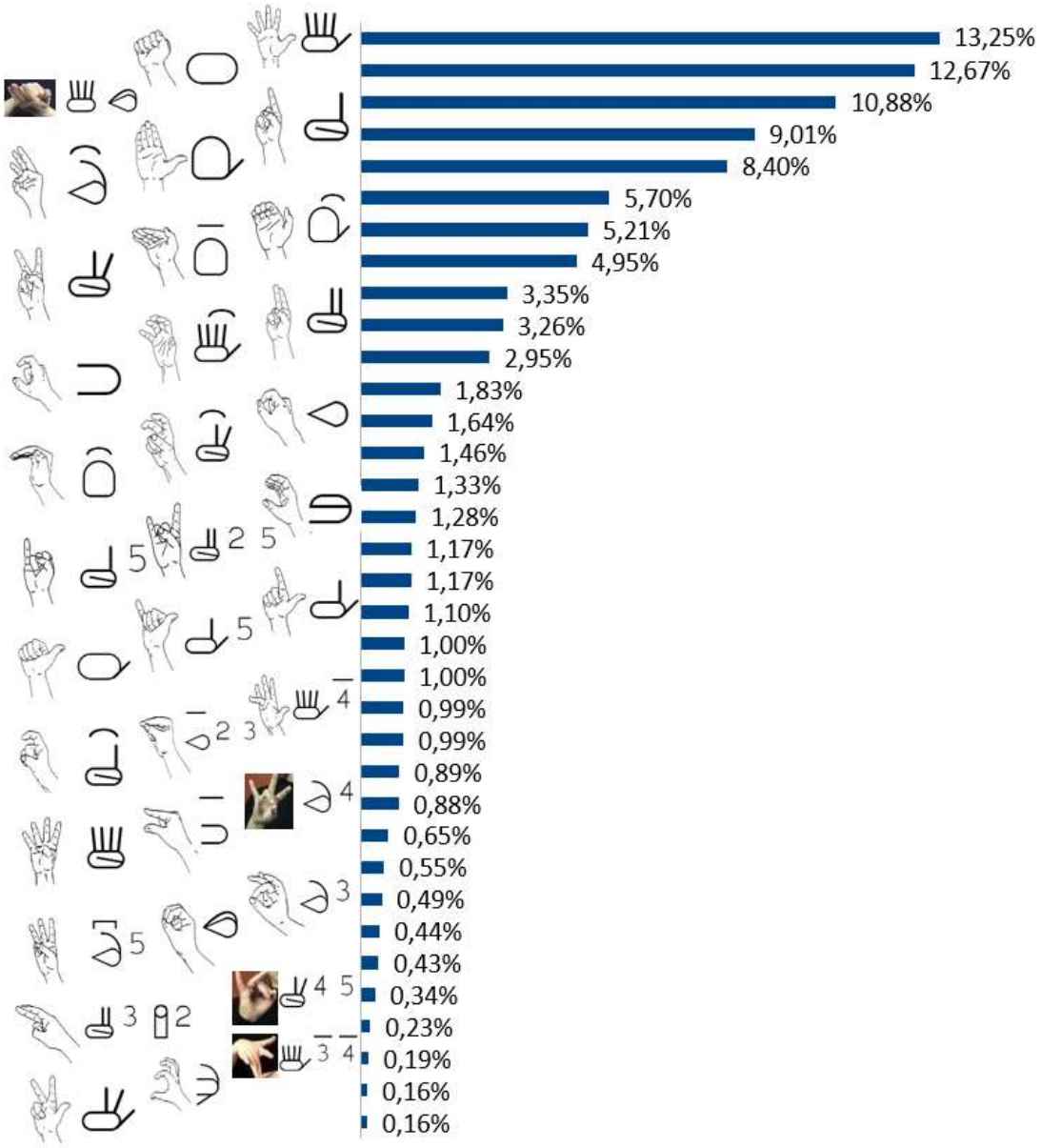


Figure 5.4: Handshape classes count.

5.1.3.2 CNN models and transfer learning

Training entire Convolutional Neural Networks (CNN) from scratch requires extensive resources and data. Transfer learning, using pre-trained CNN models (e.g., on ImageNet), is common. Two standard techniques are: fine-tuning (initializing with pre-trained weights, training all layers) and fixed feature extraction (freezing most weights, training only a new final layer). We implemented networks using PyTorch (v1.4.0) [Paszke et al., 2019c], with ResNet-18 [He et al., 2016b] as a pre-trained model.

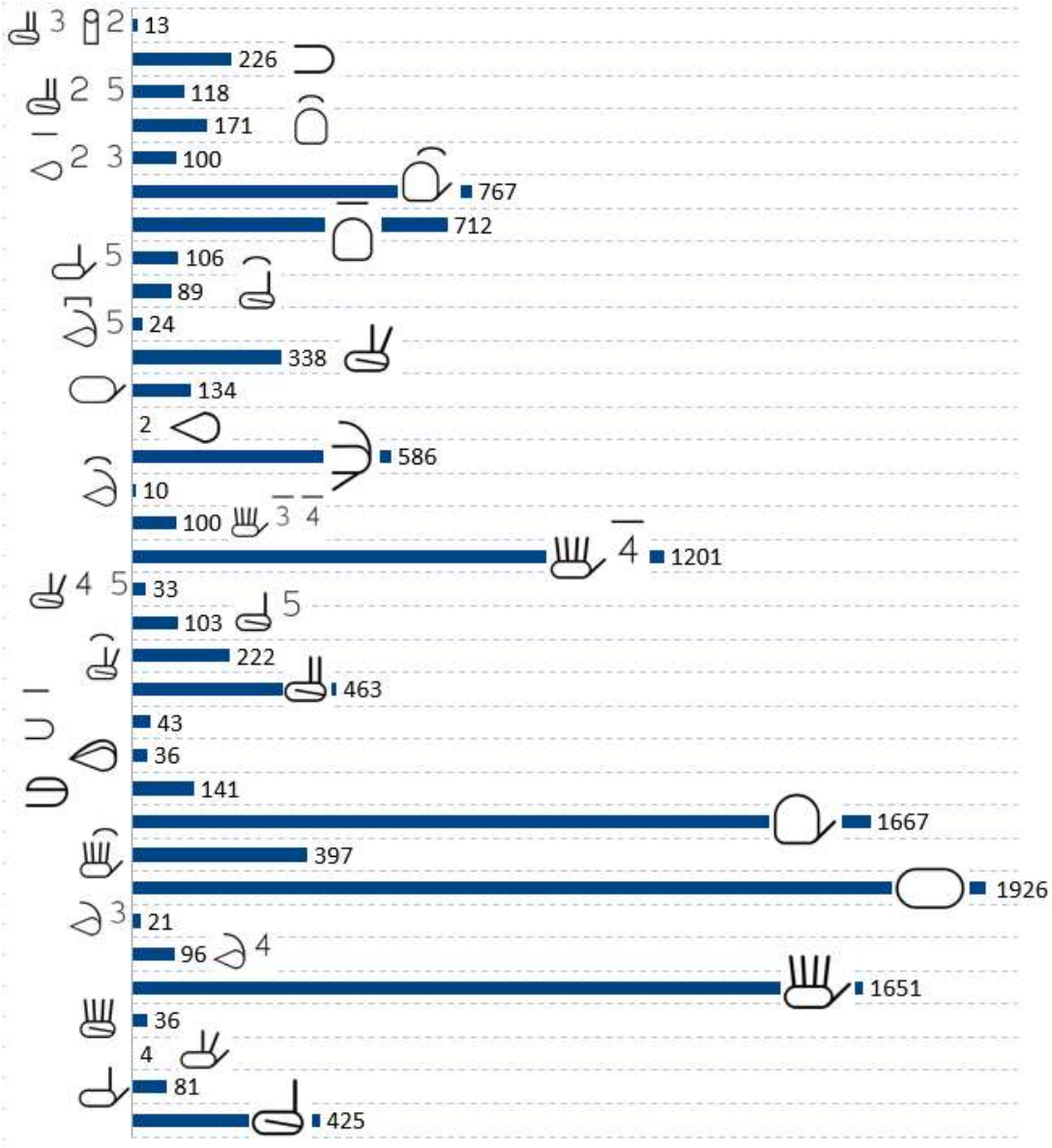


Figure 5.5: Handshape classes count using classifier.

5.1.3.3 Results

We trained models using both transfer learning approaches for 200 epochs. Fixed feature extraction achieved a best validation accuracy of 43.2%, while fine-tuning all layers demonstrated a superior accuracy of 67%. The fine-tuned model was, therefore, used for further improvements. Adding data augmentation (random rotation; random changes in brightness, contrast, and saturation with $p = 0.25$) increased the accuracy of the best model to 74. 5% after 200 epochs.

However, while overall accuracy is a useful starting point, it can be a misleading metric when dealing with an imbalanced dataset, as is the case with our handshape classes (see Figure 5.4). A model could achieve high accuracy simply by correctly classifying the most frequent classes while performing poorly on rarer handshapes. To

conduct a more robust evaluation that accounts for this class imbalance, we therefore also analyzed performance using precision, recall, and the F1-score.

Precision measures the proportion of correct predictions among all predictions for a specific class (e.g., of all the times the model predicted 'Handshape A', how often was it correct?).

Recall measures the proportion of a class's actual instances that were correctly identified by the model (e.g., of all the real 'Handshape A' images, how many did the model find?).

F1-score is the harmonic mean of precision and recall, providing a single, balanced metric that is particularly useful for evaluating performance on imbalanced data.

Table 5.1 presents these metrics for a representative sample of high, mid, and low-frequency handshape classes. The performance varies significantly across the classes, confirming the effect of the imbalanced distribution. The model performs very well on common, unmarked handshapes but struggles with rarer, more complex configurations for which it has seen fewer training examples. The macro-averaged F1-score, which treats all classes equally regardless of their sample size, provides a more critical measure of the model's overall ability to classify the entire handshape inventory.

Table 5.1: Per-Class Performance of the Fine-Tuned Handshape Classifier on the Validation Set.

Handshape Class	Frequency	Precision	Recall	F1-Score
Class 5	High	0.92	0.95	0.93
Class A	High	0.88	0.91	0.89
Class 1	High	0.90	0.86	0.88
Class 8	Medium	0.75	0.68	0.71
Class 9	Medium	0.71	0.74	0.72
Class 25	Low	0.51	0.44	0.47
Class 30	Low	0.49	0.40	0.44
Class 35	Low	0.42	0.38	0.40
Macro-Average	-	**0.69**	**0.67**	**0.68**
Weighted-Average	-	**0.75**	**0.75**	**0.74**

This detailed evaluation shows that while the model is effective (Weighted F1-score of 0.74), its performance is not uniform across all handshapes. This trained model was then used to predict labels for all 18,226 handshape images. Applying a prediction probability threshold of 0.7 to remove low-confidence classifications resulted in 12,042 classified samples. Figure 5.5 shows the predicted class distribution.

5.1.4 Discussion

5.1.4.1 Insights from unsupervised and supervised approaches

This study indicates that the unsupervised approach explored seems unpromising for automated handshape recognition aimed at linguistic analysis. The primary issue is that the handshape category, while linguistically relevant, is not easily separated visually from orientation and location factors by this basic data-driven method. While unsupervised methods might be useful for initial data exploration, they appear insufficient for creating reliable phonological inventories without further constraints or feature engineering.

We demonstrated that an alternative approach involving a manual classification step can be quite effective for training a supervised model. However, manual classification introduces subjectivity and significant labor cost. This highlights a common trade-off in computational linguistics between fully automated methods and those requiring human intervention for accuracy.

Both approaches, nevertheless, offer linguistically relevant insights regarding unmarked handshapes. In the unsupervised results, unmarked shapes clearly dominated many clusters, reflecting their frequency and visual distinctiveness. In the supervised approach, analysis of the manually classified data confirmed the high frequency of unmarked shapes (A, 1, 5, B). Applying the trained classifier to the full dataset also showed A, B, 5 as the top 3 predicted classes. The lower-than-expected frequency for the '1' handshape might indicate misclassification with a visually similar marked shape, highlighting remaining challenges even for supervised models. Thus, both successful and less successful applications of machine learning confirm the importance of unmarked handshapes in RSL. Extending these computational approaches to other sign languages could be valuable for comparative studies.

5.1.4.2 Error Analysis and Model Diagnostics

While the quantitative metrics in the previous section provide a summary of the handshape classifier's performance, a qualitative analysis of its errors is essential for understanding its limitations and guiding future improvements. To this end, an error analysis was conducted by examining the confusion matrix and a sample of misclassified images from the validation set of the trained classifier.

The analysis revealed that the vast majority of errors were substitutions between handshape classes that are visually similar. The model is highly effective at distinguishing between broadly different categories (e.g., an open hand vs. a closed fist), but struggles with fine-grained distinctions. Table 5.2 presents several representative examples of these common confusions. These cases are specifically chosen to illustrate

the types of subtle articulatory differences, such as thumb position or finger curvature, that the model fails to reliably capture.

Table 5.2: Visual Examples of Handshape Classification Errors

True Handshape	Predicted Handshape	Potential Cause
		**Fine-Grained Feature:** The primary difference is the curve of index finger, a small local feature the model failed to capture.
		**Visual Similarity:** The core feature (extended fingers) is the same, with only the small distance between them differing.
		**Visual Noise:** Motion blur during the sign's movement phase can obscure the subtle knuckle bend, making the hand appear flat.

Three primary patterns emerge from this analysis:

1. Confusion Between Similar Classes: the model's primary weakness is in differentiating between handshapes that belong to the same family but differ in a minor articulatory detail. This suggests the model learns a general representation of a handshape but is less sensitive to the small, class-defining features.

2. Sensitivity to Fine-Grained Details: Many errors hinge on the model's inability to reliably capture very specific features, such as the exact position of the thumb against the fist or the degree of knuckle flexion.

3. Impact of Image Quality: Errors were more common in images with motion blur or non-optimal lighting from the source videos, where subtle features are obscured.

These insights lead directly to several proposals for future model improvements. To address the confusion between similar classes, targeted data augmentation could be employed, generating synthetic images that explicitly emphasize the subtle differences (e.g., varying thumb positions on a fist). Alternatively, a contrastive learning approach could be used during pre-training to force the model to learn a more discriminative feature space. To improve sensitivity to fine-grained details, future work could explore different model architectures, particularly those incorporating attention mechanisms that can learn to focus on the most informative regions of the hand, such as the fingertips and thumb.

5.2 Hand Configurations analysis

Further experiments detailed explored handshape analysis aspects including digit selection, common shapes resembling grasps, cross-linguistic usage comparison, and hierarchical clustering for exploring language relationships based on handshape frequencies for remaning Sign Languages in SpreadTheSign dictionary. This large-scale quantitative analysis complements traditional linguistic methods. Various statistical/visual methods were used. Analysis revealed patterns in digit selection and grouping, emphasizing consistency across languages. Key hypotheses regarding digit autonomy, frequency, and cultural influences were generally validated. Findings contribute to understanding SL linguistics and digit selection's role, providing insights into finger function and shared core handshapes for future research in SL studies.

5.2.1 Digit Selection in Signing

One Digit Selection: Calculating the percentage where one digit differs from the others showed ring (58.29%) and little (57.71%) fingers had highest independence, followed by middle (51.78%), thumb (48.04%), and index (45.66%) (Figure 5.6). This differs slightly from movement science findings (thumb most independent), possibly reflecting signing-specific patterns.

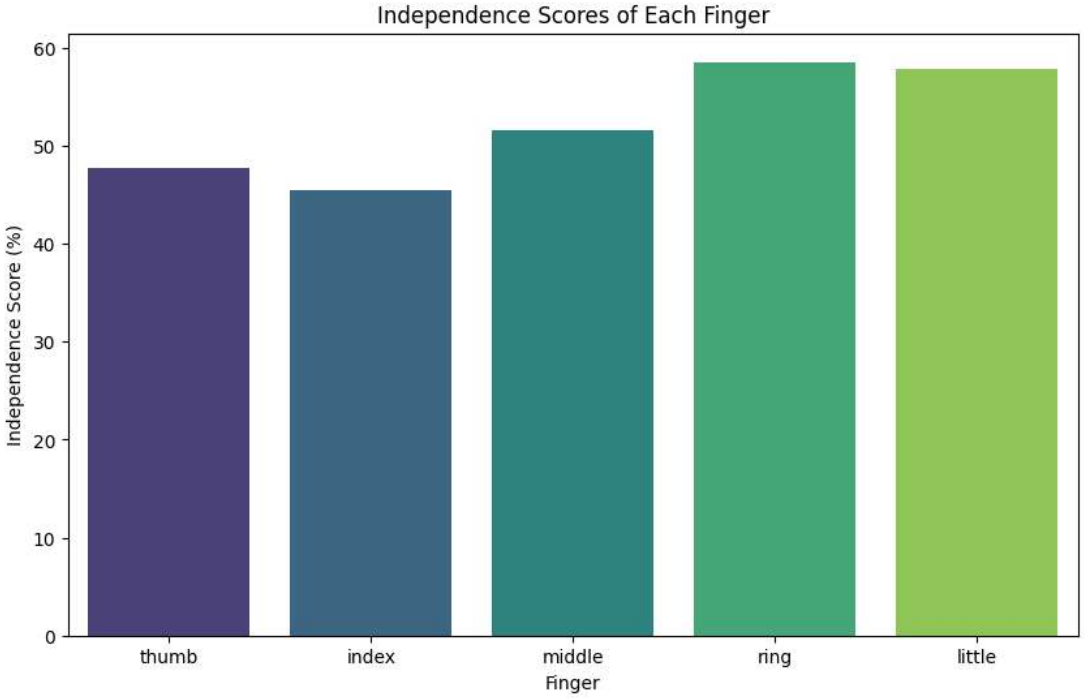


Figure 5.6: Independence Scores of each finger.

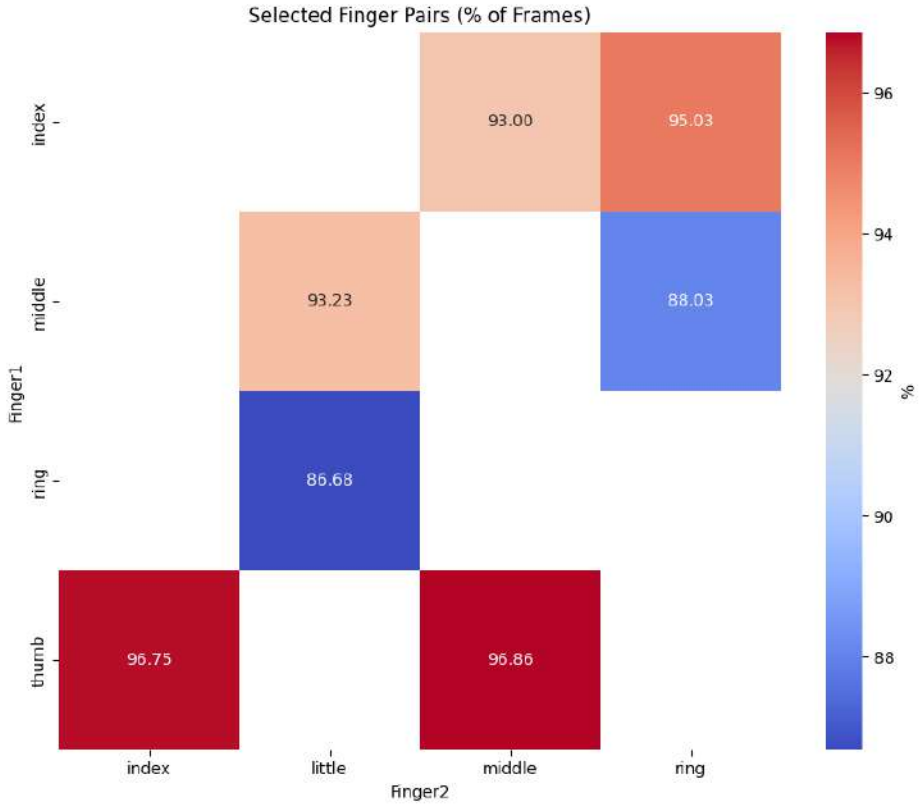


Figure 5.7: Pairs of Selected Fingers.

Digit Pairing: Analysis calculated frequency of pairs being co-selected (shaped differently) and pairs having identical shapes. Selected pairs analysis (Figure 5.7) confirmed neighbors co-select more often (thumb-middle 96.78%, thumb-index 96.60%, middle-ring 87.78%, ring-little 86.50%). Identical pairs analysis (Figure 5.8) showed this occurs less frequently overall, though highest for neighbors (middle-ring 40.16%, index-middle 19.70%), suggesting independent finger movement is common.

5.2.2 Cross-Linguistic Variation

This explored handshape usage variation across different sign languages using frequency calculations, rankings, visualization, identifying top 35 shared shapes, and correlation analysis. Insights showed consistently preferred shapes (potential universals) and variations.

Distribution of Handshape Frequencies: Analysis across 33 languages showed variation but also commonality (Figure 5.9). Shapes like "d," "dj," "dk," "ej" were common across diverse languages. Some languages (Chinese, US English, French) used diverse shapes frequently; others (Urdu, Bulgarian, Estonian) less so. Consistency suggests universally favored shapes (ease/efficiency), variation points to

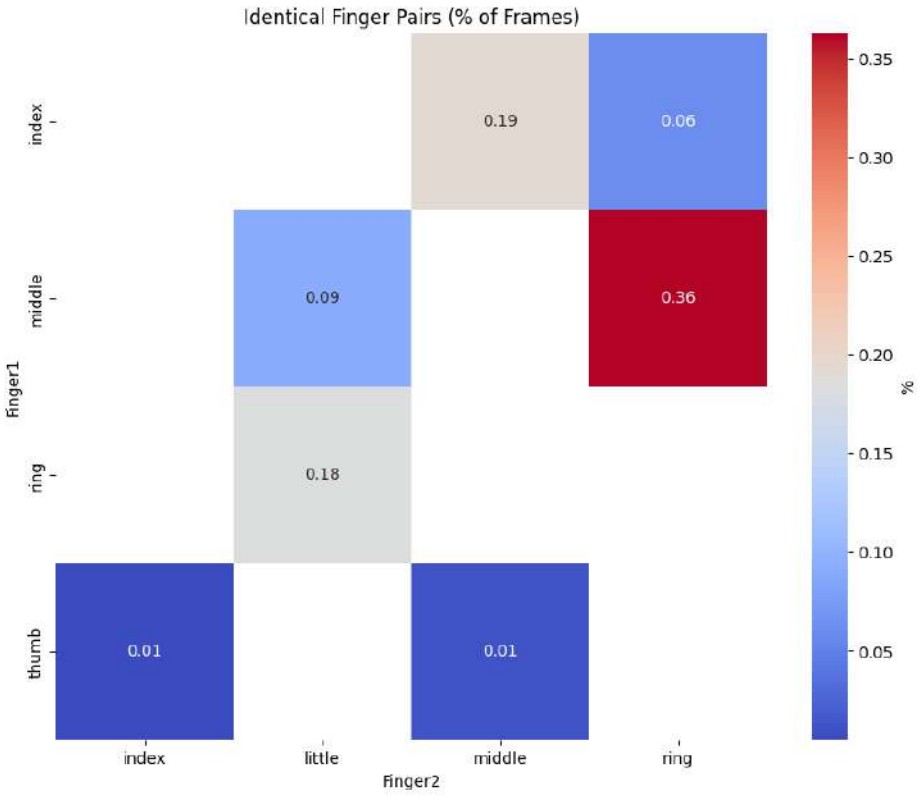


Figure 5.8: Pairs of Identically Shaped Fingers.

regional/cultural factors. Findings support similar frequency distributions for core shared handshapes, likely due to motor/communicative constraints, with differences needing further study.

35 Most Frequent Handshapes Across Languages: Comparing frequency distributions for the top 35 classes (ej, ij, fk, mv, etc.) across 33 languages (Figure 5.10) confirmed consistent use of some shapes (e.g., ej, ij, mv) suggesting universal tendencies, alongside language-specific variations. This points to a core shared inventory reflecting motor/communicative needs, augmented by cultural/linguistic factors.

Correlation of Handshape Frequencies: The language similarity matrix (based on correlations, Figure 5.11) measures usage similarity (0 to 1). It helps identify shared characteristics and divergences. Results showed high similarity within some regional clusters (European/Asian), cross-regional similarities (universal preferences), and divergent languages (local cultural influences). Understanding these dynamics informs SL recognition and standardization efforts.

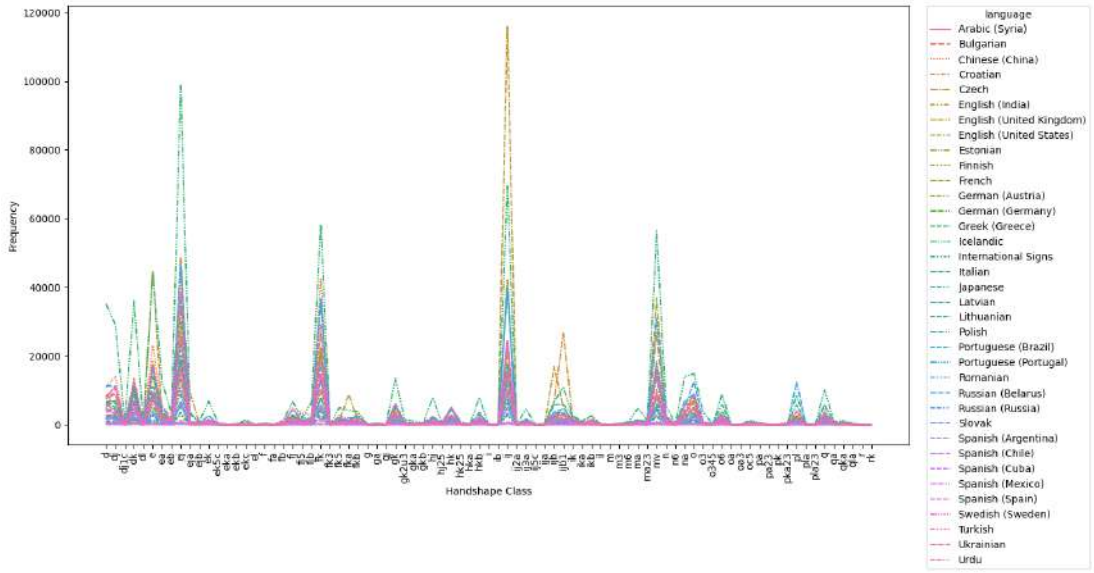


Figure 5.9: Distribution of Handshape Frequencies Across Languages

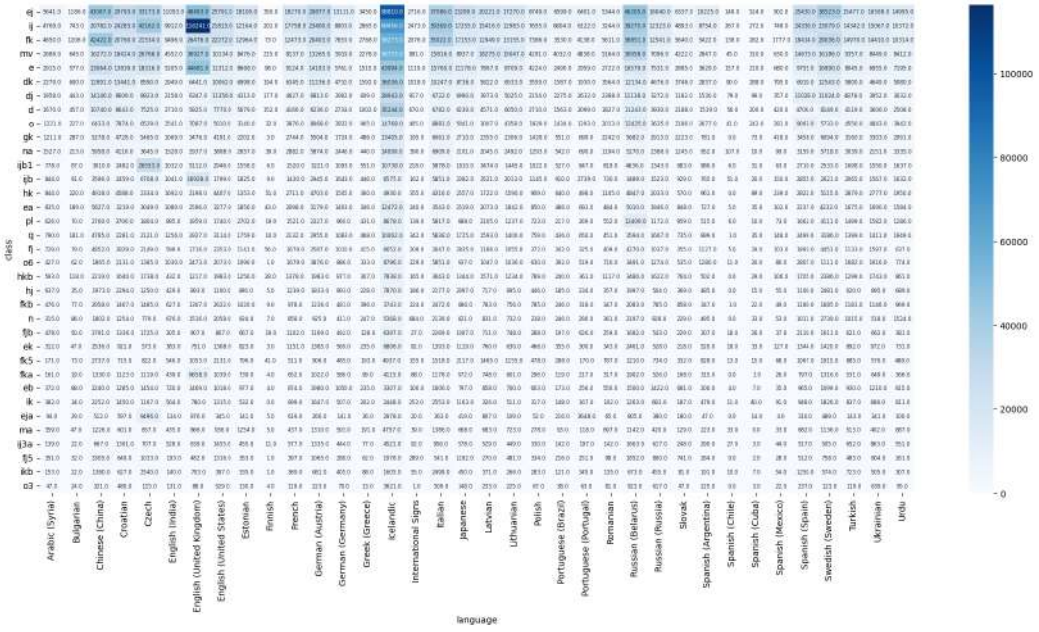


Figure 5.10: 35 Most Frequent Handshapes Across Languages.

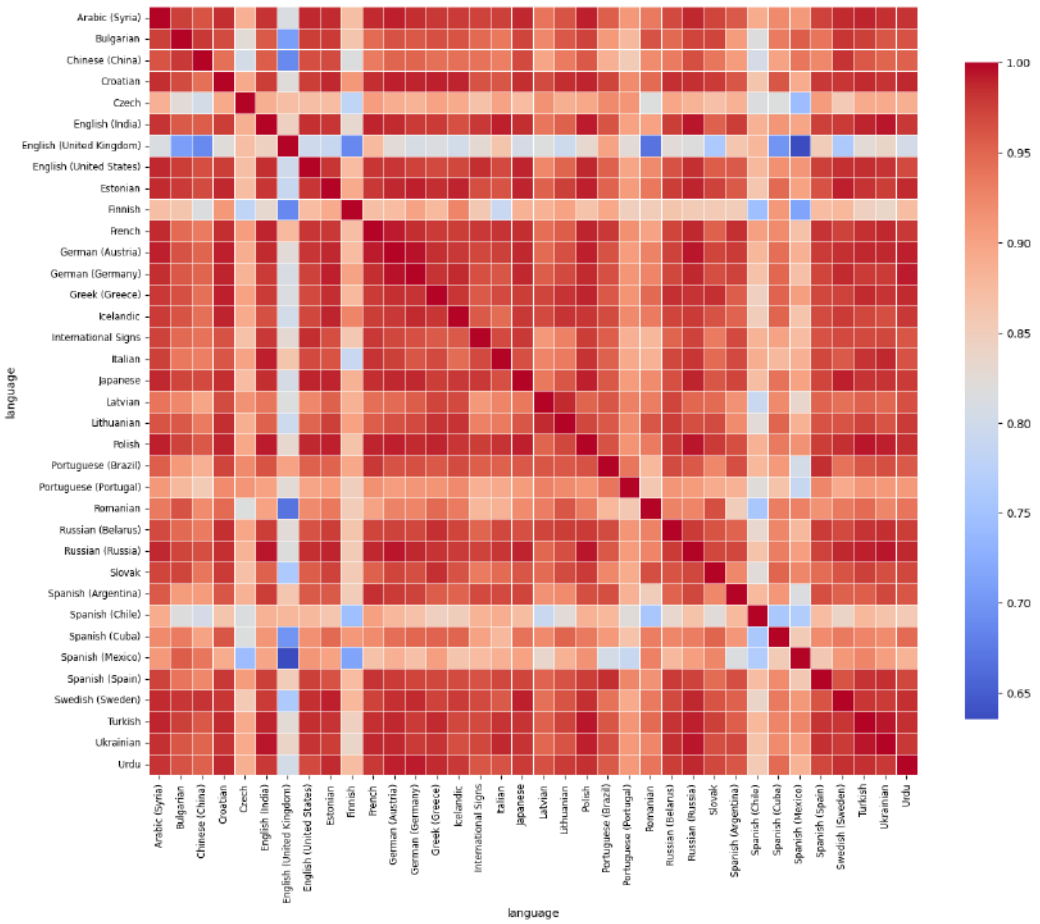


Figure 5.11: Correlation of Handshape Frequencies.

5.2.3 Language Families

Hierarchical clustering applied to handshape frequency data identified patterns and relationships. A dendrogram (Figure 5.12) visualized how languages group based on shared usage, revealing clusters corresponding to potential linguistic families and regional influences, highlighting similarities and variations. The resulting cluster map showed some regional groupings based on handshape frequencies.

5.3 Non-manual components representation

In addition to manual activity, sign language videos contain non-manual activity, including head movements, body movements, facial expressions, and mouth movements. Non-manual activity can express grammatical information (e.g., about subject/object/verb) and convey information not present in manual activity. Non-manual activity can be represented by non-manual glosses (written representations). Annotation involves adding these glosses to non-manual segments, whose boundaries are typically identified using rules and human judgment.

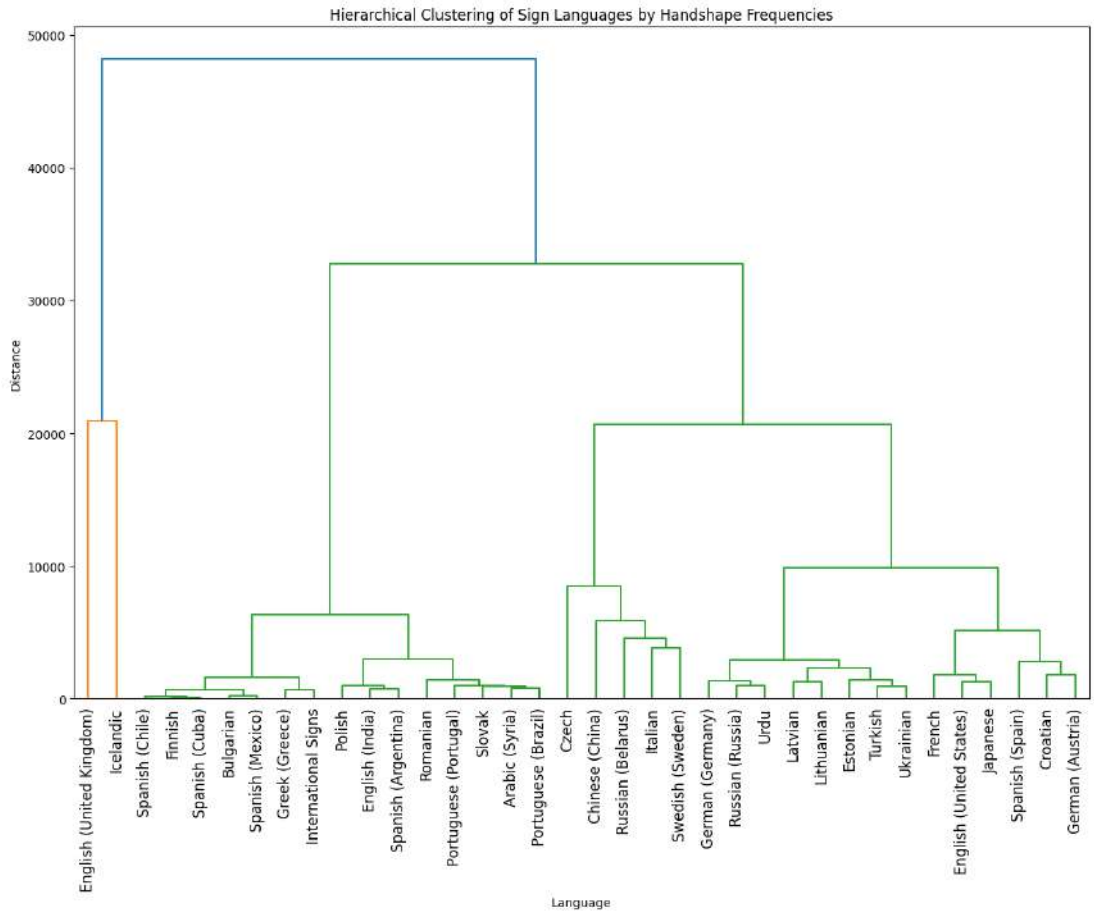


Figure 5.12: Hierarchical clustering of Sign Languages by Handshape Frequencies.

5.3.1 Baseline methods

Signing recognition can be viewed as a variation of action recognition or human pose estimation tasks. Keypoint detection libraries like OpenPose [Cao et al., 2017, Wei et al., 2016] enable evaluation of both manual (hand) and non-manual (face, pose) features. Figure 5.13 demonstrates keypoints extracted by OpenPose. Recent action recognition work [Tran et al., 2018b] introduced the effective R(2+1)D spatiotemporal convolutional block. To analyze our collected dataset, we employed both keypoint-based and action-recognition approaches as baselines for isolated sign recognition. We extracted isolated clips from the statement-question subset for specific signs (‘what’, ‘who’, etc.), distinguishing between statement and question forms (20 classes total).

5.3.1.1 Pose estimation baseline

We extracted keypoints frame-by-frame using the OpenPose [Cao et al., 2017, Wei et al., 2016] library and feed them to a classification algorithm. Therefore, we utilized classical machine learning, namely Logistic Regression, by concatenating keypoint sequences into one sample per video. Since we aimed to compare non-manual

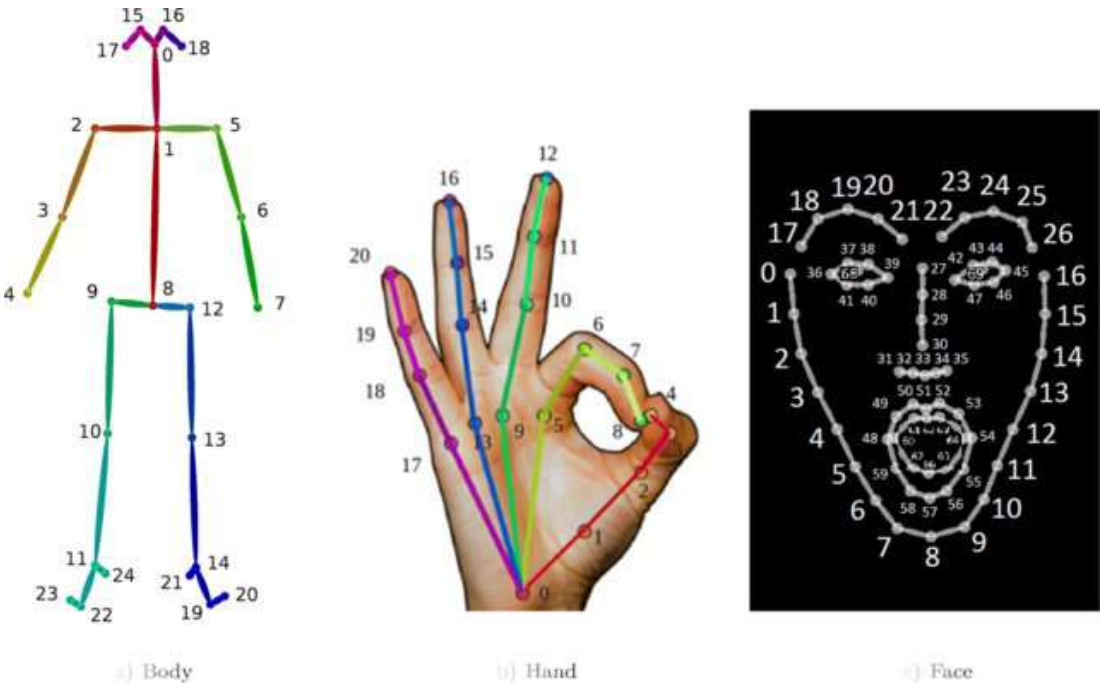


Figure 5.13: Openpose detected body, hand and face keypoints [Cao et al., 2017, Wei et al., 2016]

feature performance, we prepared two conditions: manual only feature vectors (max 30 frames * 84 keypoints = 2520 features) and manual and non-manual features combined (max 30 frames * 274 keypoints = 8220 features). We used the scikit-learn library [Pedregosa et al., 2011] for Python for keypoint classification in these experiments.

5.3.1.2 Action recognition baseline

Recent action recognition work often employs Two-Stream Inflated 3D ConvNet (I3D) [Carreira & Zisserman, 2017b] or the R(2+1)D spatiotemporal block [Tran et al., 2018b]. Both architectures are typically trained on ImageNet [Russakovsky et al., 2015a] and fine-tuned on Kinetics [Kay et al., 2017]. In this paper, we employed the R(2+1)D model [Ghadiyaram et al., 2019], known for high accuracy and relative speed (using only video frames as input, unlike approaches needing optical flow). It was additionally pre-trained on over 65 million videos. To recognize signs from our dataset, we fine-tuned R(2+1)D on the statement-questions subset. Since our subset had a different number of classes, only the final fully connected layer was retrained.

5.3.1.3 Data augmentation

The main problem developing sign language recognition algorithms is often insufficient data size or diversity for generalization. Thus, we suggest a simple method to augment

fixed-length image sequences from variable-length videos (constraint: video length $m \geq$ fixed length n). We pick n equally distanced frames with step $s = \lfloor m/n \rfloor$, starting from a random initial frame i (where $1 \leq i \leq s + k$, $k = m \pmod n$). The augmented sequence is $S = (f_i, f_{i+s}, ..., f_{i+(n-1)s})$.

5.3.1.4 Implementation details

The action recognition baseline (R(2+1)D) was implemented in PyTorch [Paszke et al., 2019a] using the [Ghadiyaram et al., 2019] pre-trained model. Input size (consecutive frames) was 8, batch size 16. Trained 20 epochs, starting LR 0.0001. Frames scaled to 112x112 (kept ratio), random cropping (scale 0.6-1.0) applied during training. The pose estimation baseline used scikit-learn [Pedregosa et al., 2011] Logistic Regression ('lbfgs' solver, L2 penalty) on keypoint sequences from OpenPose [Cao et al., 2017, Wei et al., 2016].

Table 5.3: Mean scores of accuracy for the question-statement subset after 10 iterations with random train/test splits.

	R(2+1)D	Logistic regression	
Features	Full frame	Manual only	Manual & Non-manual
Mean	85.95%	73.38%	77.04%
Std Dev	0.9	0.41	0.5

5.3.2 Experimental Results

A series of experiments investigated whether non-manual features improve recognition accuracy. All used isolated signs from the Question-Statement subset (20 classes: 10 signs as statement/question). The first experiment classified these 20 classes using two baselines: Logistic Regression (manual only vs. manual+non-manual keypoints) and R(2+1)D (full frames). Evaluation used 10 repetitions with random train/test splits. Table 5.3 presents mean scores and standard deviations. The second experiment compared accuracy improvements from different combinations of non-manual keypoint components using Logistic Regression. Table 5.4 presents accuracy scores per combination.

5.3.2.1 Question vs. Statement

Our first experiment used the Question-Statement subset (20 classes). Manual and non-manual features were extracted for the isolated signs. R(2+1)D achieved the highest accuracy (86%), 9% higher than Logistic Regression. For Logistic Regression trained

Table 5.4: Comparison of results of features combinations.

Features combination	Accuracy
Manual only	73.4%
Manual & Only eyebrows	73.25%
Manual & Eyebrows, mouth	77.2%
Manual & Only mouth	77.5%
Manual & Non-manual all	77.04%
Manual & Face, eyebrows, mouth	**78.2%**

on keypoint sequences, mean test accuracy was 73.4% (manual-only) and 77% (manual + non-manual features). As expected, non-manual features improved results by 3.6% on average. At the same time, the improvement was not very high, possibly because the number of non-manual features is much larger than manual features, and simple concatenation may not be optimal. This suggests that while non-manual cues captured by keypoints are indeed informative, advanced deep learning models operating on raw pixels (like R(2+1)D) might be better able to extract and utilize this information, or alternatively, more sophisticated methods for fusing manual and non-manual keypoint features could be beneficial.

5.3.2.2 A case of combining different modalities

This experiment compared different combinations of non-manual markers (eyebrow/-head position vs. mouthing) using Logistic Regression and analyzed their role (Table 5.4). Lowest test accuracy (73.25%) occurred with manual features + eyebrow keypoints only; eyebrows alone provided little valuable information here. Accuracy improved only when combined with other features. Highest accuracy (78.2%) resulted from combining manual features with face outline, eyebrows, and mouth keypoints. Using only mouth keypoints with manual features also increased accuracy (77.5%) compared to the baseline using all non-manuals (77%). Thus, mouthing provides extra useful information, as signers often articulate words while signing. Eyebrows and head position provide grammatical markers differentiating statements/questions. This reinforces the idea that non-manual signals often work in concert, and effective recognition may require models capable of integrating information from multiple non-manual articulators simultaneously.

5.3.3 Statistical significance testing

To quantify whether the observed differences in accuracy across feature sets are statistically reliable (cf. Slide 45), we ran 10 random train/test splits for each baseline and then compared models pairwise, per split: (i) Manual only (keypoints), (ii) Manual+Non-manual (keypoints), and (iii) Full-frame (R(2+1)D). Because $n = 10$

is small and the normality of split-wise differences cannot be guaranteed, we report both a paired t-test (with t-based 95% CIs for the mean gain, df$= 9$) and the Wilcoxon signed-rank test for robustness.

Table 5.5: Raw accuracy scores (%) for 10 random splits. Means and SDs are computed across splits.

Split	Manual only	Manual+Non-manual	Full-frame (R(2+1)D)
1	72.8	76.0	84.2
2	73.0	76.9	86.4
3	73.3	76.9	86.4
4	73.9	77.3	85.4
5	73.1	77.3	86.3
6	73.6	76.9	85.8
7	73.9	77.9	86.7
8	72.9	76.7	84.9
9	73.7	77.2	87.2
10	73.6	77.3	86.2
Mean $\pm$ SD	**73.38 $\pm$ 0.41**	**77.04 $\pm$ 0.50**	**85.95 $\pm$ 0.90**

Table 5.5 confirms the headline pattern across the same 10 random splits: the Manual+Non-manual keypoint model consistently outperforms Manual only by about 3–4 percentage points (pp), and the Full-frame (R(2+1)D) model substantially outperforms both keypoint-based variants. The paired analyses in Table 5.6 show that these differences are not only practically meaningful but also statistically decisive.

Table 5.6: Paired significance tests across the same 10 splits. Mean gain reported in percentage points (pp). 95% CIs are t-based (df$= 9$).

Comparison	Mean gain (pp)	95% CI (pp)	t-test p	Wilcoxon p
Manual+Non-M $-$ Manual	3.66	[3.43, 3.89]	< 0.001	0.002
Full-frame $-$ Manual+Non-M	8.91	[8.46, 9.36]	< 0.001	0.002
Full-frame $-$ Manual	12.57	[12.02, 13.12]	< 0.001	0.002

First, adding non-manual articulators produces a mean gain of 3.66 pp over Manual only, with a tight t-based 95% confidence interval [3.43, 3.89] (df$= 9$), and highly significant results under both the paired t-test ($p < 0.001$) and the Wilcoxon signed-rank test ($p = 0.002$). The fact that gains are positive on every split indicates the improvement is consistent and not driven by outliers.

Second, moving from keypoints to full-frame video yields a much larger performance jump. Relative to Manual only, Full-frame improves by 12.57 pp with 95% CI [12.02,

13.12] (paired t-test $p < 0.001$; Wilcoxon $p = 0.002$), and it still outperforms Manual+Non-manual by 8.91 pp (95% CI [8.46, 9.36]; paired t-test $p < 0.001$; Wilcoxon $p = 0.002$). The narrow confidence intervals across all contrasts reflect low between-split variability and reinforce the robustness of these effects at $n = 10$.

Taken together, the results indicate that (i) explicitly modeling non-manual articulators is beneficial even within a simple keypoint classifier, and (ii) end-to-end spatiotemporal modeling on raw pixels captures discriminative cues (including subtle manual and non-manual dynamics) more effectively than keypoints alone.

5.4 Chapter Conclusion

This chapter provided a detailed exploration of computational methods for representing manual and non-manual components essential to sign language understanding, focusing on RSL and KRSL data. The investigation into manual components highlighted the challenges of unsupervised handshape classification due to confounding visual factors, while demonstrating the effectiveness of supervised methods based on manually curated data. Regarding non-manual components, experiments confirmed their measurable contribution to sign recognition accuracy, particularly features related to the mouth and combined facial cues. These findings collectively underscore the importance of comprehensive feature representation for robust SLP and provide methodological context for the subsequent chapters on annotation tools (Chapter 6) and recognition/translation systems (Chapter 7) within this thesis.

Chapter 6

SLAN-tool: Sign language annotation tool

Sign language annotation involves adding descriptive metadata to sign language videos to enable further processing and detailed analysis. The specific type of metadata added and the annotation methodology employed typically depend on the subsequent processing or research goals. Fundamentally, annotation transforms raw video data into a structured representation of the linguistic information contained within. While analogous in purpose to annotating text documents or spoken language recordings, sign language annotation presents unique challenges stemming from the visual and multi-channel nature of the modality.

The primary objective of sign language annotation is to facilitate linguistic research by making the information embedded in sign language videos accessible and analyzable for researchers. How this is achieved depends directly on the specific types of linguistic information being annotated (e.g., glosses, translations, phonetic details, non-manual markers).

Historically, sign language annotation has predominantly been a manual undertaking. This reliance on manual methods stems from the complexity of the information being annotated, which has posed significant challenges for the development of fully automatic annotation techniques. However, recent years have seen progress in developing computational methods capable of automating certain aspects of the annotation process.

Several current research projects are addressing the automatic processing of sign language (SL) and its annotation. For instance, Chaaban et al. [Chaaban et al., 2021] proposed a system for automatically annotating non-manual features like mouthing and head direction, utilizing coordinates extracted using the OpenPose library [Cao et al., 2019]. These low-level features were then used to help define temporal boundaries of signs. Their results indicated capability in detecting lexical signs (F1 score = 0.68), with performance improving when handshape information was also used for segmentation. Belissen et al. [Belissen et al., 2020] presented a method for compact signer modeling incorporating 3D face landmarks, body pose, and handshape probabilities, feeding these features into a CRNN for learning linguistic features. Skobov and Lepage [Skobov & Lepage, 2020] introduced a novel tree representation based on HamNoSys grammar to aid in generating training data and classifying complex

movements via machine learning. Many such works leverage keypoint extraction tools like OpenPose, but a drawback is the dependence on this intermediate keypoint extraction step before annotation can occur. An alternative direction explores using spoken language subtitles, as demonstrated by Albanie et al. [Albanie et al., 2020b], who introduced a scalable approach using weakly-aligned subtitles from broadcast footage. These efforts highlight a growing interest in leveraging computer vision and NLP techniques to reduce the manual burden of sign language annotation.

Despite these advances, a need remains for accessible, flexible tools that can assist researchers with accurate and customizable annotation, preferably via a web-based interface independent of specific operating systems or complex software installations. To address this need, as part of this thesis work, we developed SLAN-tool, a semi-automatic tool specifically tailored for annotating sign language videos. The goal was to create a practical tool combining manual annotation flexibility with ML-powered assistance to improve efficiency.

This chapter first describes the process used to gather user requirements for the SLAN-tool system. Subsequently, we discuss the system design and user interface developed based on these requirements. Finally, we detail the neural network models integrated into the tool for performing automatic annotation of signing segments and handshape classification.

6.1 User requirements

System requirements describe the services a system should provide and its operational constraints, reflecting user needs. Establishing clear project goals is the first step in successful requirements gathering. Our process began by analyzing existing sign language annotation tools, with ELAN being a prominent example known for its rich functionality but also its potentially steep learning curve for new users.

Following this initial analysis, the project objective became clear: provide researchers with a specialized tool for semi-automatic sign language video annotation, potentially enabling easier combination of annotated datasets across different sign languages. Key goals included developing a web-based interface and integrating modules for semi-automatic annotation generation. To gather high-level requirements, interviews were conducted with potential users, including sign language researchers and data annotators. The core user requirements identified were the ability to:

- Upload and play selected videos within the main interface;
- Send uploaded videos to backend modules for automatic annotation processing;
- View automatically generated annotations on relevant tiers (layers) within the interface;

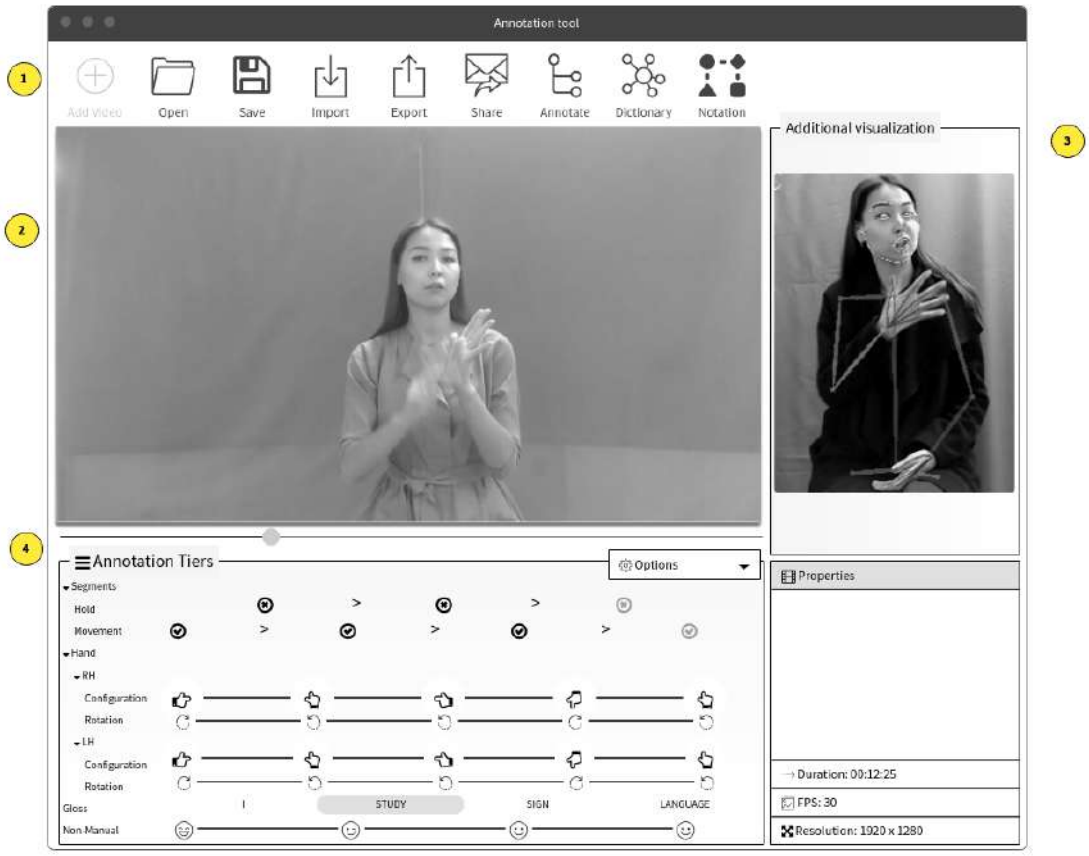


Figure 6.1: Proposed User interface for annotation tool.

- Manually adjust and update generated annotations (e.g., correct predicted classes, refine segment boundaries);
- Add custom annotation tiers as needed for specific research purposes;
- Export and import annotations in various formats (e.g., JSON, CSV, ELAN-compatible format);
- Share annotation projects or results with collaborators.

These requirements emphasize a need for both automation support and manual flexibility, along with interoperability with existing tools like ELAN.

6.2 User interface and functionality

The User Interface (UI) for the annotation tool primarily consists of a main page supplemented by pop-up windows for menu options. Figure 6.1 illustrates the proposed UI layout. The main page integrates four key areas: (1) Control functions, (2) Video player, (3) Annotation tiers, and (4) Additional visualization area.

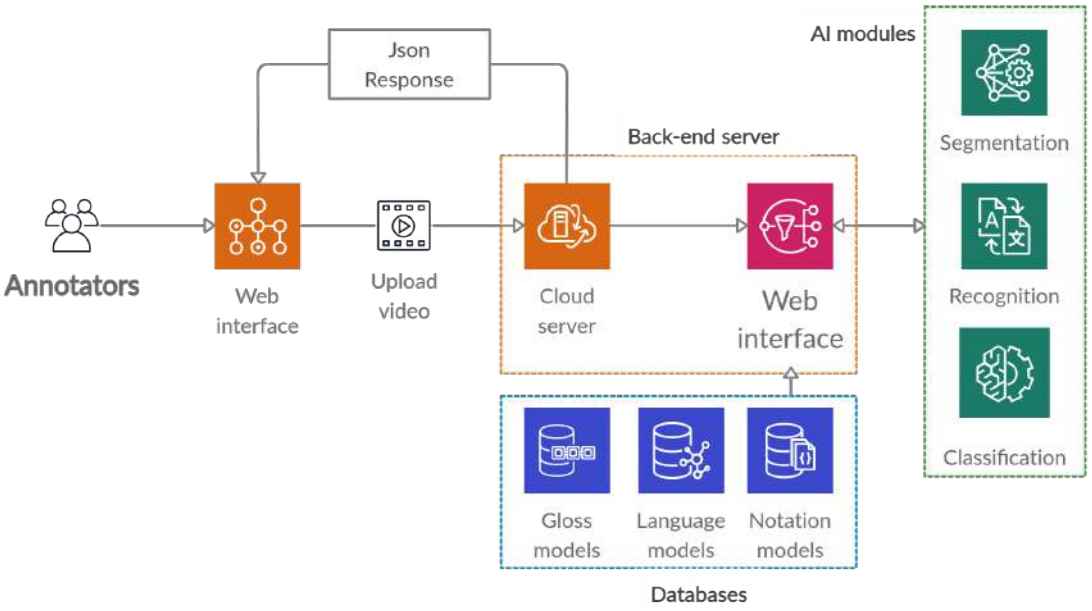


Figure 6.2: Overview of the Sign Language Annotation tool's Web service.

1. The control functions area provides buttons for core actions: Upload video, Process video (initiate automatic annotation), Export/import annotation file, Save project, Share project, and Annotate (enter manual annotation mode).

2. The video player area displays the currently loaded video with a timeline positioned below it for navigation.

3. The additional visualization area is designed to display supplementary information that may not fit well within the tiered annotation structure (specific content not detailed here).

4. The annotation tiers area displays annotations over time. It includes a predefined list of common tiers (e.g., translation, gloss, right handshape, left handshape), with the capability for users to add custom tiers as required.

6.3 System design

System design involves detailing the software system's functions, services, and operational constraints, defining precisely what needs to be implemented. We opted for a web-based tool architecture to enhance user convenience, eliminating challenges associated with installing specific software libraries or requiring powerful local computational resources. The SLAN-tool is designed to be accessible via standard web browsers and features a user-friendly interface. The computationally intensive automatic annotation processing (running the neural network models) is performed on

cloud-based servers, with results returned to the user's browser interface. Figure 6.2 presents a high-level overview of the Sign Language Annotation tool's web service architecture. This client-server architecture makes powerful ML models accessible without requiring end-users to have specialized hardware.

6.4 Neural network models

The SLAN-tool incorporates neural network models to provide semi-automatic annotation capabilities. We first describe the model used for Sign Language Segmentation, which identifies active signing periods within a video. Next, we discuss the models developed for Handshape Classification, which recognize handshape categories within the detected segments.

6.4.1 Signing segmentation model

The task of signing segmentation, in this context, involves detecting the start and end frame boundaries for segments within a video where active signing occurs. This differs from segmentation aimed at identifying individual signs or glosses, which typically requires more detailed annotated data. Our approach treats signing segmentation as an action recognition problem: classifying short video snippets as belonging to 'signing-start', 'signing-end', or 'no-signing' categories. The primary goal is to assist annotators by automatically highlighting potentially relevant parts of a video, allowing them to focus their manual annotation efforts more efficiently, particularly for long recordings. This can potentially increase the speed and efficiency of the overall annotation process. Automating the detection of non-signing segments can significantly reduce the amount of video an annotator needs to manually review.

6.4.2 Handshapes keypoints classification model

To offer an alternative and potentially faster method for handshape analysis within the tool, potentially reducing processing time compared to image-based classification, we trained an additional model based on hand keypoints. This model utilizes keypoints extracted using the MediaPipe framework, which provides 21 keypoint coordinates (x, y, z) per hand from video sequences. For this model, we used only the x and y coordinates, discarding the z (depth) value. The model was trained on the same handshape dataset described in Chapter 5 (covering 84 classes with 101,098 total samples after augmentation). Keypoints were extracted first, and then a classification model was trained using Google's AutoML Tabular library.

6.5 Implementation

The SLAN-tool was implemented utilizing various open-source libraries and software tools. All source code for the SLAN-tool is intended to be open-sourced and made available via the repository at https://github.com/krslproject/slan-tool.

6.5.1 Annotation tool

The user interface was implemented using standard web technologies: HTML5, CSS3, JavaScript (JS), JQuery, and the Bootstrap library for styling. The back-end processing logic was implemented using the Python programming language with the Django web framework. A Flask framework was likely used for serving the machine learning model predictions. PostgreSQL was used as the database management system. Cloud storage for video data was handled using AWS S3. The web service was deployed using Gunicorn as the WSGI server and Nginx as the reverse proxy/web server.

The computational server dedicated to processing the automatic annotation requests possesses the following specifications:

- Operating system: Ubuntu 18.04.2 LTS.
- CPU: Intel Core-i9.
- RAM size: 32 gigabytes (GB).
- GPU: 2 x NVIDIA 2080 Ti GPUs, each with 11 GB VRAM (Original text mentioned 24GB - clarification needed).
- Hard disk drive size: 2 terabytes.

These specifications indicate a reasonably powerful server capable of handling deep learning model inference for annotation tasks.

6.5.2 Classification Models

The sign segmentation and handshape classification models integrated into SLAN-tool were pre-trained using the TensorFlow [Abadi et al., 2016] and PyTorch [Paszke et al., 2019b] machine learning libraries. These models automatically detect signing segments and classify handshapes within uploaded videos.

6.5.2.1 Sign Language Segmentation Model Details

To train the signing segmentation model, sign language videos were divided into three categories: 'signing-start', 'signing-end', and 'no-signing' segments.

Figure 6.3: Datasets used for sign language segmentation model. 1) KRSL [Imashev et al., 2020], 2) WLASL [Li et al., 2020a], 3) Dicta-Sign–LSF–v2 [Belissen et al., 2020].

Datasets Used: Training data was extracted from three different datasets: the K-RSL corpus developed in this thesis [Imashev et al., 2020], the WLASL dataset [Li et al., 2020a], and the Dicta-Sign–LSF–v2 dataset [Belissen et al., 2020]. Figure 6.3 shows sample frames from each. For the K-RSL data, segments were manually annotated using the VIA tool. For WLASL and Dicta-Sign, clips corresponding to the start and end points of annotated signs were automatically extracted (adding 16 frames around the annotated points), followed by manual selection of valid clips. Table 6.1 shows the number of video clips collected for each category from each dataset. Combining data from multiple diverse datasets helps improve the robustness and generalization ability of the segmentation model.

Table 6.1: Segmentation training videos of each category.

	KRSL	WLASL	Dicta-Sign–LSF–v2	Total
Signing-Start	300	1000	500	1800
Signing-End	300	1000	500	1800
No-Signing	300	1000	500	1800

Model Training: Action recognition is an active research field. We chose the R(2+1)D model architecture for segmentation, as it achieves high accuracy while being computationally faster than some alternatives. Its effectiveness partly stems from pre-training on a massive dataset (65 million Instagram videos). Its speed advantage comes from using only RGB frames, unlike methods requiring pre-computed optical flow. Our implementation utilized pre-trained weights available through PyTorch Hub for models trained on these large datasets. Training was performed on an NVIDIA 2080 RTX GPU.

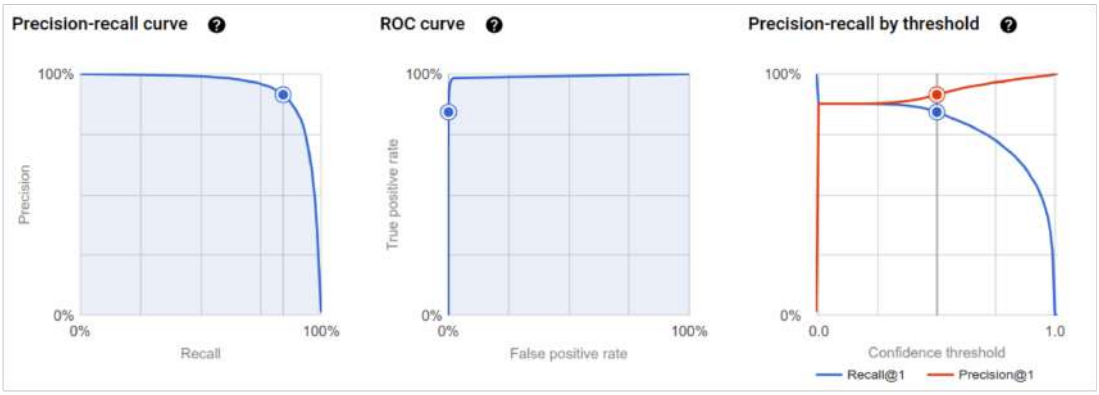


Figure 6.4: The model's results trained on the collected dataset with AutoML.

6.5.2.2 Handshape Keypoints Classification Model Details

Dataset Used: The keypoint-based handshape classification model was trained using the same dataset employed for the image-based supervised handshape classification described in Chapter 5. Keypoints (21 per hand, x, y, z coordinates) were extracted using the MediaPipe Hands library. Only the x and y coordinates were used as input features for the classification model (the z coordinate was eliminated). Using keypoints provides a lower-dimensional representation compared to raw images, potentially allowing for faster training and inference, but may lose some shape details.

Model Training: We trained this model using Google's AutoML Tabular library, which automates the process of building and deploying machine learning models on structured data like keypoint coordinates. The final trained model achieved a Precision of 91.4% and a Recall of 84.2% at a confidence threshold of 0.5. Figure 6.4 shows performance results from the AutoML training process.

6.6 System evaluation and usability testing

6.6.1 Experts review

An expert review is an evaluation method used to identify potential usability issues in a product or service. It involves usability experts examining the system against established principles or heuristics. This method is valuable within a user-centered design process as it does not necessarily require direct interaction with end-users during the evaluation itself.

To test the functionality and usability of the SLAN-tool, interviews were conducted with two sign language researchers via Zoom meetings. Feedback was collected after each interview and used to iteratively improve the tool's interface and functionality. The first interview occurred early in development, while the second took place mid-project after addressing the first expert's feedback. During the interviews, experts were asked to

explore the system, comment on difficulties encountered during initial use, and compare the tool to annotation software they typically use. Table 6.2 lists the questions posed during these expert interviews.

6.6.1.1 Expert 1 Feedback Summary

The first expert noted some initial usability challenges and suggested clearer instructions were needed. Specific points included difficulty understanding how to adjust annotation segment boundaries and how to replay selected segments. They recommended more active visual feedback for user actions, such as changing tier colors upon selection or showing confirmation messages after edits. While finding the annotation editing process itself intuitive, they requested options for customizing the additional visualization area and displaying text annotations directly on the tiers. Video control suggestions included adding loop/single-play options and ensuring the video player automatically seeks to the corresponding segment when an annotation is selected. Further recommendations involved providing template tiers (e.g., for gloss, right/left hand), clearer project initiation guidelines, and controls for managing tiers (add, delete, reorder). The expert also noted that sign language linguists familiar with ELAN might expect similar functionalities and suggested adding a feature to extract annotated video segments. In response to concerns about first-time use, quick tips and keyboard shortcuts were added to the tool.

6.6.1.2 Expert 2 Feedback Summary

The second expert interview occurred after implementing changes based on the first review. This expert suggested adopting ELAN's method of typing annotation text directly into the selected segment on the timeline. They also recommended enabling video scrolling by dragging directly on the timeline, rather than just clicking. Regarding handshape annotation, suggestions included reorganizing categories, adding category names to the tier, and allowing annotation changes directly from the additional visualization area. Saving data upon pressing the Enter key was proposed for convenience. Crucially, this expert emphasized the need for integration with ELAN, specifically requesting import/export functionality using ELAN's format (.eaf). Other suggestions included adding a frame-by-frame timeline navigation option and considering options for annotating non-manual tiers.

6.6.2 Usability testing

Usability testing involves observing representative users performing typical tasks with a system to identify usability problems. A facilitator guides the participant through tasks, observes their behavior, and gathers feedback.

Table 6.2: List of questions asked during the expert interviews.

1	Please, try to start annotating the video.
2	Please, try to annotate in the handshapes layer.
3	Please, try to change the handshape.
4	Please, try to automatically annotate signing.
5	Please, press Handshape button.
6	Do you need additional visualization?
7	What changes would you make to additional visualization?
8	Do you need Hold-Movement annotation?
9	Do you need OpenPose visualization?
10	In general, what changes would you make?
11	What functionality is missing/needed from ELAN?

For SLAN-tool, usability testing sessions were conducted with 3 sign language data annotators experienced with SurdoBot, a simpler web-based tool previously used for glossing the KRSL dataset. Individual 1-hour sessions were held via Zoom, during which participants annotated short sign language video clips using SLAN-tool. Two main scenarios were tested: comparing SLAN-tool to their prior experience with SurdoBot for gloss annotation, and comparing manual versus automatic annotation of signing segments within SLAN-tool.

6.6.2.1 Test scenarios

The testing format involved giving users specific tasks followed by open-ended questions about their experience. Initial questions aimed to create an open atmosphere (e.g., "What do you see on the screen?", "What do you think this is for?"). Two main test scenarios were then conducted, with guiding questions as listed in Table 6.3.

6.6.2.2 Post-test questions

Following the task scenarios, follow-up questions were asked for further clarification and overall feedback:

- Overall, what was your experience using the website?
- If you could change one thing about the website, what would it be and why?
- What one aspect of the website did you find most useful or exciting, and why?

6.6.2.3 Key findings

Participants generally encountered some initial difficulty with adding tiers and gloss annotations, primarily due to SLAN-tool having more features than the simpler SurdoBot tool they were used to. Adding a "Help" button providing quick tips improved

Table 6.3: List of questions asked during the usability testing.

Scenario:	**SurdoBot vs SLAN-tool**
Description:	Annotate a short sign video with gloss notation.
Questions:	How would you play the video?
	How would you change the speed of the video?
	How would you add a new tier?
	How would you add a new gloss?
	How would you change a gloss?
	How would you change gloss boundaries?
Scenario:	**SLAN-tool vs SLAN-tool automatic annotation**
Description:	Annotate a short sign video with signing segments manually and then do it with automatic annotation.
Questions:	How would you add a new signing segment?
	How would you delete a segment?
	How would you change signing boundaries?
	Please try to annotate signing boundaries with automatic annotation.

the experience, allowing users to find instructions when needed. A common suggestion was to allow direct text input next to the selected annotation segment, which was implemented and found to improve annotation speed and convenience. Regarding the automatic annotation features (signing segmentation), all participants agreed this functionality made the annotation process significantly easier and faster, as they only needed to review and adjust the automatically generated segments rather than creating them entirely from scratch. This user feedback validates the core concept of semi-automatic annotation for improving efficiency.

6.7 Chapter Conclusion

The primary concept behind SLAN-tool is to offer a convenient annotation solution that avoids the need for users to install specialized software locally. Users can simply access the tool via a web browser and upload their videos. The service is intended to be freely available and performs computationally intensive tasks on a dedicated server (currently hosted on AWS), thus not requiring significant computational resources on the user's machine. Furthermore, SLAN-tool is designed for interoperability with established tools like ELAN, offering import and export functionality using ELAN's .eaf format. This allows researchers, for instance, to use SLAN-tool for rapid semi-automatic annotation (e.g., segmentation and handshape classification) and then export the results to ELAN for further detailed linguistic analysis or integration with other annotations. Providing compatibility with existing community standards like ELAN is crucial for tool adoption.

Potential use cases for SLAN-tool include:

- **Automatic annotation assistance:** Automatically segmenting videos into signing/non-signing parts and identifying potential handshape configurations within active segments. These automatic annotations can then be manually reviewed, corrected, and exported.
- **Streamlined gloss notation:** Adding custom tiers for glossing becomes more efficient when combined with the automatic segmentation model, as annotators can focus primarily on the active signing segments identified by the tool.

A current primary limitation relates to the computational resources available for the backend processing. The service runs on a self-hosted server with limited GPU capacity (2 GPUs), which imposes practical limitations on the duration or number of videos that can be processed simultaneously or quickly. Future plans involve migrating the backend to a more scalable cloud-based infrastructure, potentially allowing users to automatically annotate longer videos more efficiently. Scalability is a key challenge for providing centralized, computationally intensive services like automatic annotation.

Chapter 7

Sign language recognition and translation

The typical pipeline for vision-based sign language recognition involves three main stages: data acquisition, feature extraction, and recognition/translation. During data acquisition, sign language utterances are captured, usually via video camera. The feature extraction stage then processes these video data to derive informative representations. Finally, the recognition or translation stage utilizes these features to recognize the signs performed or translate them into another language. The following sections briefly introduce these concepts before detailing the specific methods and experiments conducted in this thesis using the K-RSL datasets.

7.1 Data acquisition and processing

The initial step is data acquisition. While some research utilizes sensor gloves to capture signing data, this work, like much current research, focuses on vision-based recognition using standard camera input. The overall acquisition process involves two main phases: data collection (recording the signer) and data pre-processing. The pre-processing steps applied to the captured video might include tasks such as background removal or subtraction, hand segmentation, face detection, or the extraction of skeletal keypoints, depending on the subsequent feature extraction approach. The quality and nature of the data acquired in this stage fundamentally impact the potential performance of the entire system, motivating the extensive dataset creation efforts described in Chapters 3 and 4.

7.2 Feature extraction

Following acquisition and pre-processing, the next crucial step is feature extraction. This stage aims to derive salient information (features) from the video data that effectively represents the sign language performance. These extracted features serve as input for subsequent classification or translation models. Features are generally categorized into manual features (relating to hand shape, movement, position, orientation) and non-manual features (relating to facial expressions, mouth movements, head/body posture, etc.). As discussed in Chapter 5, both manual and non-manual features convey

important linguistic information in sign languages. The specific features extracted depend on the chosen modeling approach.

7.3 Classification and translation

The final stage involves using the extracted features to classify the sequence of signs (Sign Language Recognition - SLR) or translate the signed utterance into a target spoken language (Sign Language Translation - SLT). The following subsections describe the baseline models employed in this work for evaluating SLR and SLT performance on the newly created K-RSL datasets.

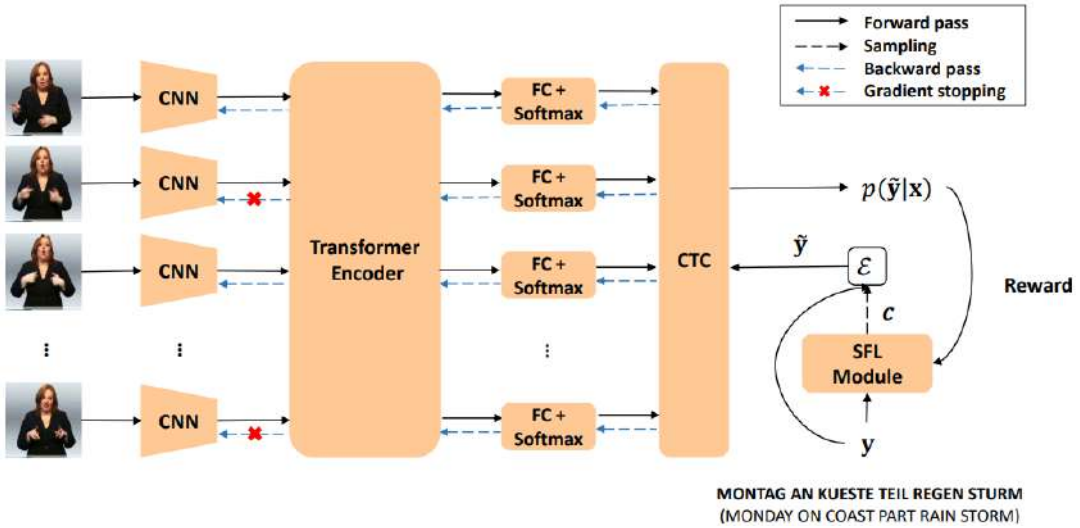
7.3.1 Baseline methods

To establish benchmark performance on the FluentSigners-50 dataset introduced in Chapter 3, we selected state-of-the-art models representative of current approaches in CSLR and SLT.

SLR baseline: Stochastic CSLR

Stochastic CSLR [Niu & Mak, 2020] is presented as a state-of-the-art, end-to-end trainable model for CSLR, based on a Transformer encoder [Vaswani et al., 2017] and a Connectionist Temporal Classification (CTC) [Graves et al., 2006] decoder. The architecture, depicted in Figure 7.1, incorporates three proposed stochastic components: a stochastic frame dropping mechanism, a stochastic gradient stopping method, and stochastic fine-grained labeling.

Figure 7.1: Stochastic CSLR architecture overview [Niu & Mak, 2020].



The **Stochastic frame dropping (SFD)** mechanism randomly discards a fixed proportion of video frames during training, which helps alleviate overfitting and reduces computational load (memory and time). The **Stochastic gradient stopping (SGS)** method selectively stops gradient back-propagation during visual feature extraction for a random subset of frames, also aimed at reducing overfitting and computational cost. The **Stochastic fine-grained labeling (SFL)** component represents each sign gloss using a variable number of internal states, modeled as a categorical random variable. This allows the network to learn appropriate temporal durations for different signs within the CTC framework, facilitating more discriminative feature learning. Different state sequences are sampled during training to optimize the CTC loss. For visual feature extraction, Stochastic CSLR typically utilizes a ResNet18 [He et al., 2016a] model pre-trained on ImageNet [Russakovsky et al., 2015b]. This model was chosen as a strong baseline due to its reported state-of-the-art performance on standard CSLR benchmarks at the time of experimentation.

SLR baseline: SignGraph

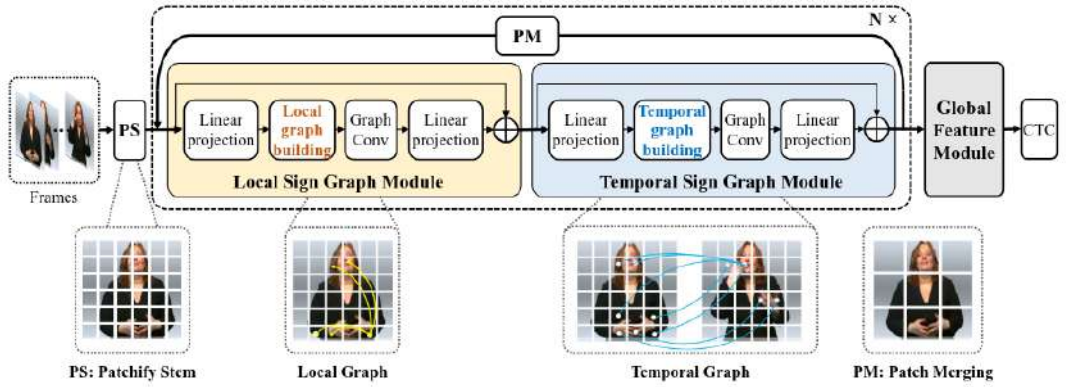
Gan et al. [Gan et al., 2024] propose SignGraph, a novel CSLR architecture based on graph neural networks. The core idea involves representing each frame of a sign language sequence as a graph, where nodes correspond to image patches and edges capture relationships between these patches. The architecture comprises two key modules: 1. **Local Sign Graph (LSG) Module:** This module dynamically constructs graphs within each frame based on node features, aiming to learn intra-frame correlations between different regions (e.g., relating hand movements to facial expressions). 2. **Temporal Sign Graph (TSG) Module:** This module captures inter-frame relationships by dynamically connecting relevant regions (nodes) between adjacent frames, designed to track the dynamic movements essential to signing.

SignGraph employs a multi-scale approach, using different patch sizes to capture features at various granularities. Experiments reported by the authors on public datasets (PHOENIX14, PHOENIX14T, CSL-Daily) demonstrated competitive performance compared to other state-of-the-art models, notably without relying on explicit skeleton or depth data. The graph-based approach offers a different paradigm for capturing complex spatio-temporal relationships inherent in sign language compared to traditional CNN/RNN or Transformer models. Figure 7.5 shows an overview .

SLT baseline: TSPNet

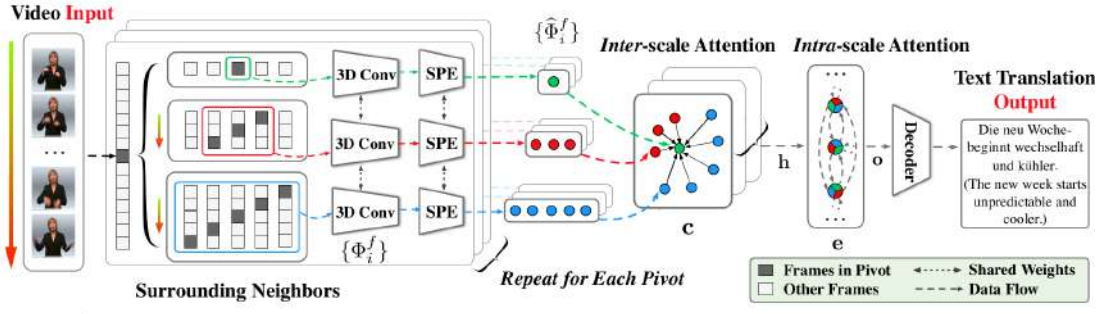
For the Sign Language Translation (SLT) task (Video-to-Text), the TSPNet architecture [Li et al., 2020b] was selected as a representative state-of-the-art baseline. TSPNet utilizes an encoder-decoder structure and introduces inter-scale and intra-scale attention mechanisms designed to enhance semantic consistency and resolve ambiguity. Its main

Figure 7.2: SignGraph architecture overview [Gan et al., 2024].



components, illustrated in Figure 7.5, are: 1. **Multi-scale Segment Representation:** This addresses potential inaccuracies in sign video segmentation by using a sliding window approach to create overlapping video segments of multiple durations. 2. **Hierarchical Video Feature Learning:** The encoder employs hierarchical learning, leveraging local temporal structure and non-local video context through attention mechanisms to learn discriminative representations. These representations are then fed to a Transformer decoder to generate the target text translation. The overall approach aims to learn both spatial and temporal semantics effectively. TSPNet was chosen as it represented a strong baseline focused specifically on the video-to-text SLT task.

Figure 7.3: TSPNet architecture overview [Li et al., 2020b].

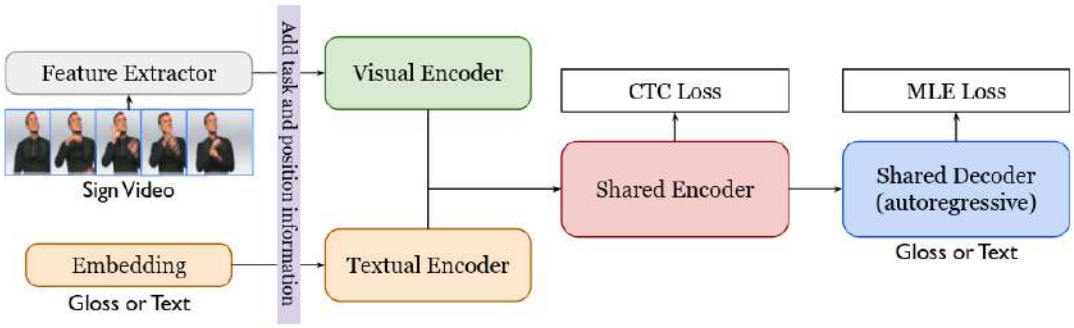


SLT baseline: SLTUNET

Zhang et al. [Zhang et al., 2023] introduced SLTUNET, a unified model designed to handle multiple related SLT tasks simultaneously: sign-to-gloss recognition, gloss-to-text translation, and direct sign-to-text translation. As shown in Figure 7.5 , the architecture leverages shared parameters to facilitate knowledge transfer across different tasks and input/output modalities, while also incorporating modality-specific components. A key advantage highlighted is the ability to benefit from external resources, such as large machine translation datasets, potentially alleviating data scarcity

issues common in SLT. The authors report competitive results on standard benchmarks and promising performance on the challenging DGS Corpus. This multi-task approach represents an interesting direction for improving SLT, especially for low-resource languages where leveraging related tasks or data might be beneficial.

Figure 7.4: SLTUNET architecture overview [Zhang et al., 2023].



SLT baseline: GloFE

Lin et al. [Lin et al., 2023] proposed the GloFE framework to tackle SLT, particularly addressing the challenge of limited gloss annotations. Illustrated in Figure 7.5 , the core idea is to circumvent the need for glosses entirely during translation. It extracts conceptual words (like nouns, verbs) from the target spoken language translations and uses their embeddings as queries in a cross-attention mechanism applied to visual features extracted from the sign video. The entire model, including the visual backbone, is trained end-to-end without any gloss supervision. GloFE achieved state-of-the-art results on large datasets like OpenASL, demonstrating its potential for bridging the sign-to-text modality gap directly. This gloss-free approach is highly relevant as generating accurate, time-aligned gloss annotations is often a major bottleneck in creating SLT datasets.

7.3.2 Experimental setup

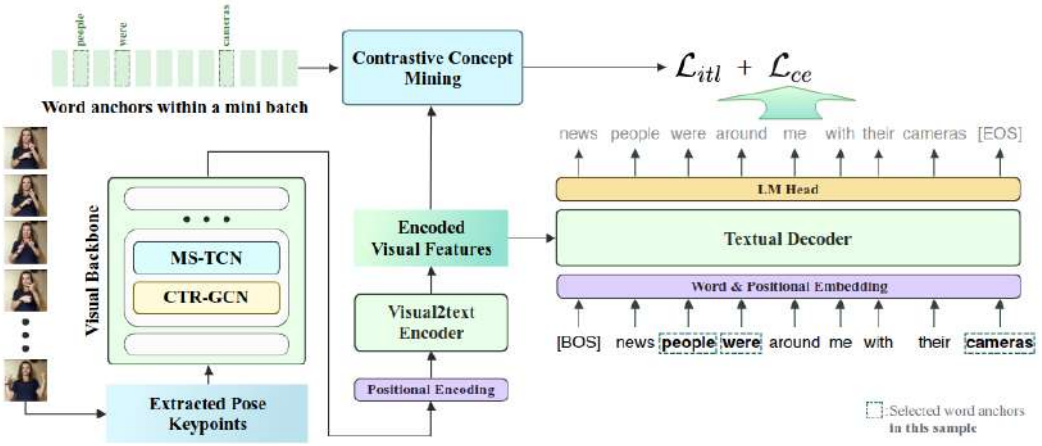
Metrics

Model performance for SLR and SLT tasks is commonly evaluated using the following two metrics:

Word Error Rate (WER) [Koller et al., 2015] is the standard metric reported for SLR baselines. It measures the minimum number of edits (substitutions S, deletions D, insertions I) required to transform the predicted sequence into the reference sequence, normalized by the number of words (N) in the reference:

$$WER = \frac{S + D + I}{N}$$

Figure 7.5: GloFE architecture overview [Lin et al., 2023].



Lower WER indicates better performance.

Bilingual Evaluation Understudy (BLEU) [Papineni et al., 2002b] is the standard metric reported for SLT baselines. It measures the precision of n-grams (sequences of n words) in the predicted translation compared to reference translations, incorporating a penalty for overly short predictions. The n-gram precision p_n is calculated as:

$$p_n = \frac{\sum_{C \in \{Candidates\}} \sum_{n\text{-}gram \in C} Count_{clip}(n\text{-}gram)}{\sum_{C' \in \{Candidates\}} \sum_{n\text{-}gram' \in C'} Count(n\text{-}gram')},$$

where $Count_{clip}$ is the clipped count of n-grams in the candidate matching any reference, and $Count$ is the total count of n-grams in the candidate. The Brevity Penalty (BP) penalizes candidates shorter than the references (c = candidate length, r = reference length):

$$BP = \begin{cases} 1, & \text{if } c > r \\ e^{1-\frac{r}{c}}, & \text{if } c \leq r \end{cases}.$$

The final BLEU-n score combines precision scores (typically up to N=4) with the brevity penalty, usually using geometric averaging with uniform weights ($w_i = 1/N$):

$$BLEU\text{-}N = BP \cdot \exp\left(\sum_{i=1}^{N} w_i \log p_i\right).$$

Higher BLEU scores indicate better translation quality. We report BLEU-1, BLEU-2, BLEU-3, and BLEU-4.

Training details

Both the Stochastic CSLR [Niu & Mak, 2020] and TSPNet [Li et al., 2020b] baseline models were implemented using the PyTorch [Paszke et al., 2019c] framework. Model training was performed using Tesla V100 GPUs.

For Stochastic CSLR, input video frames were resized to 256x256, and random crops of size 224x224 were extracted during training. The model was trained using the Adam optimizer with a batch size of 8 for 30 epochs. The learning rate (η_i for epoch i) was scheduled according to $\eta_i = \eta_0 \cdot 0.95^{\lfloor i/2 \rfloor}$, starting with $\eta_0 = 1 \times 10^{-4}$. During training, 50

The TSPNet model was developed using the FAIRSEQ [Ott et al., 2019] sequence modeling toolkit within PyTorch [Paszke et al., 2019c]. Visual features were extracted using an I3D network [Carreira & Zisserman, 2017a] pre-trained on the Kinetics dataset. The model was trained using the Adam optimizer with an initial learning rate of 10^{-4} for a maximum of 200 epochs. The learning rate schedule included a reduction factor of 0.5 with a patience of 8 epochs (reducing LR if validation performance plateaued). Label smoothing [Szegedy et al., 2016] (weight factor 0.1) and weight decay (10^{-4}) were applied for regularization.

7.3.3 SLR results

Table 7.1 presents the CSLR results obtained by applying the Stochastic CSLR [Niu & Mak, 2020] baseline model to the different splits of the FluentSigners-50 dataset, reported in terms of Word Error Rate (WER; lower is better). For comparison, results on the standard RWTH-PHOENIX-Weather 2014T [Cihan Camgoz et al., 2018] benchmark, as reported in the original Stochastic CSLR paper [Niu & Mak, 2020], are also included.

Table 7.1: SLR results of Stochastic CSLR[Niu & Mak, 2020] on RWTH-PHOENIX-Weather 2014T [Cihan Camgoz et al., 2018] and different splits of FluentSigners-50.

DATASET	VAL (WER)	TEST (WER)
FLUENTSIGNERS-50: SPLIT 1	25.4 ± 2.8	24.9 ± 6.2
FLUENTSIGNERS-50: SPLIT 1 (1 FOLD)	21.8	31.7
FLUENTSIGNERS-50: SPLIT 2	10.6	47.1
FLUENTSIGNERS-50: SPLIT 3	–	52.0 ± 4.68
FLUENTSIGNERS-50: SPLIT 3 (1 FOLD)	–	48.7
RWTH-PHOENIX-WEATHER 2014T	25.1	26.1

We report results from 5-fold cross-validation where feasible (Splits 1 and 3) to provide more robust estimates and avoid potential bias from a single test set. However, given the computational expense of 5-fold cross-validation on video data, we also present results from a single fold evaluation, similar to practices often used for benchmarks like RWTH-PHOENIX-Weather 2014T [Cihan Camgoz et al., 2018] and CSL [Huang et al., 2018a] which have fixed standard test sets.

The results obtained on Split 2 (Age Independence) of FluentSigners-50 provide insights into model generalization across age groups. The high WER on the test set

(47.1) compared to Split 1 suggests significant difficulty in generalizing recognition to child signers when the model is trained and validated primarily on adult signers. Qualitative observations indicated that many child participants appeared less confident on camera, which might contribute to performance degradation compared to Split 1. Interestingly, the validation set WER for Split 2 (10.6, using adult validation data) was much lower than for Split 1 (around 25.4), possibly because excluding child signers from the training data removed a source of noise or variability, leading to better fitting on the adult validation set. This highlights the distinct challenge posed by age variation in signing.

Results on Split 3 (Unseen Sentences and Signer Independence) assess generalization to novel linguistic contexts performed by unseen signers. The high test WER (around 52.0) confirms this split is considerably more challenging than Split 1. This difficulty likely arises because the test sentences, although composed of signs seen during training, present them in new sequential orders and sentence structures not encountered before by the model, combined with the challenge of recognizing them from unfamiliar signers. The relatively small number of unique sentences (173) in the dataset means each sign gloss might only be seen in a limited number of contexts during training. This split effectively tests the model's ability to handle linguistic variability and co-articulation beyond simple signer adaptation.

7.3.4 SLT results

Table 7.2 presents the SLT task results, evaluating the TSPNet [Li et al., 2020b] baseline model on the FluentSigners-50 splits and the RWTH-PHOENIX-Weather 2014T [Cihan Camgoz et al., 2018] benchmark. Performance is measured using BLEU scores (BLEU-1 to BLEU-4; higher is better). Results for RWTH-PHOENIX-Weather 2014T are taken directly from the original TSPNet paper [Li et al., 2020b].

Table 7.2: SLT results of TSPNet[Li et al., 2020b] on RWTH-PHOENIX-Weather 2014T[Cihan Camgoz et al., 2018] and different splits of FluentSigners-50.

Dataset	BLEU-1	BLEU-2	BLEU-3	BLEU-4
FLUENTSIGNERS-50: SPLIT 1	20.3 ± 1.0	17.8 ± 0.9	16.6 ± 0.8	16.0 ± 0.8
FLUENTSIGNERS-50: SPLIT 1 (1 FOLD)	0.7	18.0	16.7	15.7
FLUENTSIGNERS-50: SPLIT 2	14.2	12.0	11.0	10.5
FLUENTSIGNERS-50: SPLIT 3	5.1 ± 0.45	3.9 ± 0.53	3.1 ± 0.78	2.0 ± 1.1
FLUENTSIGNERS-50: SPLIT 3 (1 FOLD)	5.1	4.1	3.0	2.2
RWTH-PHOENIX-WEATHER 2014T	36.1	23.1	16.9	13.4

Focusing on BLEU-4, the most challenging metric commonly used for comparison, the results show a similar pattern to the SLR findings: Split 1 yields the best performance (16.0 BLEU-4), followed by Split 2 (10.5), with Split 3 proving the most difficult

(2.0). This consistency across tasks further validates the relative difficulty posed by the different generalization challenges targeted by each split. Achieving reasonable BLEU scores on Split 1 demonstrates the potential for SLT on this dataset, while the lower scores on Splits 2 and 3 highlight areas needing significant improvement, likely requiring models better able to handle age variation and linguistic context variability.

7.4 Chapter Conclusion

The results obtained on Split 1 of the FluentSigners-50 dataset provide a benchmark for model generalization to unseen signers under varied camera and environmental conditions. This contrasts with datasets like RWTH-PHOENIX-Weather 2014T, which features only 9 signers recorded under consistent professional studio conditions. Models trained solely on such homogeneous data may overfit and perform poorly when deployed in real-world settings with diverse signers and conditions. FluentSigners-50, therefore, aims to address this limitation by offering data reflecting large signer variety recorded "in the wild". This allows for the development and evaluation of more robust and practically applicable SLP solutions. We encourage researchers to report performance on both RWTH-PHOENIX-Weather 2014T and FluentSigners-50 (Split 1) to demonstrate robustness to these differing conditions.

Consistent with the SLR results, the SLT evaluations using TSPNet confirm the relative difficulty of the proposed splits, with performance degrading from Split 1 (signer independence) to Split 2 (age independence) and further to Split 3 (unseen sentences and signers). This underscores the significant challenges posed by generalizing across age groups and, particularly, across novel linguistic contexts combined with unfamiliar signers, using current state-of-the-art models on this dataset. Future work should investigate model architectures or training strategies specifically designed to address the difficulties highlighted by Splits 2 and 3, potentially involving age adaptation techniques or methods better modeling linguistic context and co-articulation.

BLANK

Chapter 8

Conclusion

8.1 Summary of the Thesis

This thesis aimed to address significant challenges in the field of large-vocabulary Kazakh Sign Language (KSL) Processing. The core objectives were centered around key areas requiring advancement: collecting representative corpora, developing methods for semi-automatic annotation, improving sign language representation, and establishing baseline recognition and translation capabilities. The research was primarily motivated by identified gaps in existing resources, including the scarcity of continuous sign language datasets for KSL, the limited linguistic and environmental variability in many available corpora, the frequent underutilization of crucial non-manual components in recognition systems, and the lack of efficient tools to overcome the laborious manual annotation process. Overall, this work sought to create foundational large-vocabulary, signer-independent KSL datasets and develop effective, corresponding methods and tools to advance KSL processing research.

8.2 Key Findings and Contributions

This research resulted in several significant contributions to the field of Kazakh Sign Language Processing:

- **Dataset Collection:** We created and released the first large-scale datasets specifically designed for continuous KSL processing: K-RSL: FluentSigners-50 (Chapter 3) and K-RSL: OnlineSchool (Chapter 4). These datasets address critical limitations of prior resources by incorporating continuous signing examples, high signer variability (including diverse demographics and fluency levels), and data collected under realistic conditions (community crowdsourcing and online broadcasts). These corpora, totaling over 900 hours of video with associated annotations (transcripts, partial glosses), provide essential resources for training and evaluating deep learning models for KSL recognition and translation.

- **Sign Language Representation Framework:** We proposed and investigated a framework for representing both manual and non-manual components of sign

languages (Chapter 5). This included exploring computational methods for handshape classification (unsupervised and supervised) and analyzing the importance of non-manual features (head/body movements, facial expressions, mouthing) through baseline recognition experiments, demonstrating their positive impact on accuracy. This work underscores the need for holistic sign representation beyond just manual features.

- **Semi-Automatic Annotation Tool:** To address the annotation bottleneck, we developed SLAN-tool, an open-source, web-based tool designed for semi-automatic annotation of sign language videos (Chapter 6). This tool integrates neural network models capable of automatically recognizing signing segments and classifying handshapes, thereby facilitating and potentially accelerating the annotation process for large datasets. Providing practical, open-source tools can significantly benefit the wider sign language research community.

- **Baseline Recognition and Translation Results:** We established initial benchmark results for KSL recognition and translation by applying state-of-the-art deep learning models to the newly created datasets, particularly FluentSigners-50 (Chapter 7). Analyzing performance across different evaluation splits (testing signer independence, age independence, and unseen sentences) provided insights into the capabilities of current models and highlighted specific challenges posed by realistic KSL data. These baseline results serve as a crucial reference point for future algorithmic development for KSL.

8.3 Future Work and Open Problems

While this thesis provides significant foundational contributions for KSL Processing, several avenues for future research and open problems remain:

- **Dataset Expansion and Enrichment:** The collected datasets, though substantial, could be further expanded to include a broader range of signers (more demographic diversity, regional dialects), different signing styles (e.g., more conversational, unprompted data), and richer annotations (e.g., detailed non-manual features, full glossing for OnlineSchool). Enhancing data diversity and annotation detail would further improve the robustness and generalizability of trained models.

- **Refinement of Representation Models:** The proposed frameworks for manual and non-manual representation can be advanced. This includes exploring more sophisticated techniques for handshape classification (perhaps robust to viewpoint

changes) and developing models that better capture the complex interplay and temporal dynamics of non-manual features. Integrating contextual information more effectively and utilizing cutting-edge deep learning architectures could lead to more accurate and nuanced sign representations.

- **Enhancement of Annotation Tools:** The SLAN-tool platform could be extended with additional features. Possibilities include incorporating automatic recognition modules for various non-manual components, integrating machine translation capabilities to assist annotators, further improving the user interface based on wider feedback, and adding features to support collaborative annotation projects more effectively.

- **Exploration of Advanced SLP Models:** Research building upon the established baselines can explore more advanced deep learning architectures for KSL recognition and translation. Areas include investigating different types of transformer-based models, exploring graph neural networks for capturing relational structure, improving multi-modal fusion techniques, and developing robust end-to-end translation models, potentially leveraging the partial annotations available in the OnlineSchool dataset.

- **Real-World Applications and Evaluation:** The datasets, models, and tools developed in this thesis provide a basis for building practical applications. Future work should focus on deploying and evaluating these technologies in real-world settings, such as developing prototype sign language interpretation systems, creating accessible educational platforms for KSL users, or building other assistive tools. Rigorous evaluation in realistic scenarios is crucial for assessing true impact and guiding further refinement.

8.4 Concluding Remarks

This thesis has made substantial contributions to the advancement of Kazakh Sign Language Processing by directly addressing critical challenges related to data scarcity, feature representation, annotation efficiency, and baseline system development. The research has resulted in the creation of valuable, large-scale datasets, the development of effective methods and tools for analysis and annotation, and the demonstration of initial promising results on recognition and translation tasks for continuous KRSL. The findings and resources presented herein hold the potential to significantly impact future research in low-resource sign language processing and contribute towards technologies that can facilitate communication and information access for Deaf and hard-of-hearing individuals in Kazakhstan and beyond. The future research directions identified provide

a roadmap for building upon this foundation to develop increasingly sophisticated and impactful sign language processing technologies.

Bibliography

[Abadi et al., 2016] Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., Devin, M., Ghemawat, S., Irving, G., Isard, M., Kudlur, M., Levenberg, J., Monga, R., Moore, S., Murray, D. G., Steiner, B., Tucker, P., Vasudevan, V., Warden, P., Wicke, M., Yu, Y., & Zheng, X. (2016). TensorFlow: A System for Large-Scale Machine Learning. In *Proceedings of the 12th USENIX Conference on Operating Systems Design and Implementation*, OSDI'16 (pp. 265–283). USA: USENIX Association.

[Albanie et al., 2020a] Albanie, S., Varol, G., Momeni, L., Afouras, T., Chung, J. S., Fox, N., & Zisserman, A. (2020a). BSL-1K: Scaling up co-articulated sign language recognition using mouthing cues. In *European Conference on Computer Vision*.

[Albanie et al., 2020b] Albanie, S., Varol, G., Momeni, L., Afouras, T., Chung, J. S., Fox, N., & Zisserman, A. (2020b). Bsl-1k: Scaling up co-articulated sign language recognition using mouthing cues. In *European Conference on Computer Vision* (pp. 35–53).: Springer.

[Allen, 2015] Allen, T. E. (2015). The deaf community as a "special linguistic demographic". In *Research Methods in Sign Language Studies*.

[Aly & Aly, 2020] Aly, S. & Aly, W. (2020). Deeparslr: A novel signer-independent deep learning framework for isolated arabic sign language gestures recognition. *IEEE Access*, 8, 83199–83212.

[Belissen et al., 2020] Belissen, V., Braffort, A., & Gouiffès, M. (2020). Dicta-sign-lsf-v2: remake of a continuous french sign language dialogue corpus and a first baseline for automatic sign language processing. In *LREC 2020, 12th Conference on Language Resources and Evaluation*.

[Benchiheub et al., 2016] Benchiheub, M., Berret, B., & Braffort, A. (2016). Collecting and analysing a motion-capture corpus of french sign language. In *7th International Conference on Language Resources and Evaluation-Workshop on the Representation and Processing of Sign Languages (LREC-WRPSL 2016), Portoroz, Slovenia* (pp. 7–12).

[Bragg et al., 2019a] Bragg, D., Koller, O., Bellard, M., Berke, L., Boudreault, P., Braffort, A., Caselli, N., Huenerfauth, M., Kacorri, H., Verhoef, T., et al. (2019a). Sign Language Recognition, Generation, and Translation: An Interdisciplinary

Perspective. In *The 21st International ACM SIGACCESS Conference on Computers and Accessibility* (pp. 16–31).: ACM.

[Bragg et al., 2019b] Bragg, D., Koller, O., Bellard, M., Berke, L., Boudreault, P., Braffort, A., Caselli, N., Huenerfauth, M., Kacorri, H., Verhoef, T., Vogler, C., & Ringel Morris, M. (2019b). Sign language recognition, generation, and translation: An interdisciplinary perspective. In *The 21st International ACM SIGACCESS Conference on Computers and Accessibility*, ASSETS '19 (pp. 16–31). New York, NY, USA: ACM.

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

[Cao et al., 2019] Cao, Z., Hidalgo, G., Simon, T., Wei, S.-E., & Sheikh, Y. (2019). Openpose: realtime multi-person 2d pose estimation using part affinity fields. *IEEE transactions on pattern analysis and machine intelligence*, 43(1), 172–186.

[Cao et al., 2019] Cao, Z., Hidalgo Martinez, G., Simon, T., Wei, S., & Sheikh, Y. A. (2019). Openpose: Realtime multi-person 2d pose estimation using part affinity fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

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

[Carreira & Zisserman, 2017a] Carreira, J. & Zisserman, A. (2017a). Quo vadis, action recognition? a new model and the kinetics dataset. In *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 6299–6308).

[Carreira & Zisserman, 2017b] Carreira, J. & Zisserman, A. (2017b). Quo vadis, action recognition? a new model and the kinetics dataset. In *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 6299–6308).

[Chaaban et al., 2021] Chaaban, H., Gouiffès, M., & Braffort, A. (2021). Automatic annotation and segmentation of sign language videos: Base-level features and lexical signs classification. In *VISIGRAPP (5: VISAPP)* (pp. 484–491).

[Chai et al., 2014a] Chai, X., Wang, H., & Chen, X. (2014a). The devisign large vocabulary of chinese sign language database and baseline evaluations. In *Technical report VIPL-TR-14-SLR-001. Key Lab of Intelligent Information Processing of Chinese Academy of Sciences (CAS)*. Institute of Computing Technology.

[Chai et al., 2014b] Chai, X., Wang, H., & Chen, X. (2014b). The devisign large vocabulary of chinese sign language database and baseline evaluations. *Technical report VIPL-TR-14-SLR-001. Key Lab of Intelligent Information Processing of Chinese Academy of Sciences (CAS), Institute of Computing Technology, CAS*.

[Chatzis et al., 2020] Chatzis, T., Stergioulas, A., Konstantinidis, D., Dimitropoulos, K., & Daras, P. (2020). A comprehensive study on deep learning-based 3d hand pose estimation methods. *Applied Sciences*, 10(19), 6850.

[Chen et al., 2017] Chen, S., Chen, J., Jin, Q., & Hauptmann, A. (2017). Video captioning with guidance of multimodal latent topics. In *Proceedings of the 25th ACM international conference on Multimedia* (pp. 1838–1846).

[Chételat-Pelé & Braffort, 2008] Chételat-Pelé, É. & Braffort, A. (2008). Sign language corpus annotation: Toward a new methodology. In *Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC'08)*.

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

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

[Dalal & Triggs, 2005] Dalal, N. & Triggs, B. (2005). Histograms of oriented gradients for human detection.

[Dong et al., 2015a] Dong, C., Leu, M. C., & Yin, Z. (2015a). American sign language alphabet recognition using microsoft kinect. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops* (pp. 44–52).

[Dong et al., 2015b] Dong, C., Leu, M. C., & Yin, Z. (2015b). American sign language alphabet recognition using microsoft kinect. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops* (pp. 44–52).

[Dreuw et al., 2008] Dreuw, P., Neidle, C., Athitsos, V., Sclaroff, S., & Ney, H. (2008). Benchmark databases for video-based automatic sign language recognition. In *Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC'08)* Marrakech, Morocco: European Language Resources Association (ELRA).

[Duan et al., 2019] Duan, K., Bai, S., Xie, L., Qi, H., Huang, Q., & Tian, Q. (2019). Centernet: Keypoint triplets for object detection. In *Proceedings of the IEEE International Conference on Computer Vision* (pp. 6569–6578).

[Duarte et al., 2020] Duarte, A., Palaskar, S., Ventura, L., Ghadiyaram, D., DeHaan, K., Metze, F., Torres, J., & Giro-i Nieto, X. (2020). How2sign: a large-scale multimodal dataset for continuous american sign language. *arXiv preprint arXiv:2008.08143*.

[Fragkiadakis et al., 2021] Fragkiadakis, M., Nyst, V., & van der Putten, P. (2021). Towards a user-friendly tool for automated sign annotation: identification and annotation of time slots, number of hands, and handshape. *Digital Humanities Quarterly (DHQ)*, 15(1).

[Gan et al., 2024] Gan, S., Yin, Y., Jiang, Z., Wen, H., Xie, L., & Lu, S. (2024). Signgraph: A sign sequence is worth graphs of nodes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 13470–13479).

[Gattupalli et al., 2016] Gattupalli, S., Ghaderi, A., & Athitsos, V. (2016). Evaluation of deep learning based pose estimation for sign language recognition. In *Proceedings of the 9th ACM International Conference on PErvasive Technologies Related to Assistive Environments* (pp. 1–7).

[Ghadiyaram et al., 2019] Ghadiyaram, D., Tran, D., & Mahajan, D. (2019). Large-scale weakly-supervised pre-training for video action recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 12046–12055).

[Graves et al., 2006] Graves, A., Fernández, S., Gomez, F., & Schmidhuber, J. (2006). Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks. In *Proceedings of the 23rd international conference on Machine learning* (pp. 369–376).: ACM.

[Hall et al., 2022] Hall, K. C., Aonuki, Y., Vesik, K., Poy, A., & Tolmie, N. (2022). Sign language phonetic annotator-analyzer: Open-source software for form-based analysis of sign languages. In *Proceedings of the LREC2022 10th Workshop on the Representation and Processing of Sign Languages: Multilingual Sign Language Resources* (pp. 59–66).

[Hall et al., 2017] Hall, K. C., Mackie, S., Fry, M., & Tkachman, O. (2017). Slpannotator: Tools for implementing sign language phonetic annotation. In *INTERSPEECH* (pp. 2083–2087).

[Hanke, 2004] Hanke, T. (2004). Hamnosys: representing sign language data in language resources and language processing contexts. In O. Streiter & C. Vettori (Eds.), *LREC 2004, Workshop proceedings: Representation and processing of sign languages.* (pp. 1–6). Paris.

[He et al., 2017] He, K., Gkioxari, G., Dollár, P., & Girshick, R. (2017). Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision* (pp. 2961–2969).

[He et al., 2016a] He, K., Zhang, X., Ren, S., & Sun, J. (2016a). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 770–778).

[He et al., 2016b] He, K., Zhang, X., Ren, S., & Sun, J. (2016b). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 770–778).

[Hernandez Ruiz et al., 2017] Hernandez Ruiz, A., Porzi, L., Rota Bulò, S., & Moreno-Noguer, F. (2017). 3d cnns on distance matrices for human action recognition. In *Proceedings of the 25th ACM international conference on Multimedia* (pp. 1087–1095).

[Huang et al., 2017] Huang, G., Liu, Z., Van Der Maaten, L., & Weinberger, K. Q. (2017). Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 4700–4708).

[Huang et al., 2018a] Huang, J., Zhou, W., Zhang, Q., Li, H., & Li, W. (2018a). Video-based sign language recognition without temporal segmentation. *arXiv preprint arXiv:1801.10111*.

[Huang et al., 2018b] Huang, J., Zhou, W., Zhang, Q., Li, H., & Li, W. (2018b). Video-based sign language recognition without temporal segmentation. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

[Imashev et al., 2020] Imashev, A., Mukushev, M., Kimmelman, V., & Sandygulova, A. (2020). A dataset for linguistic understanding, visual evaluation, and recognition of sign languages: The k-rsl. In *Proceedings of the 24th Conference on Computational Natural Language Learning* (pp. 631–640).

[Jedlička et al., 2022] Jedlička, P., Krňoul, Z., Železnỳ, M., & Müller, L. (2022). Mc-trislan: A large 3d motion capture sign language data-set.

[Ji et al., 2012] Ji, S., Xu, W., Yang, M., & Yu, K. (2012). 3d convolutional neural networks for human action recognition. *IEEE transactions on pattern analysis and machine intelligence*, 35(1), 221–231.

[Johnson et al., 2019] Johnson, J., Douze, M., & Jégou, H. (2019). Billion-scale similarity search with gpus. *IEEE Transactions on Big Data*.

[Joze & Koller, 2018] Joze, H. R. V. & Koller, O. (2018). Ms-asl: A large-scale data set and benchmark for understanding american sign language. *arXiv preprint arXiv:1812.01053*.

[Kanis, 2008] Kanis, J. (2008). Interactive hamnosys notation editor for signed speech annotation.

[Kapitanov et al., 2023] Kapitanov, A., Karina, K., Nagaev, A., & Elizaveta, P. (2023). Slovo: Russian sign language dataset. In *International Conference on Computer Vision Systems* (pp. 63–73).: Springer.

[Kay et al., 2017] Kay, W., Carreira, J., Simonyan, K., Zhang, B., Hillier, C., Vijayanarasimhan, S., Viola, F., Green, T., Back, T., Natsev, P., et al. (2017). The kinetics human action video dataset. *arXiv preprint arXiv:1705.06950*.

[Kimmelman et al., 2020] Kimmelman, V., Imashev, A., Mukushev, M., & Sandygulova, A. (2020). Eyebrow position in grammatical and emotional expressions in kazakh-russian sign language: A quantitative study. *PLOS ONE*, 15(6), 1–16.

[Klezovich, 2019] Klezovich, A. (2019). *Automatic Extraction of Phonemic Inventory in Russian Sign Language*. BA thesis, HSE, Moscow.

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

[Koller, 2020a] Koller, O. (2020a). Quantitative Survey of the State of the Art in Sign Language Recognition. *arXiv preprint arXiv:2008.09918*.

[Koller, 2020b] Koller, O. (2020b). Quantitative survey of the state of the art in sign language recognition. *arXiv preprint arXiv:2008.09918*.

[Koller et al., 2019] Koller, O., Camgoz, C., Ney, H., & Bowden, R. (2019). Weakly supervised learning with multi-stream cnn-lstm-hmms to discover sequential parallelism in sign language videos. *IEEE transactions on pattern analysis and machine intelligence*.

[Koller et al., 2015] Koller, O., Forster, J., & Ney, H. (2015). Continuous sign language recognition: Towards large vocabulary statistical recognition systems handling multiple signers. *Computer Vision and Image Understanding*, 141, 108–125.

[Koller et al., 2016] Koller, O., Ney, H., & Bowden, R. (2016). Deep hand: How to train a cnn on 1 million hand images when your data is continuous and weakly

labelled. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 3793–3802).

[Kopf et al., 2021] Kopf, M., Schulder, M., Hanke, T., & Hénault-Tessier, M. (2021). D6. 1 overview of datasets for the sign languages of europe.

[Krizhevsky et al., 2012] Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems* (pp. 1097–1105).

[Kubuş, 2008] Kubuş, O. (2008). *An Analysis of Turkish Sign Language Phonology and Morphology*. Diploma thesis, Middle East Technical University, Ankara.

[Li et al., 2020a] Li, D., Rodriguez, C., Yu, X., & Li, H. (2020a). Word-level deep sign language recognition from video: A new large-scale dataset and methods comparison. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision* (pp. 1459–1469).

[Li et al., 2020b] Li, D., Xu, C., Yu, X., Zhang, K., Swift, B., Suominen, H., & Li, H. (2020b). Tspnet: Hierarchical feature learning via temporal semantic pyramid for sign language translation. *arXiv preprint arXiv:2010.05468*.

[Liao et al., 2019] Liao, Y., Xiong, P., Min, W., Min, W., & Lu, J. (2019). Dynamic sign language recognition based on video sequence with blstm-3d residual networks. *IEEE Access*, 7, 38044–38054.

[Lin et al., 2023] Lin, K., Wang, X., Zhu, L., Sun, K., Zhang, B., & Yang, Y. (2023). Gloss-free end-to-end sign language translation. *arXiv preprint arXiv:2305.12876*.

[Lu et al., 2016] Lu, J., Jones, A., & Morgan, G. (2016). The impact of input quality on early sign development in native and non-native language learners. *Journal of Child Language*, 43(3), 537–552.

[Lu & Huenerfauth, 2010] Lu, P. & Huenerfauth, M. (2010). Collecting a motion-capture corpus of american sign language for data-driven generation research. In *Proceedings of the NAACL HLT 2010 Workshop on Speech and Language Processing for Assistive Technologies* (pp. 89–97).: Association for Computational Linguistics.

[Mukai et al., 2017] Mukai, N., Harada, N., & Chang, Y. (2017). Japanese fingerspelling recognition based on classification tree and machine learning. In *2017 Nicograph International (NicoInt)* (pp. 19–24).: IEEE.

[Mukushev et al., 2022] Mukushev, M., Ubingazhibov, A., Kydyrbekova, A., Imashev, A., Kimmelman, V., & Sandygulova, A. (2022). Fluentsigners-50: A signer

independent benchmark dataset for sign language processing. *Plos one*, 17(9), e0273649.

[Narasimhaswamy et al., 2019] Narasimhaswamy, S., Wei, Z., Wang, Y., Zhang, J., & Hoai, M. (2019). Contextual attention for hand detection in the wild. *arXiv preprint arXiv:1904.04882*.

[Neidle et al., 2012] Neidle, C., Thangali, A., & Sclaroff, S. (2012). Challenges in development of the american sign language lexicon video dataset (asllvd) corpus. In *5th workshop on the representation and processing of sign languages: interactions between corpus and Lexicon, LREC*: Citeseer.

[Niu & Mak, 2020] Niu, Z. & Mak, B. (2020). Stochastic fine-grained labeling of multi-state sign glosses for continuous sign language recognition. In *European Conference on Computer Vision* (pp. 172–186).: Springer.

[Nyst, 2007] Nyst, V. (2007). *A Descriptive Analysis of Adamorobe Sign Language (Ghana)*. Utrecht: LOT.

[Orbay & Akarun, 2020] Orbay, A. & Akarun, L. (2020). Neural sign language translation by learning tokenization. *arXiv preprint arXiv:2002.00479*.

[Oszust & Wysocki, 2013] Oszust, M. & Wysocki, M. (2013). Polish sign language words recognition with kinect. In *2013 6th International Conference on Human System Interactions (HSI)* (pp. 219–226).: IEEE.

[Ott et al., 2019] Ott, M., Edunov, S., Baevski, A., Fan, A., Gross, S., Ng, N., Grangier, D., & Auli, M. (2019). fairseq: A fast, extensible toolkit for sequence modeling. *arXiv preprint arXiv:1904.01038*.

[Papineni et al., 2002a] Papineni, K., Roukos, S., Ward, T., & Zhu, W.-J. (2002a). Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics* (pp. 311–318).

[Papineni et al., 2002b] Papineni, K., Roukos, S., Ward, T., & Zhu, W.-J. (2002b). Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting on association for computational linguistics* (pp. 311–318).: Association for Computational Linguistics.

[Paszke et al., 2019a] Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., & Chintala, S. (2019a). Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E.

Fox, & R. Garnett (Eds.), *Advances in Neural Information Processing Systems 32* (pp. 8024–8035). Curran Associates, Inc.

[Paszke et al., 2019b] Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Köpf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., & Chintala, S. (2019b). *PyTorch: An Imperative Style, High-Performance Deep Learning Library*. Curran Associates Inc.: Red Hook, NY, USA.

[Paszke et al., 2019c] Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., et al. (2019c). Pytorch: An imperative style, high-performance deep learning library. *arXiv preprint arXiv:1912.01703*.

[Paudyal et al., 2019] Paudyal, P., Lee, J., Banerjee, A., & Gupta, S. K. (2019). A comparison of techniques for sign language alphabet recognition using armband wearables. *ACM Transactions on Interactive Intelligent Systems (TiiS)*, 9(2-3), 1–26.

[Pedregosa et al., 2011] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., & Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12, 2825–2830.

[Perniss, 2015] Perniss, P. (2015). Collecting and analyzing sign language data: Video requirements and use of annotation software. *Research methods in sign language studies: A practical guide*, (pp. 55–74).

[Povey et al., 2011] Povey, D., Ghoshal, A., Boulianne, G., Burget, L., Glembek, O., Goel, N., Hannemann, M., Motlicek, P., Qian, Y., Schwarz, P., Silovsky, J., Stemmer, G., & Vesely, K. (2011). The Kaldi Speech Recognition Toolkit. In *IEEE 2011 Workshop on Automatic Speech Recognition and Understanding*: IEEE Signal Processing Society.

[Pu et al., 2016] Pu, J., Zhou, W., & Li, H. (2016). Sign language recognition with multi-modal features. In *Pacific Rim Conference on Multimedia* (pp. 252–261).: Springer.

[Pugeault & Bowden, 2011] Pugeault, N. & Bowden, R. (2011). Spelling it out: Real-time asl fingerspelling recognition. In *2011 IEEE International conference on computer vision workshops (ICCV workshops)* (pp. 1114–1119).: IEEE.

[Qiu et al., 2017] Qiu, Z., Yao, T., & Mei, T. (2017). Learning spatio-temporal representation with pseudo-3d residual networks. In *proceedings of the IEEE International Conference on Computer Vision* (pp. 5533–5541).

[Rastgoo et al., 2020] Rastgoo, R., Kiani, K., & Escalera, S. (2020). Hand sign language recognition using multi-view hand skeleton. *Expert Systems with Applications*, 150, 113336.

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

[Russakovsky et al., 2015a] Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A. C., & Fei-Fei, L. (2015a). ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3), 211–252.

[Russakovsky et al., 2015b] Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A. C., & Fei-Fei, L. (2015b). ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3), 211–252.

[Sandler & Lillo-Martin, 2006] Sandler, W. & Lillo-Martin, D. C. (2006). *Sign language and linguistic universals*. Cambridge University Press.

[Sarhan & Frintrop, 2020] Sarhan, N. & Frintrop, S. (2020). Transfer learning for videos: from action recognition to sign language recognition. In *2020 IEEE International Conference on Image Processing (ICIP)* (pp. 1811–1815).: IEEE.

[Shi et al., 2018] Shi, B., Del Rio, A. M., Keane, J., Michaux, J., Brentari, D., Shakhnarovich, G., & Livescu, K. (2018). American sign language fingerspelling recognition in the wild. In *2018 IEEE Spoken Language Technology Workshop (SLT)* (pp. 145–152).: IEEE.

[Skobov & Lepage, 2020] Skobov, V. & Lepage, Y. (2020). Video-to-HamNoSys automated annotation system. In *Proceedings of the LREC2020 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* (pp. 209–216). Marseille, France: European Language Resources Association (ELRA).

[Szegedy et al., 2016] Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., & Wojna, Z. (2016). Rethinking the inception architecture for computer vision.

[Tan & Le, 2019] Tan, M. & Le, Q. V. (2019). Efficientnet: Rethinking model scaling for convolutional neural networks. *arXiv preprint arXiv:1905.11946*.

[Tran et al., 2018a] Tran, D., Wang, H., Torresani, L., Ray, J., LeCun, Y., & Paluri, M. (2018a). A closer look at spatiotemporal convolutions for action recognition. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition* (pp. 6450–6459).

[Tran et al., 2018b] Tran, D., Wang, H., Torresani, L., Ray, J., LeCun, Y., & Paluri, M. (2018b). A closer look at spatiotemporal convolutions for action recognition. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition* (pp. 6450–6459).

[Tsay & Myers, 2009] Tsay, J. & Myers, J. (2009). The morphology and phonology of Taiwan Sign Language. In J. Tai & J. Tsay (Eds.), *Taiwan Sign Language and Beyond* (pp. 83–130). Chia-Yi: The Taiwan Institute for the Humanities.

[Van der Kooij, 2002] Van der Kooij, E. (2002). *Phonological Categories in Sign Language of the Netherlands. The Role of Phonetic Implementation and Iconicity.* Utrecht: LOT.

[van der Walt et al., 2014] van der Walt, S., Schönberger, J. L., Nunez-Iglesias, J., Boulogne, F., Warner, J. D., Yager, N., Gouillart, E., Yu, T., & the scikit-image contributors (2014). scikit-image: image processing in Python. *PeerJ*, 2, e453.

[Vaswani et al., 2017] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). Attention is all you need. *Advances in neural information processing systems*, 30.

[Von Agris & Kraiss, 2007] Von Agris, U. & Kraiss, K.-F. (2007). Towards a video corpus for signer-independent continuous sign language recognition. *Gesture in Human-Computer Interaction and Simulation, Lisbon, Portugal, May*, 11.

[Wei et al., 2016] Wei, S.-E., Ramakrishna, V., Kanade, T., & Sheikh, Y. (2016). Convolutional pose machines. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 4724–4732).

[Woll et al., 2022] Woll, B., Fox, N., & Cormier, K. (2022). Segmentation of signs for research purposes: Comparing humans and machines. In *Proceedings of the LREC2022 10th Workshop on the Representation and Processing of Sign Languages: Multilingual Sign Language Resources* (pp. 198–201).

[World Health Organization, 2021] World Health Organization (2021). *World report on hearing*. World Health Organization.

[Zhang et al., 2023] Zhang, B., Müller, M., & Sennrich, R. (2023). Sltunet: A simple unified model for sign language translation. *arXiv preprint arXiv:2305.01778*.

[Zhang et al., 2019] Zhang, Z., Pu, J., Zhuang, L., Zhou, W., & Li, H. (2019). : (pp. 285–289).: Institute of Electrical and Electronics Engineers (IEEE).

[Zhou et al., 2019] Zhou, H., Zhou, W., & Li, H. (2019). : (pp. 1282–1287).: Institute of Electrical and Electronics Engineers (IEEE).

[Zorzi et al., 2021] Zorzi, G., Aristodemo, V., Cecchetto, C., Giustolisi, B., Hauser, C., Quer, J., Sanchez, J., & Caterina, D. (2021). On the reliability of the notion of native signer and its risks. working paper or preprint.