



UNIVERSITY OF THESSALY

SCHOOL OF ENGINEERING

DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING

Automatic Sign Language Recognition Algorithms and their Incorporation into a Greek Sign Language Educational Platform

A dissertation submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy

Katerina Papadimitriou

December 2023



UNIVERSITY OF THESSALY

SCHOOL OF ENGINEERING

DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING

Automatic Sign Language Recognition Algorithms and their Incorporation into a Greek Sign Language Educational Platform

A dissertation submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy

Katerina Papadimitriou

December 2023



ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΙΑΣ

ΠΟΛΥΤΕΧΝΙΚΗ ΣΧΟΛΗ

ΤΜΗΜΑ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

**Αλγόριθμοι Αυτόματης Αναγνώρισης Νοηματικής Γλώσσας
και Ενσωμάτωσή τους σε Εκπαιδευτική Πλατφόρμα
Ελληνικής Νοηματικής**

Διατριβή η οποία υποβλήθηκε για τη μερική εκπλήρωση
των υποχρεώσεων απόκτησης του Διδακτορικού Διπλώματος

Αικατερίνη Παπαδημητρίου

Δεκέμβριος 2023



UNIVERSITY OF THESSALY

SCHOOL OF ENGINEERING

DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING

Automatic Sign Language Recognition Algorithms and their Incorporation into a Greek Sign Language Educational Platform

PhD Dissertation

Katerina Papadimitriou

Advisory committee

Gerasimos Potamianos, Associate Professor, Department of Electrical and Computer Engineering, University of Thessaly (Supervisor)

Christophoros Nikou, Professor, Department of Computer Science and Engineering, University of Ioannina

Antonios Argyriou, Associate Professor, Department of Electrical and Computer Engineering, University of Thessaly

Examination committee

Gerasimos Potamianos, Associate Professor, Department of Electrical and Computer Engineering, University of Thessaly (Supervisor)

Christophoros Nikou, Professor, Department of Computer Science and Engineering, University of Ioannina

Antonios Argyriou, Associate Professor, Department of Electrical and Computer Engineering, University of Thessaly

Nikolaos Bellas, Professor, Department of Electrical and Computer Engineering, University of Thessaly

Stavroula-Evita Fotinea, Researcher A, Institute for Language and Speech Processing/Athena Research Center

Dimitrios Katsaros, Associate Professor, Department of Electrical and Computer Engineering, University of Thessaly

Dimitrios Rafailidis, Associate Professor, Department of Electrical and Computer Engineering, University of Thessaly

December 2023

DISCLAIMER ON ACADEMIC ETHICS AND INTELLECTUAL PROPERTY RIGHTS

Being fully aware of the implications of copyright laws, I expressly state that this PH.D. dissertation, as well as the electronic files and source codes developed or modified in the course of this thesis, are solely the product of my personal work and do not infringe any rights of intellectual property, personality and personal data of third parties, do not contain work / contributions of third parties for which the permission of the authors / beneficiaries is required and are not a product of partial or complete plagiarism, while the sources used are limited to the bibliographic references only and meet the rules of scientific citing. The points where I have used ideas, text, files and / or sources of other authors are clearly mentioned in the text with the appropriate citation and the relevant complete reference is included in the bibliographic references section. I also declare that the results of the work have not been used to obtain another degree. I fully, individually and personally undertake all legal and administrative consequences that may arise in the event that it is proven, in the course of time, that this thesis or part of it does not belong to me because it is a product of plagiarism.

The declarant

Katerina Papadimitriou

Acknowledgments

This Ph.D. Thesis constitutes the result of extensive personal endeavor, accompanied by continuous support from several individuals whose contribution was instrumental in the completion of this work. First I would like to express my sincere gratitude to my supervisor Gerasimos Potamianos for his trust, significant guidance in my research pursuits, and dedicated encouragement throughout my research enterprise. The expertise he possesses, together with his high expectations and valuable insights, motivated me to improve both my research efficiency and the completion of this Thesis. I would also like to thank Profs. Christophoros Nikou and Antonios Argyriou for their participation in my three-member committee. Christophoros Nikou was my Master Thesis supervisor at the Department of Computer Science and Engineering at University of Ioannina, where his academic knowledge made our collaboration a remarkable experience.

I am also grateful to all partners of the “SL-ReDu” project. Specifically, I would like to thank Eleni Efthimiou, Stavroula-Evita Fotinea, Petros Maragos, and Galini Sapountzaki for their dedication and expertise. Without their assistance and wise guidance, this project would not have been the same. Special thanks also to all contributors to the Greek Sign Language data collection process, especially Kyriaki Vasilaki, Theodoros Goulas, and Panagiotis Karioris.

Moreover, I would like to extend my appreciation to my husband, Alexandros, for his patience and encouragement, as well as my parents and my brother for providing me with persistent support throughout my long years of study. This accomplishment would not have been possible without them.



Finally, it should be noted that this work has been funded by the Hellenic Foundation for Research and Innovation (H.F.R.I.) under the “1st Call for H.F.R.I. Research Projects to support Faculty Members & Researchers and the procurement of high-cost research equipment grant” (Project “SL-ReDu”, Project Number HFRI-FM17-2456).

PhD Dissertation

Automatic Sign Language Recognition Algorithms and their Incorporation into a Greek Sign Language Educational Platform

Katerina Papadimitriou

Abstract

This Ph.D. Thesis focuses on the problem of automatic Sign Language Recognition (SLR) from video data, with a use-case of learning Greek Sign Language (GSL) through a properly designed platform with integrated functionality of automatic GSL recognition. The SLR problem is complex and challenging, due to the multitude of articulators, the complexity of its formation, the variation of the articulation process among signers, the difficulty of robust visual detection and tracking in a variety of non-ideal environments with often low-quality camera sensors, as well as the lack of large-volume SLR databases. At the same time, the process of learning sign language as a “2nd language” requires a very significant effort and student-teacher interaction both during learning as well as objective assessment of students by their teachers, both lacking appropriate supporting tools. The main innovation and contribution of the Thesis lies in the combination of these two problems, i.e., both in the development of innovative and effective algorithms for SLR and their successful integration into a GSL learning platform, demonstrating success both through the evaluation of recognition on various SLR databases, as well as the evaluation of the learning platform objectively and subjectively by a large number of learner users. Specifically, in this Thesis:

- The largest GSL database to date has been collected (at least 3 times longer in duration and number of interpreters than the largest database previously), allowing the training and evaluation of GSL models and their integration into the educational platform, while also made available to the scientific community to further advance the field.
- Numerous algorithms have been investigated for visual detection and tracking of signing articulators and then for extracting visual features from them. Further, it was determined which ones are most suitable for SLR, through experiments on isolated sign recognition. Many of these algorithms have not been used before for this particular problem.

- Innovative algorithms were developed for SLR, with a primary emphasis on sequential learning architectures. These were used to recognize isolated signs, continuous signing, continuous fingerspelling, and finally “cued speech”. These techniques were evaluated on multiple databases, including the aforementioned GSL one. In many cases the algorithms advanced the state-of-the-art.
- Finally, computationally efficient models for GSL recognition (specifically for isolated signing, continuous signing, and continuous fingerspelling) were developed, which were integrated into the GSL educational platform, further aided by appropriate GSL language models. The system was evaluated by a large number of users (GSL students at University of Thessaly and GSL experts), achieving excellent subjective and objective results, the latter being in perfect agreement with the off-line SLR results.

Keywords

Isolated sign language recognition; continuous sign language recognition; continuous fingerspelling recognition; cued speech recognition; visual detection and tracking; feature extraction; sequence learning; Greek sign language recognition

Διδακτορική Διατριβή

Αλγόριθμοι Αυτόματης Αναγνώρισης Νοηματικής Γλώσσας και Ενσωμάτωσή τους σε Εκπαιδευτική Πλατφόρμα Ελληνικής Νοηματικής

Αικατερίνη Παπαδημητρίου

Περίληψη

Η Διατριβή επικεντρώνεται στο πρόβλημα της αυτόματης αναγνώρισης νοηματικής γλώσσας από δεδομένα βίντεο, με πεδίο εφαρμογής την εκμάθηση Ελληνικής Νοηματικής Γλώσσας (ΕΝΓ) μέσω μίας κατάλληλα σχεδιασμένης πλατφόρμας με ενσωματωμένη λειτουργικότητα αυτόματης αναγνώρισης ΕΝΓ. Η αναγνώριση αναγνώρισης νοηματικής αποτελεί ένα πολύπλοκο και δύσκολο πρόβλημα, λόγω της πληθώρας των αρθρωτών νοηματικής, της πολυπλοκότητας του σχηματισμού αυτής, της διαφοροποίησης της διαδικασίας άρθρωσης μεταξύ νοηματιστών, της δυσκολίας ανθεκτικής οπτικής ανίχνευσης και παρακολούθησης σε ποικίλα μη ιδανικά περιβάλλοντα και δεδομένα από κάμερες χαμηλής ποιότητας, όπως και της έλλειψης βάσεων δεδομένων νοηματικής μεγάλου όγκου. Παράλληλα, η διαδικασία της εκμάθησης νοηματικής ως «2ης γλώσσας» απαιτεί πολύ σημαντική προσπάθεια και αλληλεπίδραση μαθητών-εκπαιδευτών τόσο κατά τη διάρκεια της εκμάθησης όσο και για την αντικειμενική αξιολόγηση των μαθητών από τους εκπαιδευτές τους, που στερείται υποστηρικτικών εργαλείων. Η κύρια καινοτομία και συνεισφορά της Διατριβής έγκειται στο συνδυασμό των δύο αυτών προβλημάτων, δηλαδή τόσο στην ανάπτυξη καινοτόμων και αποτελεσματικών αλγορίθμων για την αναγνώριση νοηματικής όσο και στην επιτυχή ενσωμάτωση αυτών σε πλατφόρμα εκμάθησης ΕΝΓ, επιδεικνύοντας την επιτυχία του εγχειρήματος τόσο μέσω αξιολόγησης της αναγνώρισης σε κλασσικές βάσεις δεδομένων νοηματικής, όσο και της αξιολόγησης της πλατφόρμας εκμάθησης αντικειμενικά και υποκειμενικά από μεγάλο αριθμό χρηστών αυτής. Συγκεκριμένα, στη Διατριβή αυτή:

- Συλλέχθηκε η μεγαλύτερη μέχρι σήμερα βάση δεδομένων ΕΝΓ (τουλάχιστον 3 φορές μεγαλύτερη σε διάρκεια και αριθμό νοηματιστών από την μέχρι πρότινος μεγαλύτερη βάση), η οποία επέτρεψε την εκπαίδευση και αποτίμηση μοντέλων ΕΝΓ για τη Δια-

τριβή και την ενσωμάτωσή τους στην εκπαιδευτική πλατφόρμα, ενώ έγινε διαθέσιμη στην επιστημονική κοινότητα για περαιτέρω πρόοδο στο πεδίο.

- Ερευνήθηκε πληθώρα αλγορίθμων για οπτική ανίχνευση και παρακολούθηση των αρθρωτών της εκφοράς νοηματισμού και στη συνέχεια για την εξαγωγή οπτικών χαρακτηριστικών από αυτούς. Περαιτέρω, διαπιστώθηκε ποιες είναι οι πιο κατάλληλες για το πρόβλημα της αναγνώρισης νοηματικής, μέσω πειραμάτων για μεμονωμένο νοηματισμό. Πολλοί από τους αλγορίθμους αυτούς δεν είχαν χρησιμοποιηθεί στο παρελθόν για το συγκεκριμένο πρόβλημα.
- Αναπτύχθηκαν καινοτόμοι αλγόριθμοι για την αναγνώριση νοηματικής γλώσσας, με κύρια έμφαση στις αρχιτεκτονικές ακολουθιακής μάθησης. Οι αλγόριθμοι αυτοί χρησιμοποιήθηκαν για αναγνώριση μεμονωμένου νοηματισμού, συνεχούς νοηματισμού, συνεχούς δακτυλοσυλλαβισμού, και τέλος «συνεπαγόμενης ομιλίας» (“cued speech”). Η αξιολόγηση των τεχνικών αυτών έγινε σε πολλαπλές βάσεις δεδομένων, συμπεριλαμβανομένης της ENΓ. Στις πολλές περιπτώσεις οι αλγόριθμοι προήγαγαν τεχνολογία αιχμής.
- Τέλος, αναπτύχθηκαν υπολογιστικά αποτελεσματικά μοντέλα για αναγνώριση ENΓ (συγκεκριμένα για μεμονωμένο νοηματισμό, συνεχή νοηματισμό, και συνεχή δακτυλοσυλλαβισμό), τα οποία ενσωματώθηκαν στην εκπαιδευτική πλατφόρμα εκμάθησης ENΓ, με περαιτέρω υποβοήθηση από κατάλληλα γλωσσικά μοντέλα ENΓ. Το σύστημα αξιολογήθηκε από μεγάλο αριθμό χρηστών (φοιτητών του Πανεπιστημίου Θεσσαλίας και εμπειρογνομόνων ENΓ), επιτυγχάνοντας εξαιρετικά υποκειμενικά και αντικειμενικά αποτελέσματα, με τα τελευταία να βρίσκονται σε απόλυτη συμφωνία με τα πειραματικά αποτελέσματα των αναπτυγμένων αλγορίθμων αναγνώρισης.

Λέξεις Κλειδιά

Αναγνώριση μεμονωμένων νοημάτων, αναγνώριση συνεχούς νοηματισμού, αναγνώριση συνεχούς δακτυλοσυλλαβισμού, αναγνώριση συνεπαγόμενης ομιλίας, οπτική ανίχνευση και παρακολούθηση, εξαγωγή χαρακτηριστικών, ακολουθιακή μάθηση, αναγνώριση Ελληνικής νοηματικής γλώσσας

Table of Contents

Abstract	xiii
Abstract in Greek	xv
Table of Contents	xvii
List of Figures	xxi
List of Tables	xxv
Abbreviations	xxvii
1 Introduction	1
1.1 Sign Language Recognition Pillars	4
1.2 Thesis Contribution	6
1.3 Thesis Organization	9
2 Sign Language Recognition Datasets	11
2.1 Introduction	11
2.2 Harvesting Pre-Existing SLR Corpora	13
2.3 The Collected SLR Resources	19
2.3.1 SL-ReDu Corpus Content	20
2.3.2 Studio Data Collection	20
2.3.3 Non-Studio Data Collection	25
2.4 Conclusions	27
3 Detection and Tracking of Visual SL Articulators	29
3.1 Introduction	29

3.2	Hand Detection and Type Classification	30
3.3	Pose-Based Articulation Detection	32
3.3.1	2D Skeleton Regression	32
3.3.2	3D Skeleton Regression	34
3.3.3	Expressive 3D Full Body Pose and Shape via 3D Reconstruction . .	34
3.4	Data and Experimental Framework	36
3.5	Experimental Results	37
3.6	Conclusions	40
4	Extraction of Visual Features of Tracked Articulators	43
4.1	Introduction	43
4.2	Extraction of Visual Features	44
4.2.1	Pose Features in 2D and 3D	45
4.2.2	Expressive Body Pose and Shape Representation in 3D	46
4.2.3	Appearance Features	46
4.2.4	Optical Flow Features	49
4.3	Data and Experimental Framework	50
4.4	Experimental Results	52
4.5	Conclusions	54
5	Sign Language Recognition Architectures	57
5.1	Introduction	57
5.2	Single-Stream SLR	58
5.2.1	Single-Stream Sequence Learning Models	58
5.2.2	Single-Stream Sequence Learning Models with Attention	60
5.2.3	Inference via Beam Search	65
5.3	Multi-Stream SLR Fusing Multiple Modalities	66
5.4	Data and Experimental Framework	67
5.5	Experimental Results	70
5.6	Conclusions	72
6	Bimodal System for Isolated SLR	75
6.1	Introduction	75
6.2	The Isolated SLR System	77

6.2.1	Appearance-Based Modality	77
6.2.2	Skeleton-Based Modality	79
6.2.3	Multi-Modal Fusion	82
6.3	Experimental Framework	83
6.4	Experimental Results	84
6.5	Conclusions	88
7	Bimodal System for Continuous SLR	89
7.1	Introduction	89
7.2	The Continuous SLR system	91
7.2.1	Visual Module	92
7.2.2	Sequence Learning	93
7.2.3	Alignment Module	94
7.2.4	Ensemble Module	95
7.3	Experimental Framework	96
7.4	Experimental Results	97
7.5	Conclusions	100
8	Continuous Fingerspelling Recognition	103
8.1	Introduction	103
8.2	The Continuous Fingerspelling Recognition System	105
8.2.1	Visual Module	105
8.2.2	Sequence Learning	107
8.3	Experimental Framework	107
8.4	Experimental Results	108
8.5	Conclusions	111
9	Cued Speech Recognition	113
9.1	Introduction	113
9.2	The Cued Speech Recognition System	115
9.2.1	Visual Module	116
9.2.2	Sequence Learning	119
9.3	Experimental Framework	120
9.4	Experimental Results	121

9.5	Conclusions	124
10	Real-Time Sign Language Recognition for a GSL Education Platform	127
10.1	Introduction	127
10.2	The SL-ReDu Prototype System	129
10.3	Overview of the GSL Recognition Approach	131
10.3.1	Visual Pre-Processing	133
10.3.2	Recognition of Isolated GSL Signs	134
10.3.3	Recognition of Continuous GSL Signing	136
10.3.4	Recognition of Continuous GSL Fingerspelling	139
10.3.5	Summary of GSL Recognizer Performance	142
10.4	GSL Education Platform Evaluation	142
10.4.1	Objective Assessment of GSL Production	143
10.4.2	Subjective Assessment of the GSL Education Platform	145
10.5	Conclusions	147
11	Conclusions and Future Directions	149
11.1	Conclusions	149
11.2	Future Directions	152
	References	153
	List of Publications Related to the Thesis	177

List of Figures

1.1	Upper-body articulation cues that contribute to the formation of basic signs	2
1.2	Most frequent handshape configurations in GSL	3
1.3	Complexity dimensions of the SLR problem	3
1.4	Articulation variability among signers	4
1.5	Sample frame sequences corresponding to different types of signing	5
2.1	Sample frames from the <i>Dicta-Sign database</i>	14
2.2	Example frames from the <i>Polytropon GSL corpus</i>	15
2.3	Sample frames from the <i>ITI Greek Sign Language Dataset</i>	16
2.4	Sample frames from the <i>AUTSL dataset</i>	16
2.5	Sample frames from the German CSLR datasets	17
2.6	Example frames of the <i>Chicago Fingerspelling in the Wild</i>	18
2.7	Example video frames from the cued speech datasets	18
2.8	Distribution of the 369 GSL signs of the SL-ReDu isolated sign corpus . . .	19
2.9	Snapshots of the monitor, showing messages and elicitation material	22
2.10	SL-ReDu corpus example frames from the RGB HDV camera and the depth Intel RealSense camera	23
2.11	Example frames from the continuous GSL studio data collected	24
2.12	Example frames of non-studio recordings of isolated signs	26
2.13	Example frames from the continuous GSL non-studio video data	27
3.1	Block diagram of the hybrid HDTC system	30
3.2	HDTC algorithm evaluation on sample SL video frames	31
3.3	2D and 3D human skeleton regression	33
3.4	ExPose regression model pipeline	35
3.5	FrankMocap regression model architecture	36

3.6	Accuracy comparison of the HDTC scheme against all 2D and 3D human skeleton regression frameworks	37
3.7	Confusion matrices of (a) the HDTC algorithm; (b) OpenPose; and (c) MediaPipe on all three evaluation sets	39
3.8	Qualitative evaluation of the 3D full body pose and shape reconstruction methods	40
4.1	General overview of the various visual feature stream based models	44
4.2	Optical flow motion informative images obtained by the various optical flow models	49
5.1	Algorithmic flow-chart of SLR relying on single-stream sequence learning models with CTC-based decoding	59
5.2	Attention-based LSTM encoder-decoder architecture	61
5.3	Attention-based BiLSTM encoder-decoder architecture	62
5.4	Attention-based GRU encoder-decoder architecture	63
5.5	Attention-based CNN encoder-decoder	64
5.6	Multi-modal SLR via early fusion	67
5.7	Multi-modal SLR via late fusion	68
6.1	Proposed ISLR model architecture	76
6.2	Deformable convolution	78
6.3	The SiLU and ReLU activation functions	79
6.4	3D body reconstruction on sample frames of the evaluated datasets	80
6.5	Self-attention module integrated into the ST-GCN unit	82
6.6	Examples of signs of the SL-ReDu isolated corpus that are frequently confused by our model due to their similar handshapes and hand orientation	87
7.1	Proposed bimodal CSLR model	91
7.2	Window-based RNN module	93
7.3	Self-attention enhancement with local modeling via Gaussian bias and relative position encoding	94
7.4	Model predictions generated by the proposed CSLR model, as well as several variations of it against the reference (ground truth)	99

8.1	Architecture of the introduced fingerspelling recognition model	104
8.2	Model predictions generated by the proposed fingerspelling recognition model, as well as several variations of it against the reference (ground truth)	110
9.1	French cued speech phonetic encoding systems	114
9.2	Architecture of the introduced cued speech recognition system	116
9.3	Examples of articulator detection and keypoint feature extraction	117
9.4	Model architecture for the generation of 3D hand skeleton from correspond- ing 2D information	118
9.5	Location-based area division of possible signing hand positions relative to the mouth	119
9.6	Comparative evaluation of various sequence learning models on both datasets in terms of phoneme error rate (PER, %) using all feature streams fused syn- chronously or asynchronously	123
10.1	Illustration of the SL-ReDu prototype system design	128
10.2	The adopted SL-ReDu communication between the web application, recog- nition device, and intermediate server	130
10.3	Algorithmic flow-chart of the GSL recognition module of the SL-ReDu plat- form	132
10.4	Examples of (a) 3D body, (b) left hand, and (c) right hand skeleton represen- tations, produced by the MediaPipe holistic pipeline	133
10.5	Isolated GSL recognition performance (in WAcc, %) with the integration of the class-based LM, as well as without it, under both MS and SI experimental frameworks	136
10.6	Continuous GSL recognition performance (reported in terms of sentence ac- curacy (SAcc, %)) when no LM is used, as well with the integration of the class-based LM or the 3-gram LM, under both MS and SI experimental frame- works	139
10.7	GSL recognition performance for continuous fingerspelling (reported in terms of word and letter accuracies, %) when no LM is used, as well as with the integration of the RNN LM or the 5-gram LM, under both MS and SI exper- imental frameworks	142

10.8 Summary of the subjective responses to 8 questions on the Likert scale over all 44 evaluation participants	146
--	-----

List of Tables

2.1	Summary of large-scale SLR datasets	12
2.2	Core phrase types in the SL-ReDu GSL corpus	21
4.1	Sign classification accuracy (%) of isolated GSL recognition on the Polytropon and ITI GSL corpora, employing various feature representations	52
4.2	Sign classification accuracy (%) of isolated GSL recognition on the Polytropon corpus, when combining various well-performing visual features by late fusion	53
4.3	Sign classification accuracy (%) of isolated GSL recognition on the ITI GSL corpus (under both experimental settings), when combining various well-performing visual features by late fusion	54
5.1	Word error rate (%) of continuous GSL recognition on the multi-signer ITI continuous GSL corpus, using various visual feature streams independently in conjunction with the single-stream sequence learning models	71
5.2	Word error rate (%) of continuous GSL recognition on the multi-signer ITI continuous GSL corpus, using various feature stream combinations under early or late fusion in conjunction with the BiLSTM model	72
6.1	Sign recognition accuracy (%) on all three datasets using various 3D-CNNs	84
6.2	Sign recognition accuracy (%) on all three datasets using ST-GCN variants	85
6.3	Sign recognition accuracy (%) on all three datasets using various streams for the skeleton graph construction	85
6.4	Ours vs. literature results on both datasets	86
7.1	Ablation study for the RNN module (RNN), the Relative Position (RP) encoding, and the Gaussian Bias (GB) of single and both modalities	96

7.2	Ablation study concerning the various loss functions	98
7.3	Evaluation comparison on the RWTH-2014 dataset in terms of WER (%) .	100
7.4	Evaluation comparison on the RWTH-2014T dataset in terms of WER (%) .	100
8.1	Ablation study concerning the auxiliary loss functions and the video feature learner	109
8.2	Letter accuracy (LAcc, %) comparison of state-of-the-art on the ChicagoF-SWild dataset under SI setting	111
9.1	Phoneme error rate (PER, %) on the French and British English cued speech datasets, employing various feature stream combinations	122
9.2	Evaluation comparison on both datasets in terms of phoneme error rate (PER, %)	124
10.1	Summary of GSL recognition results of the previous experimental sections for the three tasks considered under both the MS and SI experimental frameworks	143
10.2	Summary over all users of the results of GSL production exercises in the system evaluation	144
10.3	Comparison of objective evaluation recognition results for each GSL recognition task	145
10.4	Median, maximum, and minimum values of the subjective evaluation of the platform for each of the eight questions over all 44 evaluation participants .	146

Abbreviations

ASL	American Sign Language
BiLSTM	Bidirectional Long Short-Term Memory
CGSL	Continuous Greek Sign Language
CNN	Convolutional Neural Network
CSLR	Continuous Sign Language Recognition
CTC	Connectionist Temporal Classification
dim	dimensional
FC	Fully-Connected
FGSL	Fingerspelling Greek Sign Language
GB	Gaussian Bias
GCN	Graph Convolutional Network
GLU	Gated Linear Unit
GRU	Gated Recurrent Unit
GSL	Greek Sign Language
HCI	Human-Computer Interaction
HDTC	Hand Detection and Type Classification
HMM	Hidden Markov Model
HOG	Histogram of Oriented Gradients
IGSL	Isolated Greek Sign Language
IoU	Intersection-over-Union
ISLR	Isolated Sign Language Recognition
L2	Second Language
LAcc	Letter Accuracy
LBP	Local Binary Patterns
LM	Language Model

LSTM	Long Short-Term Memory
MS	Multi-Signer
PCA	Principal Component Analysis
PER	Phoneme Error Rate
R-CNN	Region-based Convolutional Neural Network
ReLU	Rectified Linear Unit
RNN	Recurrent Neural Network
ROI	Region-Of-Interest
RP	Relative Position
SAcc	Sentence Accuracy
SED	Special Education Department
SGDM	Stochastic Gradient Descent with Momentum
SI	Signer-Independent
SIFT	Scale-Invariant Feature Transform
SiLU	Sigmoid Linear Unit
SL	Sign Language
SLR	Sign Language Recognition
SL-ReDu	Sign Language Recognition in Education
ST-GCN	Spatio-Temporal Graph Convolutional Network
TCN	Temporal Convolution Network
TDS	Time-Depth Separable
UTH	University of Thessaly
WAcc	Word Accuracy
WER	Word Error Rate

Chapter 1

Introduction

Sign Language (SL) constitutes a non-verbal means of communication, occurring in the 3D visible space around the signer's upper torso, encapsulating numerous upper-body articulation cues to facilitate communication for the deaf and hard of hearing. Specifically, it is a natural language, which relies on the signer's manual articulation, i.e. the shape, motion pattern, and relative position of one or both hands, as well as non-manual articulators, in particular body leaning, shoulder motion, head pose, mouthing patterns, and eyebrow movement, all contributing to the formation of basic signs [1–3] (see also Figure 1.1). SLs are full-fledged language systems, comprising their own distinct lexicon and grammar/syntax rules, with the latter being entirely different from spoken languages. Such languages vary across the globe, where countries and regions have independent SLs associated with their own linguistic structures, reflecting the diverse cultural and linguistic characteristics of the corresponding deaf communities. According to the World Federation of the Deaf, there are over 300 SLs around the world [4]. One such example is the Greek Sign Language (GSL), an evolving and adaptive language, where new signs with specific morphological and articulatory features are constantly being incorporated to meet the communication needs of the Greek deaf community (see also Figure 1.2).

In the meanwhile, the need for communication and education in SL as native language and as second language (L2) has dramatically increased, well beyond the approximately 1% of the deaf population [6], as a result of European and national policies on inclusion and accessibility, as well as the official recognition of national SLs. However, non-native SL education is still a time-consuming process that necessitates intensive and iterative one-to-one feedback between instructors and students, suffering from a high level of instructor subjectivity in evaluating students' proficiency [7, 8].

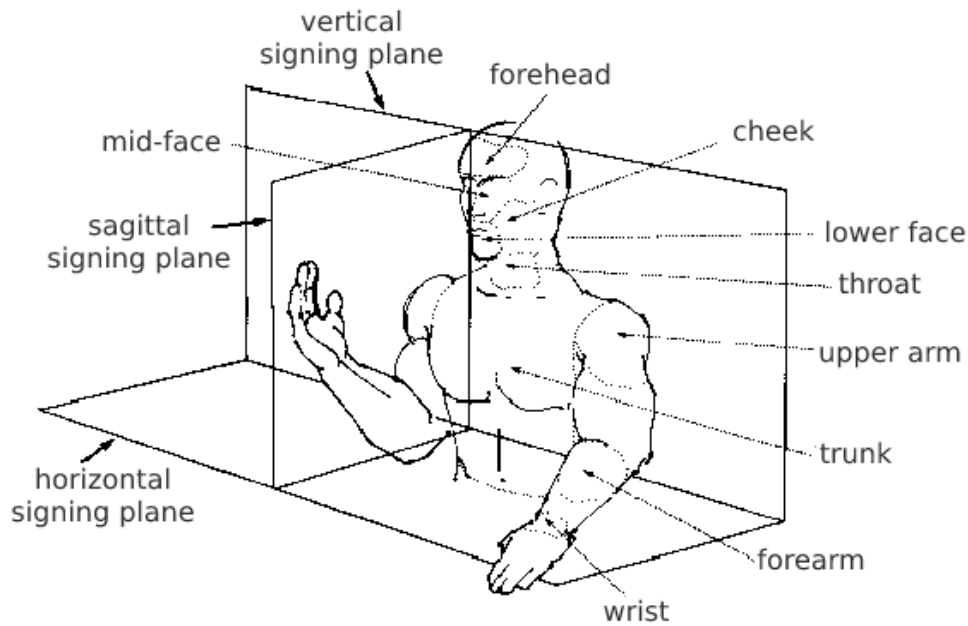


Figure 1.1: Upper-body articulation cues that contribute to the formation of basic signs (figure adapted from [5]).

To that end, there has been substantial research activity in the development of technologies addressing Sign Language Recognition (SLR) from videos, which have the potential of significantly bridging the communication barrier between deaf-deaf and deaf-hearing individuals, being particularly relevant in the areas of SL teaching, drilling, and assessment of non-native users. Despite much progress in the domain of SLR due to recent deep learning advances in the fields of computer vision and human language technologies, as well as the collection of suitable large SL corpora [9–11], the problem remains challenging due to the multitude, complexity, and strong spatio-temporal correlation of the SL articulators participating in signing, as well as the intricacies of the visual environment (see also Figure 1.3). Such issues are substantially more challenging in the instance of SLR under a signer-independent (SI) setting, due to the inherent articulation variability among signers (see also Figure 1.4).

To address such issues, a significant portion of SLR approaches in the literature rely on input from wearable markers or hand glove motion capturing devices [17–19], facilitating hand tracking and, by extension, sign capturing, however at the expense of naturalness in Human-Computer Interaction (HCI) [20,21]. In the last few years, significant attention has been given to depth (RGB-D) cameras [22–25], such as the Kinect [26] and Intel RealSense [27], assisting in body segmentation. The majority of systems, though, consider traditional RGB video input, with many incorporating motion-based algorithms [28] to exploit the dynamic nature of

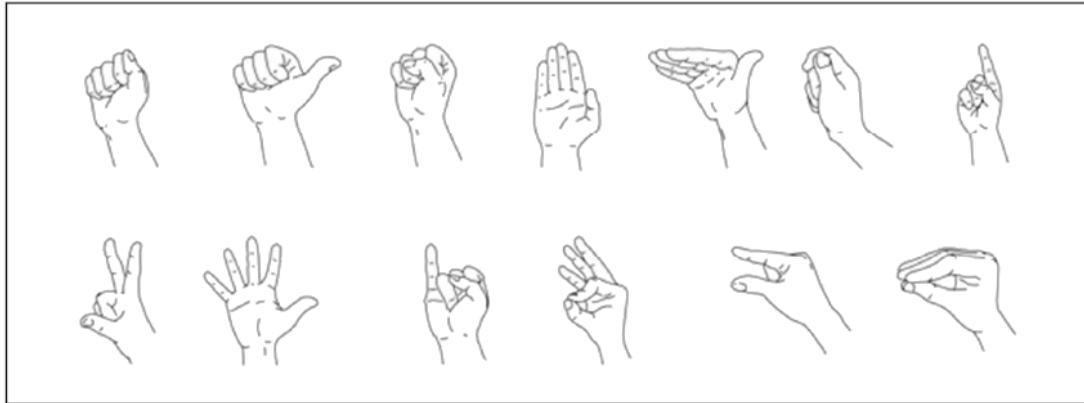


Figure 1.2: Most frequent handshape configurations of the Greek sign language isolated signs (figure adapted from [12]).

gestures enhancing robustness to challenging lighting conditions and cluttered backgrounds.

Among the SLR tasks in the literature, most works have focused on Isolated Sign Language Recognition (ISLR) [30–34], as well as the more challenging instance of continuous recognition of SL gloss sequences (Continuous Sign Language Recognition - CSLR) [10, 13, 35–39], where the goal is to predict a sequence of glosses from the SL video. However, a small number of studies have focused on the issue of continuous fingerspelling recognition, which is important for sign-based communication since it is frequently used for well-known words without distinctive signs, including names, technical terms, or foreign words [16, 40–44]. Furthermore, some works address the recognition of continuous cued speech [45–47], which

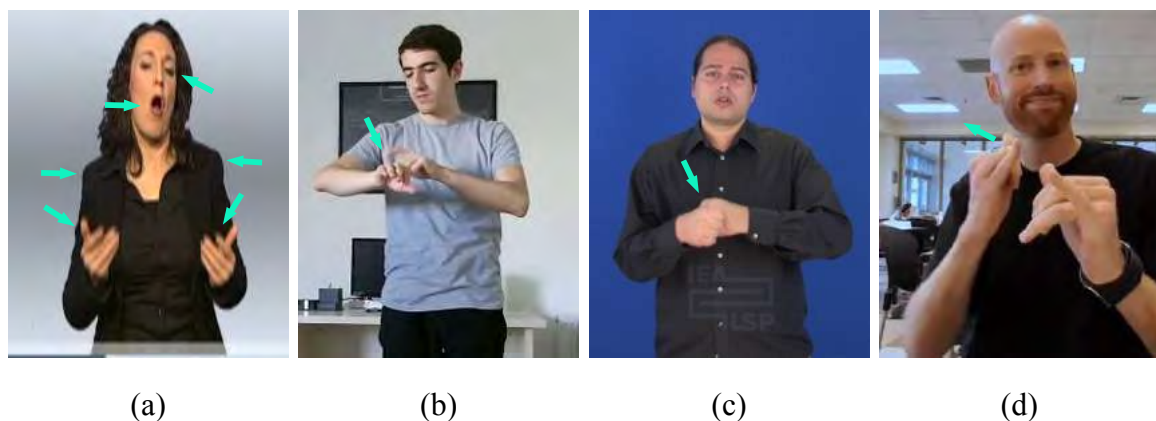


Figure 1.3: Complexity dimensions of the SLR problem: Sample frames from (a) the *RWTH-PHOENIX Weather 2014 multi-signer corpus* [13], (b) the *AUTSL Turkish SL dataset* [14], (c) the *Polytropon corpus* [15], and (d) the *ChicagoFSWild dataset* [16], showing the multitude of SL articulators (hands, eyebrows, mouth, and shoulder leaning), handshape complexity, and occlusion of the SL articulation, as well as the variety of visual environments.



Figure 1.4: Articulation variability among two different signers regarding the fingerspelling signing of the Greek alphabet letter “M” (1st row) and the isolated signing of number “7” (2nd row). Sample frames are from the *SL-ReDu corpus* [29].

is a non-vocal communication system that uses lip movements in conjunction with hand positional and gestural cues to disambiguate phonetic information and make it accessible to the speech and hearing impaired. Speechreading is crucial to speech perception for the hearing impaired, even though it can be inaccurate due to the confusability of visual speech patterns. Note that recognizing continuous signed phrases, fingerspelled words, and continuous cued speech are more intricate tasks compared to the ISLR problem (see also Figure 1.5), since there is no prior knowledge of gloss, letter, or phoneme-level segmentation, respectively.

1.1 Sign Language Recognition Pillars

Critical components in addressing SLR are the accurate visual detection and tracking in space and time, the extraction of visual feature representation in the respective articulatory streams, and subsequently the employment of suitable learning approaches for sign prediction. Reaching these objectives calls for an interdisciplinary approach that makes use of cutting-edge methods from the fields of sign linguistics, computer vision, and machine learning, as well as large amounts of SL data.



Figure 1.5: Sample frame sequences corresponding to isolated signing (1st row); continuous signing (2nd row); fingerspelling signing (3rd row); and cued speech (4th row). Sample frames are from the *SL-ReDu corpus* [29] (rows 1-3) and the *French cued speech dataset* [48].

Detection and tracking of the various SL articulators constitutes an important aspect in SLR. Due to the inherent advantages of the depth modality information and the availability of the human skeleton data stream, a significant portion of work in the field is sensor-based, using specialized data acquisition tools, such as depth cameras [49] or motion capturing systems [17, 50]. The problem of vision-based hand detection has attracted great interest [45, 51], while substantial attention has been dedicated to the field of face detection and recognition [52]. In particular, there is a large portion of such algorithms relying on traditional RGB video input, leveraging for example skin color consistency to apply skin-based color models for manual and non-manual SL articulation detection [53], while others exploit the dynamic nature of gesturing incorporating motion-based algorithms for tracking [54, 55]. Further, a number of promising works [37, 56–58] rely on 2D, as well as 3D hand and body skeletal data, employing state-of-the-art techniques in computer vision.

Apart from tracking, another important aspect of SLR concerns the type of visual features that derive from the detected articulation, capturing SL information at a local or global level. Such could be RGB or RGB-D appearance representations, as well as optical flow motion representations of articulation. For modeling such representations, early works rely on handcrafted features, such as Principal Component Analysis (PCA) representations of the regions-of-interest, as well as more advanced sets, such as Gabor filterbank energies, Scale-Invariant Feature Transform (SIFT), Local Binary Patterns (LBP), and Histogram of

Oriented Gradients (HOG) [32, 59, 60]. Lately, deep learning approaches have provided a breakthrough in the field of feature learning, primarily in the form of 2D Convolutional Neural Networks (CNNs) [33, 35, 36, 38, 57, 61–66], that demonstrate superior performance. Among the proposed architectures, 3D-CNNs [67–71] that capture the spatio-temporal SL dependencies, as well as deep auto-encoders have gained substantial popularity. Recently, there has been a tendency towards pose-related features in the form of 2D and 3D human skeletal landmarks [33, 35]. In addition, there are several systems relying on the combination of articulation position and/or movement estimation, as well as appearance-based descriptors [35, 56, 57]. Inspired by the recent advances in whole-body reconstruction, we have been among the first that integrated 3D expressive body pose and shape representations into the SLR system.

Another important axis of SLR research lies on the sequence modeling approach for gloss prediction. Many researchers have developed automated speech recognition systems, treating signs similarly to phonemes or words, and considering SLR as a linguistic task. Early systems concerning sequence modeling have considered Hidden Markov Models (HMMs) as a possible solution to the problem [30, 72, 73], including parallel HMMs [74], as well as segmental conditional random fields [40, 75]. Most contemporary SLR schemes, though, employ Recurrent Neural Networks (RNNs) [35, 38, 62, 68, 76], typically the Long Short-Term Memory (LSTM) [77] or Bidirectional LSTM (BiLSTM) [78], as well as the Transformers [79], such as the works in [66, 80], in conjunction with Connectionist Temporal Classification (CTC) [81]. Furthermore, a number of introduced systems make use of attention-based encoder-decoders [36, 68, 71], treating SLR as a neural machine translation task. Finally, Graph Convolutional Networks (GCNs) [56, 82–84] have been integrated into some schemes in the literature because of their adaptability to signing dynamics [37, 58].

1.2 Thesis Contribution

The main innovation and contribution of this Dissertation lies in the development of novel and effective algorithms for SLR from RGB video data and, subsequently, their successful incorporation into a GSL learning platform, demonstrating success both through the evaluation of recognition on various SLR databases, as well as the evaluation of the learning platform objectively and subjectively by a large number of learner users. Specifically, in this Thesis:

- Since the content required for the education use-case is not sufficiently covered in existing available GSL databases, we have collected the largest GSL set, the so-called SL-ReDu GSL corpus. In particular, we strived to improve the field of GSL recognition, alignment, and translation by providing a database 3 times longer in duration and number of interpreters than the previously largest GSL database. This resulted in the SL-ReDu GSL corpus, an extensive RGB-D video collection, containing three distinct video subsets: (i) isolated signs; (ii) continuous signing; and (iii) fingerspelling, recorded under studio conditions suitable for GSL recognition. This database allowed the training and evaluation of GSL models and their integration into the educational platform, while also made available to the scientific community to further advance the field.
- We have investigated numerous algorithms for visual detection and tracking of signing articulators and then for extracting visual feature representations from the respective articulatory streams. In particular, we have considered various networks for visual tracking and feature extraction that rely on human pose and 3D shape regression, as well as appearance and motion representation extraction. Further, we have determined which ones are most suitable for SLR, through experiments on two isolated GSL recognition datasets. In addition, we have explored a number of sequence learning architectures processing the feature representation streams, as well as their fusion at the appropriate temporal level, for sign prediction. Subsequently, we have evaluated these approaches on a continuous GSL database. Many of these methods have not been used before for this particular problem.
- We have addressed isolated SLR from RGB video, introducing an innovative deep-learning framework that leverages multi-modal appearance- and skeleton-based information. Specifically, we have introduced three components for the first time in SLR: (i) a modified version of the ResNet2+1D network [85] to capture signing appearance information, where spatial and temporal convolutions are substituted by their deformable counterparts, accomplishing both prevalent spatial modeling potential and motion-aware modeling adaptability; (ii) a novel spatio-temporal graph convolutional network (ST-GCN) [82] that integrates a GCN variant [86], involving weight and affinity modulation for modeling diverse correlations between different body joints beyond the physical human skeleton structure; and (iii) the PIXIE 3D human pose and shape

regression scheme [87] to generate 3D joint-rotation parameterization used for graph construction. Both appearance- and skeleton-based streams have been ensembled in the proposed system and evaluated on three datasets of isolated signs, two in Greek and one in Turkish.

- In addition, we have introduced a novel Transformer-based approach for CSLR from videos, aiming to address the shortcomings of traditional Transformers [79] in learning local semantic context of SL. Specifically, our proposed approach has relied on two distinct components: (a) a window-based RNN module [88] to capture local temporal context and (b) a Transformer encoder, enhanced with local modeling, through Gaussian bias [89] and relative position information [90], as well as global structure modeling via multi-head attention. To further improve model performance, we have designed a bimodal framework that applies the proposed to both appearance and motion signing streams, aligning their posteriors through a guiding CTC technique [91]. Further, we have achieved visual feature and gloss sequence alignment by incorporating a knowledge distillation loss [92]. Experimental evaluation on two popular German CSLR datasets, as well as one in GSL, has demonstrated the superiority of our model.
- We have also tackled the challenging task of continuous fingerspelling recognition from videos, introducing an end-to-end deep learning approach [36], relying on spatio-temporal feature representations derived from a 3D-CNN image feature learner, coupled with a BiLSTM encoder and a CTC alignment module for letter prediction. Specifically, we have examined the contribution of the ResNet2+1D network [85] in modeling the visual representations of the video frames, being the first use of a 3D-CNN visual feature learner in this task, and we have integrated an alignment module that combines the CTC loss with a knowledge distillation loss [92] to align the probability distributions generated by the sequence learning model and the 3D-CNN based visual module. Further, we have incorporated an auxiliary loss function focusing on the visual module to enhance the representation capacity of the 3D-CNN visual feature learner. We have conducted experiments on two fingerspelling datasets, one American and one Greek, and compared our approach to existing state-of-the-art SLR methods, demonstrating its superiority.

- We have also addressed the task of automatic recognition of cued speech from upper-body videos, developing a deep learning system [47] relying on visual features incorporating hand position embeddings, 2D-CNN and 3D-CNN based appearance features of the lips and signing hand regions, as well as skeletal information of the lips and hand articulators. We have fused these features via simple concatenation and propagated them to a sequence learning model consisting of a time-depth separable (TDS) convolution block structure based encoding module [93, 94], followed by a fully-connected layer, as well as a CTC training loss and a corresponding decoding technique [81]. We have investigated the contribution of the above to cued speech recognition, evaluating our model on a French and a British English cued speech video dataset, reporting significant gains over the state-of-the-art on both sets.
- To radically improve accessibility of interaction via SL and to address the need for standardized teaching of SL as L2, we have built an education learning and assessment system [34], as part of the SL-ReDu project, supporting both self-monitoring and objective evaluation of receptive and productive GSL learning. We have also developed the GSL recognizer, integrating it into the SL-ReDu prototype system, in conjunction with appropriate GSL language models. Specifically, the GSL recognition module has been designed to be efficient / “light-weight” enough to allow such recognition to occur with small latency, accommodating three recognition tasks, namely: (i) isolated signs within a large-scale vocabulary (a vocabulary size of 337 GSL signs); (ii) continuous signing (within a set of 695 unique GSL phrases); and (iii) continuously fingerspelled letter sequences, with a vocabulary size of 24 Greek letters. The system has been evaluated by a large number of users (GSL students at University of Thessaly and GSL experts), achieving excellent subjective and objective results, the latter being in perfect agreement with the off-line recognition results.

1.3 Thesis Organization

The rest of Thesis is organized as follows:

- In Chapter 2, we first discuss some already existing datasets of SLR that are freely available in the literature and we harvest to advance our work. Subsequently, we present

our GSL data collection, as well as the corpus content, the studio setup, and the recording protocol for data acquisition.

- In Chapter 3, we examine the detection and tracking of the various SL articulators of interest employing traditional detection and tracking techniques, as well as deep learning-based approaches.
- In Chapter 4, we explore the visual feature extraction of the detected articulators, traditional shape, appearance, and motion-based feature representations, as well as more advanced deep-learning representations.
- In Chapter 5, we build upon the visual tracking and feature extraction methods that we developed in the two previous chapters, investigating single- and multi-stream sequence learning schemes that we adapt to the problem of SLR.
- In Chapter 6, we present the system we have developed to solve the problem of isolated SLR, followed by our experiments.
- In Chapter 7, we focus on the continuous SLR task, presenting our bimodal framework and our experimental results.
- In Chapter 8, we address continuous fingerspelling recognition, detailing our developed algorithms and presenting our experimental results.
- In Chapter 9 we focus on the cued speech recognition task, providing detailing of our developed modeling approach and presenting our experimental results.
- In Chapter 10 we present our GSL recognition models, which are integrated into the SL-ReDu prototype system, accompanying each model presentation with experimental results on SL-ReDu data. Further, we report the findings of the evaluation campaign for the prototype system, performed by both students and experts involved in GSL teaching and learning at University of Thessaly.
- Finally, in Chapter 11, we conclude this Thesis, summarizing our contributions and providing indicative future research directions.

Chapter 2

Sign Language Recognition Datasets

2.1 Introduction

Although SL data collection and annotation constitutes a more “expensive” process, as compared to speech recognition corpora, a considerable amount of annotated SLR video datasets has been released. As also depicted in Table 2.1, there are various available corpora in the literature, containing isolated signing, continuous signing, and continuous fingerspelling of letter sequences.

As already mentioned in Section 1, the main contribution of this Thesis constitutes the development of novel SLR deep-learning algorithms to be integrated into an education tool, in a focused use-case of supporting standardized teaching of GSL as second language by automating the process of student self-monitoring and evaluation. Achieving these goals requires the availability of suitable GSL data resources relevant to the SL-ReDu education use-case. To this end, three GSL corpora are deemed useful to our work, allowing research on GSL automatic recognition algorithms, such as the *Dicta-Sign database* [95], the *Polytropon parallel corpus* [15], and the *ITI GSL corpus* [96]. However, since these databases do not focus on the education use-case, the collection of additional data is deemed necessary for the development of the system prototype. Further, such data with enriched signer, environment, and lexical variability are critical to data-intensive deep-learning algorithms. For this reason, we collected the SL-ReDu GSL corpus [29] to improve the field of GSL recognition, alignment, and translation by providing a sizable signer population and extended lexical coverage. As it can be observed in Table 2.1, the resulting dataset constitutes the largest set of GSL data appropriate for SLR, in terms of number of signers (21), duration (36 hrs), and content.

Table 2.1: Summary of large-scale SLR datasets of isolated signing (Iso.), continuous signing (Con.), and fingerspelling (Fing.), including the SL-ReDu GSL dataset. Duration is in hrs:min.

Dataset	Type	Language	Signers	Vocabulary	Videos	Duration	Data
Signum [97]	Iso.	German	25	455	11,375	8:26	RGB
MSASL [98]	Iso.	English	222	1,000	25,513	24:39	RGB
Polytropon [15]	Iso.	Greek	1	2,703	3,517	8:19	RGB-D
ITI GSL [96]	Iso.	Greek	7	310	40,785	6:26	RGB-D
AUTSL [14]	Iso.	Turkish	43	226	38,336	21:00	RGB-D
BOBSL [99]	Iso.	English	39	2,281	1,940	1,467:—	RGB
CSL [100]	Iso.	Chinese	50	500	125,000	67:45	RGB-D
Phoenix [13]	Con.	German	9	1,231	6,841	10:43	RGB
Phoenix-T [9]	Con.	German	9	1,066	8,257	10:32	RGB
CSL [68]	Con.	Chinese	50	178	25,000	100+:—	RGB-D
Dicta-Sign [95]	Con.	Greek	16	1,704	375	14:17	RGB
ITI GSL [96]	Con.	Greek	7	310	10,295	9:35	RGB-D
Signum [97]	Con.	German	25	780	19,500	55:18	RGB
ChicagoFSWild [16]	Fing.	English	160	26	7,304	1:47	RGB
ChicagoFSWild+ [43]	Fing.	English	260	26	55,232	—	RGB
SL-ReDu	Iso.	Greek	21	369	22,632	25:15	RGB-D
	Con.	Greek	21	408	5,930	8:24	RGB-D
	Fing.	Greek	21	24	1,554	2:17	RGB-D

This chapter describes our efforts in addressing the data needs of our research. First, already available databases are harvested from pre-existing works in the literature, allowing research on four automatic recognition tasks, namely that of isolated and continuous signing, continuous fingerspelling, as well as continuous cued speech phoneme recognition. Further, we overview our large-scale data collection effort to support the three recognition tasks of the SL-ReDu prototype system (isolated, continuous, and continuous fingerspelling). Specifically, we first briefly discuss the SL-ReDu corpus content. Then, we report our semi-automatic data collection process, which has been developed to facilitate speedier collection under the supervision of a single only person in the recording studio, enable fully automated data post-processing for corpus content organization, and allow automatic synchronization of the RGB and depth video cameras used in our studio setup. Further, in addition to the studio-based data, we have also collected GSL videos employing a typical web-cam under non-ideal visual conditions, in order to improve data variability and have a sufficient amount of data that lie closer to the envisaged operating conditions of the SL-ReDu prototype system. Parts of the collected data (both under studio and non-studio conditions) have been utilized in the development of the GSL recognizer integrated into the SL-ReDu prototype system.

The rest of this chapter is structured as follows: In Section 2.2, we overview corpora that were already available in the literature, which we utilized for the development of innovative and effective algorithms for SLR. Further, in Section 2.3, we briefly overview the SL-ReDu corpus content, and next we describe in detail our studio and non-studio data collection process, as well as the collected data, for both studio and non-studio environments. Finally, we conclude the chapter in Section 2.4.

2.2 Harvesting Pre-Existing SLR Corpora

We have conducted a thorough literature review concerning already existing SLR data resources in order to enable the development of SLR algorithms, before the collection of SL-ReDu data dedicated to the SL-ReDu project education use-case. This search has yielded three GSL datasets: Two of them have been collected in the past at the Athena Research Center, namely the GSL part of the *Dicta-Sign database* [95] and the *Polytropon parallel corpus* [15]. In addition to these, a third GSL dataset (*ITI GSL corpus*) has recently been released by the Visual Computing Lab at the Information Technologies Institute (ITI) of the Centre for Research and Technology-Hellas (CERTH) [96]. Another existing useful dataset of SL that is freely available in the literature is the *AUTSL dataset* [14], which is a large-scale, RGB-D Turkish SL corpus that has been created at Ankara University. In our experiments, we have also considered two German SL datasets, the well-known *RWTH-PHOENIX Weather 2014* [13] and *RWTH-PHOENIX Weather 2014T* [9], which have been collected from the German TV station PHOENIX news and weather forecast. Further, in the context of this Thesis we have also leveraged the *ChicagoFSWild dataset* [16] that constitutes the first collection of American sign language fingerspelling data, acquired from web videos. Finally, two publicly available cued speech datasets in *French* [48] and *British English* [101] are also considered in our experiments, allowing research on automatic recognition of cued speech. These corpora exhibit significant differences among them, concerning the acted task and the recorded subjects, thus offering desirable variation in signers, signing content, and recording setup.

As also mentioned before, three different GSL datasets are exploited for the development of the SLR algorithms. More details are provided below.

Dicta-Sign Database [95]: This is a multilingual corpus in four sign languages, involving



Figure 2.1: Sample frames from the *Dicta-Sign database* [95].

GSL. The corpus comprises 10 different tasks with session duration of approximately 2 hours on the same elicitation material, covering various interaction formats ranging from monologues to sequences of very short turns, also with different levels of predictability. To help annotators with their signing interpretation, a complex studio setup with seven cameras, two of them being stereo ones, capturing signing from different view-points: front, side, and bird's eye view. GSL video data, expressed by 8 pairs of different informants (16 signers in total) consisting of approximately 14 hours of signing, are provided at 720×576 -pixel resolution (see also Figure 2.1). Corpus annotations are based on the iLex export format [102], as well as ELAN [103, 104], and are provided at two different levels (signed sentence and signed word), containing labels and time-stamps. The corpus signed vocabulary consists of 1,704 unique words.

Polytropon Parallel Corpus [15]: Polytropon was created in an attempt to build a bilingual parallel corpus for the language pair of GSL/Modern Greek, in order to serve as a significant linguistic resource for addressing SL processing needs in the framework of human language technology applications, as well as investigating GSL recognition from video data. In particular, the dataset was built through a validation process based on a set of 2,000 lemmas, originally derived from the GSL segment of the Dicta-Sign corpus. The corpus volume in-

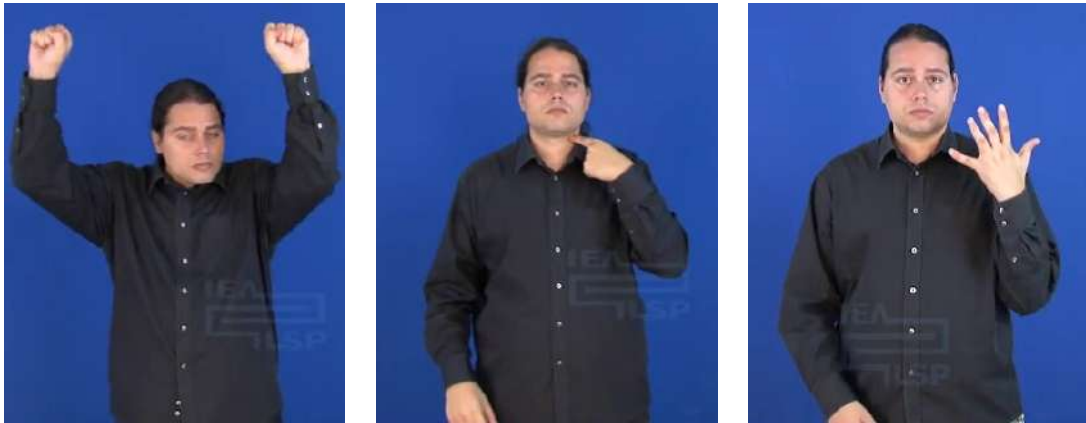


Figure 2.2: Example frames from the *Polytropon GSL corpus* [15].

corporate 3 repetitions of 3,600 sentences performed by a single signer, recorded by two frontal-view cameras, a Kinect and an RGB one. RGB video frames are shown in Figure 2.2. Video data are provided at a frame-rate of 25 Hz and 848×480 -pixel resolution. Corpus annotations based on ELAN are available at both the signed sentence and signed word level. The corpus signed vocabulary includes 2,703 unique words corresponding to proper nouns, adverbs, and verbs.

ITI Greek Sign Language Corpus [96]: This is a large-scale GSL corpus, involving real-life conversations, which has been introduced in an attempt to reinforce scientific research, especially GSL recognition, as well as GSL translation to Modern Greek. The database includes 5×3 different dialogues organized in sets of 5 individual tasks relevant to each of 3 public service types, i.e. police departments, hospitals, and citizen service centers, which are performed by 7 different native GSL signers. These dialogues concern communication between a deaf person and a single service employee, are pre-defined, and are performed by each signer 5 consecutive times, thus the total dialogues are $5 \times 7 \times 5 \times 3$. Signing is captured by an Intel RealSense D435 RGBD camera [27] at a rate of 30 Hz, providing simultaneously RGB data and a 24-bit depth stream at the same spatial resolution of 648×480 pixels per frame. During recording, camera pose adjustments are made, offering a desirable variation in the videos (see also Figure 2.3). Corpus annotation is conducted by GSL linguistic experts at both the signed sentence and signed word levels, yielding the *ITI GSL continuous dataset* and the *ITI GSL isolated dataset*, respectively. Both datasets' signed vocabulary consists of 310 unique glosses (40,785 gloss instances), while the *ITI GSL continuous dataset* includes 331 unique sentences (10,295 sentences), with 4.23 glosses per sentence on average.



Figure 2.3: Sample frames from the *ITI Greek Sign Language Dataset* [96].

In addition, we consider three additional corpora, one isolated Turkish SL corpus (see also Figure 2.4) and two continuous German SL datasets (see also Figure 2.5):

AUTSL Dataset [14]: This comprises a large-scale isolated Turkish sign language dataset that provides RGB, depth, and skeleton data at a frequency of 30 Hz, recorded using Microsoft Kinect [26]. It consists of 226 signs performed by 43 different subjects, yielding 38,336 isolated video samples (see also RGB sample frames in Figure 2.4). The dataset involves SL videos with 20 different backgrounds; the data are collected from different indoor and outdoor environments. Clipping and resizing operations to RGB and depth data are applied, leading to 512×512 pixel resolution.

RWTH-PHOENIX Weather 2014 Dataset [13]: This is a continuous German SL multi-

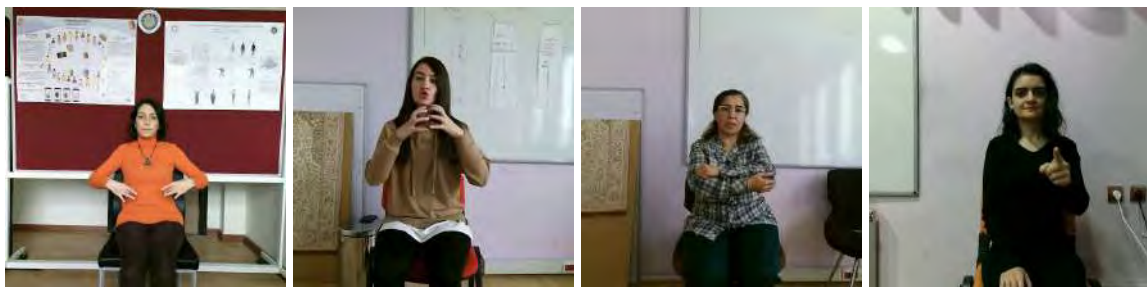


Figure 2.4: Sample frames from the *AUTSL dataset* [14].



Figure 2.5: Sample frames from the *RWTH-PHOENIX Weather 2014* [13] and the *RWTH-PHOENIX Weather 2014T* [9] datasets.

signer dataset, providing 6,841 different gloss sentences, extracted from the German TV station PHOENIX news and weather forecast. The corpus signed vocabulary consists of 1,231 unique glosses (around 80,000 gloss instances) performed by 9 different signers. Video data are provided at a frame-rate of 25 Hz and 210×260 -pixel resolution.

RWTH-PHOENIX Weather 2014T Dataset [9]: This dataset constitutes an expansion of the RWTH-PHOENIX Weather 2014 [13] corpus, involving both gloss and spoken language translation sentence pairs for German SL video samples of weather forecasts. The corpus volume involves in total 8,257 German SL sequences with frame resolution of 210×260 at a rate of 25 Hz expressed by 9 signers leading to a vocabulary of 1,066 unique glosses and 2,887 spoken language words, respectively.

Further, we employ the well-known *Chicago Fingerspelling in the Wild dataset* for building fingerspelling recognition algorithms (see also Figure 2.6):

Chicago Fingerspelling in the Wild Dataset (ChicagoFSWild) [16]: This dataset contains clips of fingerspelling sequences collected from online videos such as webcam videos and lectures, and annotated through ELAN. It consists of 7,304 American Sign Language (ASL) fingerspelling clips performed by 160 signers, annotated by students who have studied ASL



Figure 2.6: Example frames of the *Chicago Fingerspelling in the Wild (ChicagoFSWild)* [16], used here for continuous ASL fingerspelling.

(see also Figure 2.6). No frame-level labeling has been released, but only sequence-level labels are available. The corpus involves a vocabulary of 3,553 unique fingerspelled words. The majority of the fingerspelling sequences are right-handed (6782 sequences), with many fewer being left-handed (522 sequences) and even fewer two-handed (121 sequences). The frame resolution has a mean and standard deviation of $640 \times 360 \pm 290 \times 156$.

Finally, we tackle the cued speech recognition problem using two popular single-signer datasets, the *French* and the *British English cued speech dataset* (see also Figure 2.7):

French Cued Speech Dataset [48]: The French cued speech dataset composes of 2 repeti-

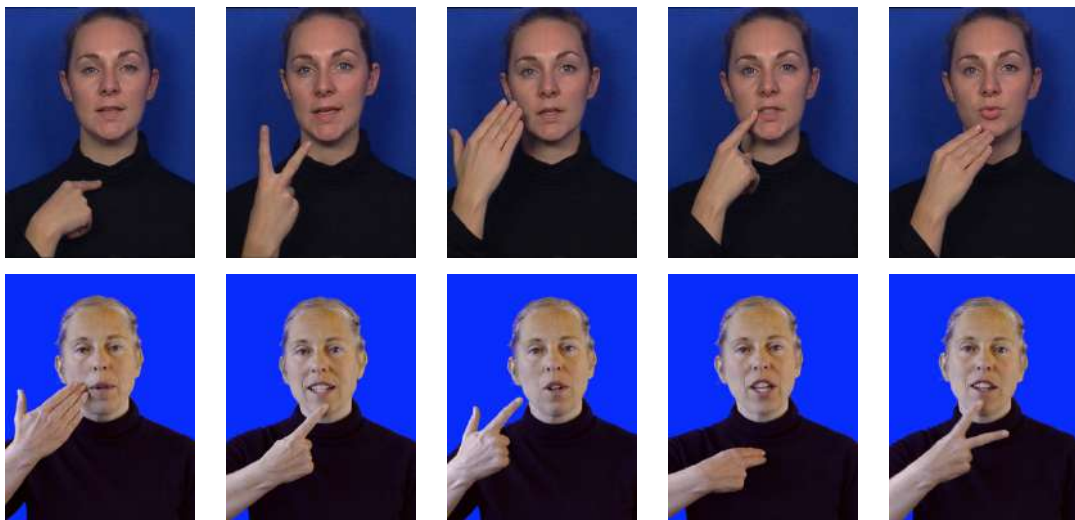


Figure 2.7: Example video frames from the French cued speech dataset [48] (upper row) and the British English cued speech database [101] (lower row) that are used in this Thesis, showing various combinations of hand shapes, mouthing patterns, and hand positions.

tions of 238 French sentences expressed by a professional cued speech interpreter consisting of about 11,770 phonemes totally. RGB video images including the interpreter's upper body are available at 50 Hz and 720×576 pixel resolution. The French cued speech encapsulates 8 lip patterns, 8 handshapes, and 5 different hand positions encoding a set of 34 phonetic classes that composes of 14 vowels and 20 consonants.

British English Cued Speech Dataset [101]: The British English cued speech dataset is recorded by a professional cued speech interpreter consisting of a set of 97 British English sentences. Color video images of the interpreter's upper body are available at 25 Hz, with a spatial resolution of 720×1280 . The dataset composes of 4 hand positions for encoding the 12 monophthongs and 4 hand slips for encoding the 8 diphthongs, while 8 hand shapes are used to encode the 24 consonants.

2.3 The Collected SLR Resources

We have created an SLR database aiming at developing the SL-ReDu prototype system's educational use-case in GSL. Such a dataset will also enhance the field of GSL recognition by enabling the creation of data-hungry, state-of-the-art deep-learning algorithms, providing desirable recording variability, a sizable signer population, and extended lexical coverage. As already mentioned, two different types of data have been collected as part of the SL-ReDu project, i.e., studio and non-studio data. For that purpose, signing data by multiple volunteer

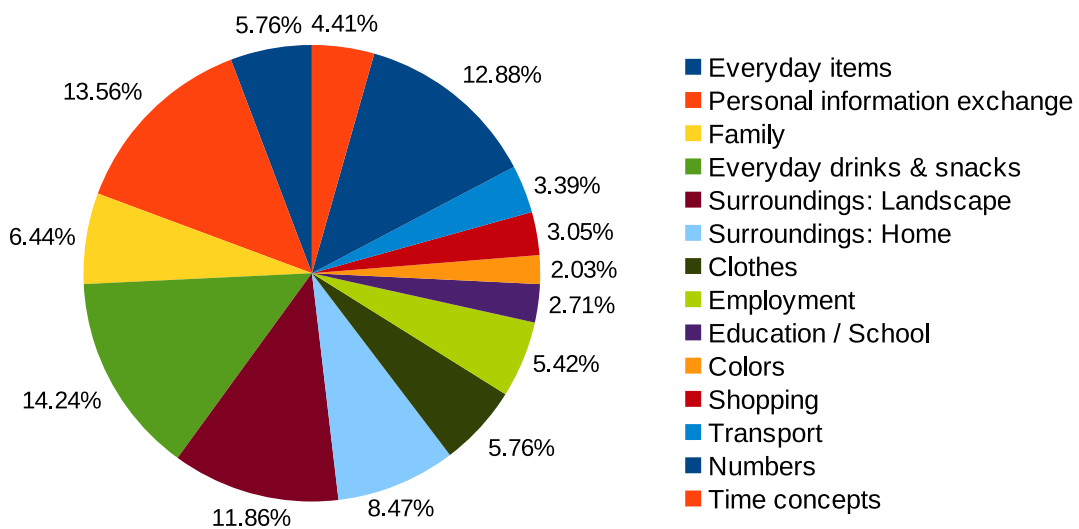


Figure 2.8: Distribution of the 369 GSL signs of the SL-ReDu isolated sign corpus part into 14 semantic categories.

informants (both native and non-native in GSL) have been collected. In this section, we first describe the corpus content (Section 2.3.1), followed by details of the studio data collection setup and recording methodology/protocol (Section 2.3.2), and finally details about the non-studio data collection process (Section 2.3.3).

2.3.1 SL-ReDu Corpus Content

The content of the corpus derives from the definition of the language material for levels A0-A1 of the Common European Framework of Reference for Languages (CEFR) [105] for GSL as L2, which is addressed in detail in our works [12, 106]. The corpus is organized in three subsets: (a) single-word GSL units; (b) GSL phrases; and (c) GSL alphabet fingerspelled words. The single-word content of the corpus covers isolated sign productions of 369 numeral and non-numeral lexemes. A distribution of the number of isolated signs per semantic category is shown in Figure 2.8. Regarding phrases, their formation entails noun-phrase constructions, all basic GSL predicative structures, and the multichannel productions for questioning and negation incorporating the necessary prosodic features. Specifically, 799 phrases (with about 3 glosses per sentence on average and a 294-gloss vocabulary) are involved in the SL-ReDu corpus content. Semantically, the content covers several important aspects of elementary communication skills involving personal information exchange, description of everyday objects, environment, numbers, time concepts, etc. Table 2.2 provides an overview of the typological categories of the corpus, along with examples of constructs corresponding to each type. Finally, concerning fingerspelling, both the 24 GSL alphabet letters in isolation, as well as continuously fingerspelled words are collected.

2.3.2 Studio Data Collection

We next detail the setup used for the studio data collection process, namely the two cameras employed for recording, as well as the tool developed for automated signing video data collection, and the synchronization of the two camera data streams.

Studio Setup: Studio video recordings took place at the recording studio of the Institute for Language and Speech Processing at Athena Research Center in Maroussi, Greece. The studio setup involves two cameras:

- A Sony HVR-Z7N camera with ClearVID 3CMOS technology, allowing for high-

Table 2.2: Core phrase types in the SL-ReDu GSL corpus with examples that include linguistic notation (capitalized) and translation into spoken Greek (lower-case, but provided here in their English translation for readability purposes).

Phrase Types	Examples (annotation/translation)
Affirmative	IX3 APPLE WANT He/she wants
Interrogative	MONEY HAVE IX2 Do you have money?
Negative	IX3 SIBLINGS HAVE He/she doesn't have siblings
Noun phrase	WATER cl-vol: thick a lot of water
Verb phrase	1GIVE3 give-to-multiple-recipients
Numerals	IX1 SIBLINGS HAVE THREE I have three siblings
Quantifiers	WATER cl-vol: thin a little water
Qualifiers	JUMPER cl-vol: thick a thick jumper
Classifiers	PLATE cl-SASS: round a round plate
Inflection for person	IX3 CYCLE CANNOT He/she can't cycle
Inflection for number	THREE-HOURS three-hours
Topic-comment	IX3 HAVE-A-WALK LIKE NOT He/she doesn't like going-for-walks
Agent-patient	2ASK3a,3b, 3c You ask them
Expression	IX2 PAY Have you paid?
Shoulders	IX3 COFFEE LIKE Does he like coffee?

definition video capture at a 1920×1080 -pixel resolution at 60 Hz (interlaced video). This camera involves pieces of hardware designed to be used independently of a computer to capture videos. Here, this external camera is “turned into” a “webcam” by connecting the camera’s HDMI output port via an HDMI cable to the female HDMI port of a special USB stick, which is connected to the computer using the male USB 3.0 port of the USB stick. This USB stick captures the HDMI video signal, converting

the external camera to a webcam.

- An Intel RealSense D435i camera [27] that records data at a rate of 30 Hz. In our collection, only the depth stream is acquired (at 848×480 -pixel resolution with 16-bit depth format), since high-quality RGB data are already available from the first camera.

The cameras are adjusted to record informants capturing the signing space. To increase variability in the videos, some informants are placed in a sitting position, while others in a standing one. Note that a monitor is placed in front of the informant to project the elicitation material.

Recording Process: For the studio recording process, we developed a tool which allows semi-automated signing video data collection. Specifically, the application starts with a message displayed in the form of an image in the monitor asking if the user is ready for the recording process. In case of a positive response, the person overseeing the collection process types “yes” in the terminal and, subsequently, a message is shown in the monitor stating: “Let’s see the signing video!”. Afterwards, a signing video is displayed in the monitor as elicitation material, in order to notify the informant of the sign to be performed. Note that our application provides the potential for watching the video as many times as the signer needs before signing. Specifically, the informant is asked through a message that is displayed in the

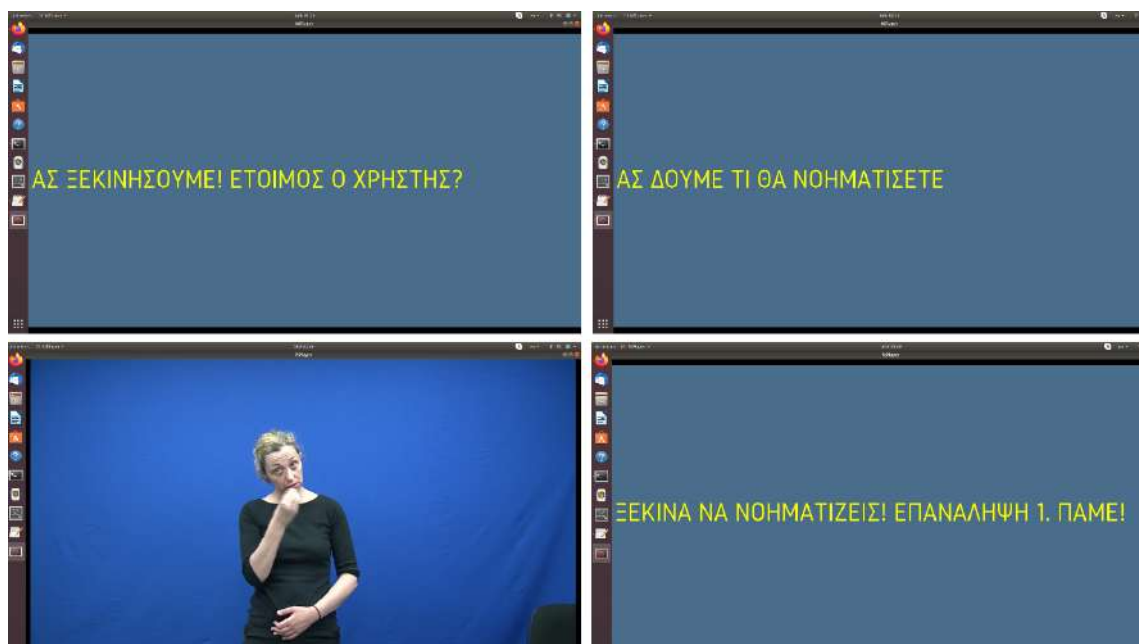


Figure 2.9: Snapshots of the monitor, showing messages and elicitation material provided to the informant.

monitor stating: “Do you need a repetition?”. When the signer comprehends the sign to be expressed, a message in the form of an image is displayed in the monitor stating: “Repetition 1. Let’s sign!”. Then the informant performs the specific sign that was displayed in the monitor. Both cameras then record for a specific time interval, depending to the sign type recorded, e.g., isolated sign with duration of 4 sec, fingerspelled word lasting 6 sec, or phrase signing that has 6 sec duration. The process is repeated three times for each isolated sign, while only once in the case of fingerspelling and phrases. Snapshots of the monitor messages are shown in Figure 2.9. To obtain synchronization of the two cameras, we adopt the use of multithreading in the Python programming language. Specifically, we use two different threads, one for each camera, where each one records for equal time duration, and their execution starts at the same exact time. This is made possible due to the connection of both cameras to the same computer.

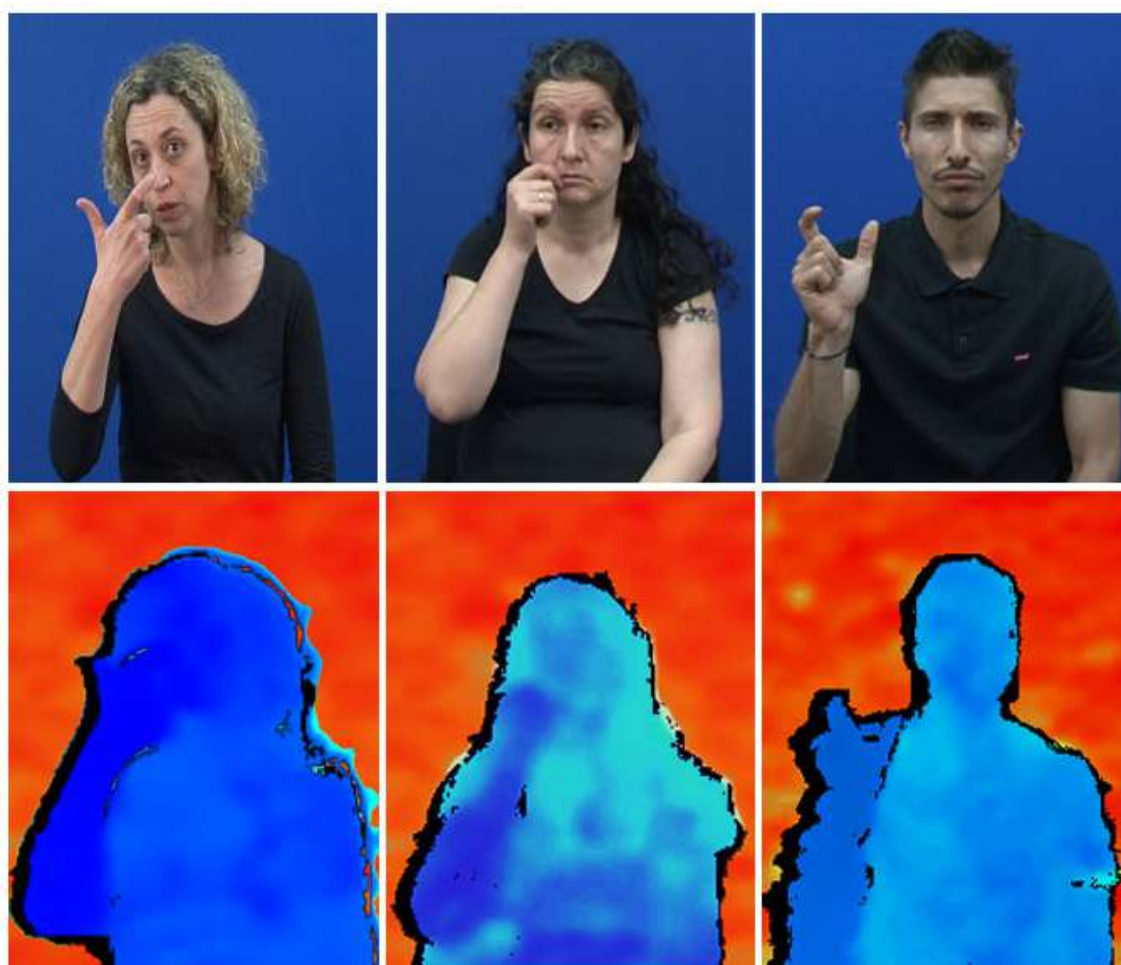


Figure 2.10: SL-ReDu corpus example frames from the RGB Sony HVR-Z7N camera (upper row) and the depth Intel RealSense camera (lower row).

After signing, the informant is asked if he/she needs a signing repetition and if the response is negative, the two signing videos, namely the RGB and depth one, are saved. As mentioned earlier, the Intel RealSense 435i captures only the depth stream at a frame-rate of 30 Hz and 848×480 -pixel resolution in order to reduce the data bandwidth, ensuring that few (if any) frames are lost during recording (dropped frames). For the Sony HVR-Z7N camera, as already discussed earlier, we record interlaced video at 60 Hz and 1920×1080 -pixel resolution. We have chosen video interlacing to be able to capture fine temporal information of SL articulation at some loss of vertical spatial resolution. Example frames of recorded RGB and depth streams are shown in Figure 2.10.

Collected Studio Data: The studio data contain recordings for all three SLR tasks by 21 native signers (11 females and 10 males), yielding 30,116 video files with a total duration of 36 hrs. Note that this is much larger than the only other existing multi-signer SLR resource for GSL, namely the ITI GSL corpus [96] that contains about 10k videos of 7 different signers and has a duration of less than 10 hrs. More specifically:

- For isolated signs, each of 21 signers articulated the vocabulary of 369 signs up to 3 consecutive times, yielding 22,632 videos with a total duration of 25 hrs and 15 mins.



Figure 2.11: Example frames from the continuous GSL studio data collected, showing the 21 different signers.

- In the case of continuous signing, the 799 phrases (with a vocabulary of 294 glosses and an average of 2.86 glosses per phrase) were divided to 3 different groups. Each of the 21 signers performed the phrases of one group once (one signer performed more), yielding a total of 5,930 videos with a duration of 8 hrs and 24 mins.
- In the case of fingerspelling, each informant signed once the 24 GSL alphabet letters in isolation, as well as a set of 50 fingerspelled words composed of 3-6 letters each, which was unique to each signer. Note that each signer signed each alphabet letter at least 4 times in total. This process yielded 1,554 videos with a duration of 2 hrs and 17 mins.

Note that the recorded videos were checked by GSL experts, removing videos with incorrect signing content. In Figure 2.11, example RGB frames of the continuous signing videos performed by the 21 informants under the studio recording conditions are shown. The studio data have been made publicly, available for downloading at: <https://sl-redu.e-ce.uth.gr/corpus>.

2.3.3 Non-Studio Data Collection

Recording Setup and Process: Non-studio signing data have been collected via a tool installed on a laptop, which allows automated video data collection. Specifically for elicitation, a message is shown at the laptop monitor stating: “Let’s see the signing video!”. Afterwards, a signing video is displayed on the monitor, notifying the informant of the sign to be performed. Then, the informant is asked by a message displayed on the monitor if she/he needs a repetition, providing the potential for watching the video as many times as needed. When the signer comprehends the sign to be expressed, a message in the form of an image is displayed at the monitor stating: “Let’s sign!”. Then the informant performs the specific sign.

The data have been recorded indoors, under realistic, non-studio conditions with varying background and lighting, using a Logitech C615 web-camera (connected to the laptop device) at a frame rate of 30 Hz, YUV411 video format, and 640×480-pixel resolution. To increase variability in the videos and enhance SLR robustness, some signers are placed in a sitting position while others in a standing one (see also Figure 2.12, where examples of non-studio data are given).

Collected Non-Studio Data: Non-studio signing data by volunteer informants (both native and non-native in GSL) have been collected for three tasks: isolated GSL recognition (a large



Figure 2.12: Example frames of non-studio recordings of isolated signs.

vocabulary of 369 signs is considered, including numerals), continuous GSL recognition (799 unique sentences), and continuous recognition of fingerspelled sequences of the 24 Greek alphabet letters. Specifically:

- For isolated signs, 14 signers (8 females and 6 males) performed once a number of signs from 50 to 90, while 3 signers performed 30 signs 3 consecutive times, resulting in a total of 1,267 videos. In addition, data by 20 signers (14 females and 6 males) have been collected, where each signer articulated 18 numerals 5 consecutive times, resulting in a total of 1,800 videos. Moreover, 17 signers (12 females and 5 males) articulated 36 non-numeral isolated signs five times, thus, resulting in a total of 3,060 videos. This process resulted in a total of 6,127 video data with a duration of 8 hrs and 51 mins, performed by 34 different signers.
- For continuous signs, 799 unique sentences were divided into sub-groups, with each of the 17 signers (11 females and 6 males) performing one group once, yielding 906



Figure 2.13: Example frames from the continuous GSL non-studio video data (14 out of the 17 signers are shown).

videos in total with a duration of 1 hrs and 51 mins (see also Figure 2.13).

- For fingerspelling, data from 12 signers (11 females and 1 male) have been recorded. Each informant signed once the 24 Greek alphabet letters in isolation, as well as 50 fingerspelled words (unique to each signer) composed of 4-5 letters. In addition, 7 of these informants performed 16 words (common to all) composed of 3-7 letters, and 3 signers expressed an extra 71 words of 4-5 letters. This process resulted in a total of 1,071 videos with duration of 1 hrs and 87 mins. Note that each informant has signed each letter at least 4 times.

2.4 Conclusions

In this chapter, we overviewed already available SLR databases in the literature with an emphasis on the existing corpora that we have exploited for SLR algorithm development and evaluation, covering four automatic recognition tasks, namely that of isolated and continuous signing, continuous fingerspelling, as well as continuous cued speech recognition. For that purpose, we considered a Turkish and three GSL isolated SL datasets, a Greek and two German continuous SL corpora, an American fingerspelling SL corpus, as well as a French and a British English cued speech set. Further, we presented the Greek SLR resources that we have collected as part of the SL-ReDu project, focusing on the education use-case of systematic teaching of GSL as second language. In particular, we summarized the collected resources

that were recorded under both studio and non-studio conditions suitable for GSL recognition, containing three distinct video subsets: (i) isolated signs; (ii) continuous signing; and (iii) fingerspelling. Moreover, we described in detail the database acquisition procedures, including lexical content, studio setup, and recording protocol. The collected data were employed in the training of the GSL recognizer integrated into the GSL learning prototype system. Note that the studio collected data, which constitutes the largest GSL database in the literature in terms of duration and number of signers, has been publicly released to the research community in [107].

Chapter 3

Detection and Tracking of Visual SL Articulators

3.1 Introduction

This chapter describes in detail the evaluated systems concerning the detection and tracking of manual and non-manual SL articulators. In more detail, a traditional image processing scheme is first explored that we introduced in [54], where facial information accomplished by an efficient off-the shelf detector drives skin-tone based hand segmentation complemented with motion tracking to drive localization. Further, we investigate recent state-of-the-art techniques in 2D and 3D human pose regression, as well as 3D shape and pose estimation. In more detail, we investigate the OpenPose [108], HRNet [109], and HPRNet [110] whole-body pose estimators, which detect 2D human body skeletal joints including hand skeleton and a number of facial landmarks of the signer. In addition, we infer 3D coordinates of the signer's body, face, and hand joints from RGB data using the MediaPipe holistic pipeline [111]. Further, we obtain 3D shape, pose, and facial expressions of the signer from single RGB images using the ExPose method [112], the FrankMocap [113] framework, and the PIXIE [87] estimator.

The remainder of this chapter is structured as follows: In Section 3.2, we overview the traditional scheme employed for hand detection and segmentation. Further, the pose-based methods used for detection and tracking of the visual articulators (both manual and non-manual) are explored in Section 3.3. In Section 3.4, we overview the experimental framework of the evaluation, followed by our experimental results in Section 3.5. Finally, in Section 3.6 we conclude this chapter.

3.2 Hand Detection and Type Classification

A traditional Hand Detection and Type Classification (HDTC) scheme, introduced in our previous work in [54], addresses the extraction of the signer's hands and their classification into left and right (from the camera's perspective), utilizing old-fashioned techniques for accurate localization and tracking of SL articulation, together with deep learning methods for the detection and recognition of hand types. Note that our approach assumes that the frontal head posture data of the signer are visible, which typically occurs in SL videos. More precisely, the system comprises two dominant components, namely the image pre-processing pipeline for hand region segmentation and the hand type classification stage, as also shown in Figure 3.1. In more detail, the objective of the pre-processing phase comprises the production of the set of proposal windows that are, subsequently, fed into the CNN-based hand type classification component.

The pipeline initiates with the estimation of the skin color range using the Viola-Jones face detection algorithm [52], which is known for its computational efficiency and high efficacy in real-time face detection. The primary goal of the face detection process is the extraction of the facial region central square, which precludes background information and therefore provides the most accurate representation of skin tone (see also Figure 3.2(a)). Specifically, the nose area, which covers 13% of the facial bounding box, is returned and, subsequently, converted to the YCbCr color space [114]. After transforming the full-frame into the YCbCr color space, the range of the corresponding YCbCr values of the extracted nose region is used to classify skin pixels (see also Figure 3.2(b)). In cases of missing face detection, the frame is subjected to skin segmentation in the YCbCr color space using specific threshold values defined in [114]. After skin segmentation, we impose segmented regions on motion tracking

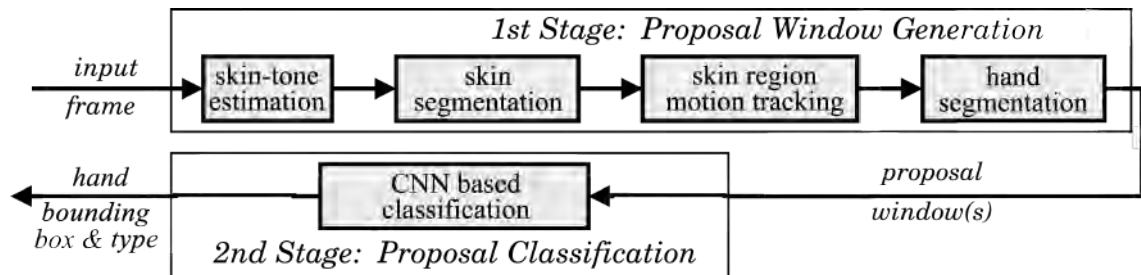


Figure 3.1: Block diagram of the hybrid, two-stage HDTC system employed for hand detection and type classification.

using the motion-based Kalman filtering technique [115] in an attempt to handle overlapping of hand and facial regions. Considering hands as the only moving objects in the frame, the Kalman filter algorithm assigns detections to relevant tracks based on their previous location (see also Figure 3.2(c)). Therefore, Kalman filtering is employed to localize and track the hand regions, while neglecting skin-like objects that do not correspond to moving tracks. Following the hand motion tracking step, rectangular bounding boxes enclosing the hand regions are generated. The image processing pipeline is then completed by removing the background from the bounding boxes using Otsu’s thresholding method [116]. The generation of the bounding boxes, also known as proposal windows, including the target objects, is crucial for the classification phase that follows.

In the classification phase, the resulting proposal windows are fed to a 2D-CNN classifier to classify them into three distinct classes: (i) “no hand”; (ii) “left hand”; and (iii) “right hand” (see also Figure 3.2(d)). For that purpose, we adopt the AlexNet 2D-CNN architecture [117], due to its wide endorsement by the computer vision community and its high-performance capability. In more detail, the hand rectangular bounding boxes are resized to the AlexNet appropriate input size (227×227) before being processed by the classifier for label prediction. The AlexNet CNN consists of five convolutional and three fully-connected layers, and it is pre-trained on the ImageNet corpus [118], which contains a large set of 1,000 labeled images

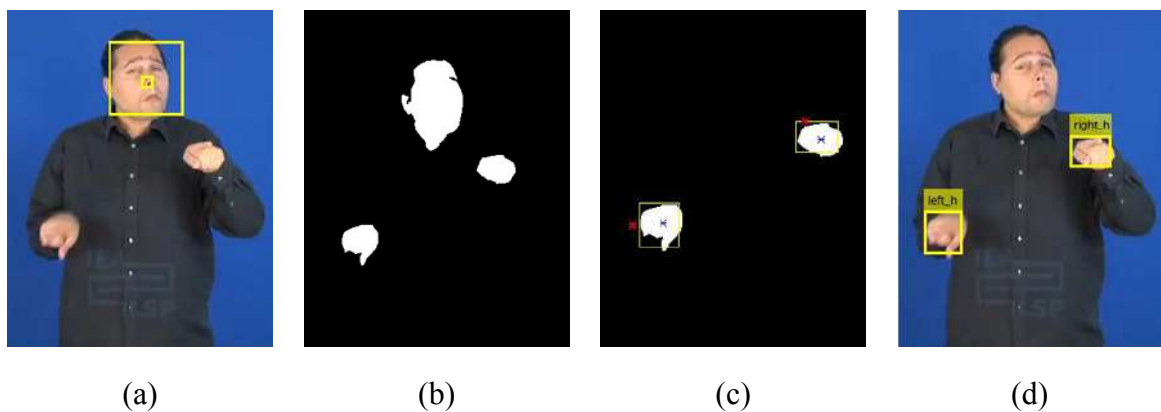


Figure 3.2: An example of the HDTC algorithm applied on the Polytropon GSL corpus [15]. Depicted, left to right: (a) image frame marked with a rectangular box encompassing the detected facial region and a central square of the nose region; (b) skin region segmentation; (c) hand tracking via Kalman filtering (yellow rectangles depict detected objects, red stars the predicted object positions, and blue stars their corrected positions); and (d) frame marked with rectangular boxes classified into the signer’s left and right hands.

for each of 1,000 categories. To adapt the network to our hand detection and type classification task, the final fully-connected layer is modified to have the same size as the number of classes of interest (three classes). The network is then fine-tuned using the Visual Geometry Group hand dataset from Oxford University [119] that includes around 6,000 labeled images of hand types. The dataset is randomly divided into training and test sets, with 70% being used for training and 30% used for testing. During training, Stochastic Gradient Descent with Momentum (SGDM) is used with an initial learning rate of 0.004, which is decayed by a factor of 0.5. The maximum number of epochs is set to 60, and a mini-batch size of 128 images is utilized.

3.3 Pose-Based Articulation Detection

Further, we explore various human pose regression schemes for detection and tracking of the manual and non-manual articulators, which infer 2D whole-body skeletal joints, including body, hand, and facial keypoints, from single RGB images using deep learning-based models. Further, we capture 3D human pose information incorporating the depth dimension and, thus, providing our SLR model with richer spatio-temporal representations of the input video. In addition, we employ the 3D joint rotation parameterization derived from monocular RGB videos through 3D whole body reconstruction models, yielding expressive 3D body pose and shape of the signer.

3.3.1 2D Skeleton Regression

In this section we explore various human pose detection models that infer 2D whole-body skeletal joints, including body, hand, and facial keypoints from single RGB images using deep learning-based models, namely the following:

OpenPose [108]: The OpenPose human joint detector [108] utilizes deep convolutional pose models to regress a detailed motion and structural representation of the human body, including body pose, hands, and face. The OpenPose network first extracts visual features using the initial 10 layers of the VGG-19 model [120], which are then passed to two parallel branches of convolutional layers. The initial branch produces a collection of confidence maps, with each one representing a distinct part of the human pose skeleton. The second branch generates a set of part affinity fields [108], which indicate the degree of confidence in the association

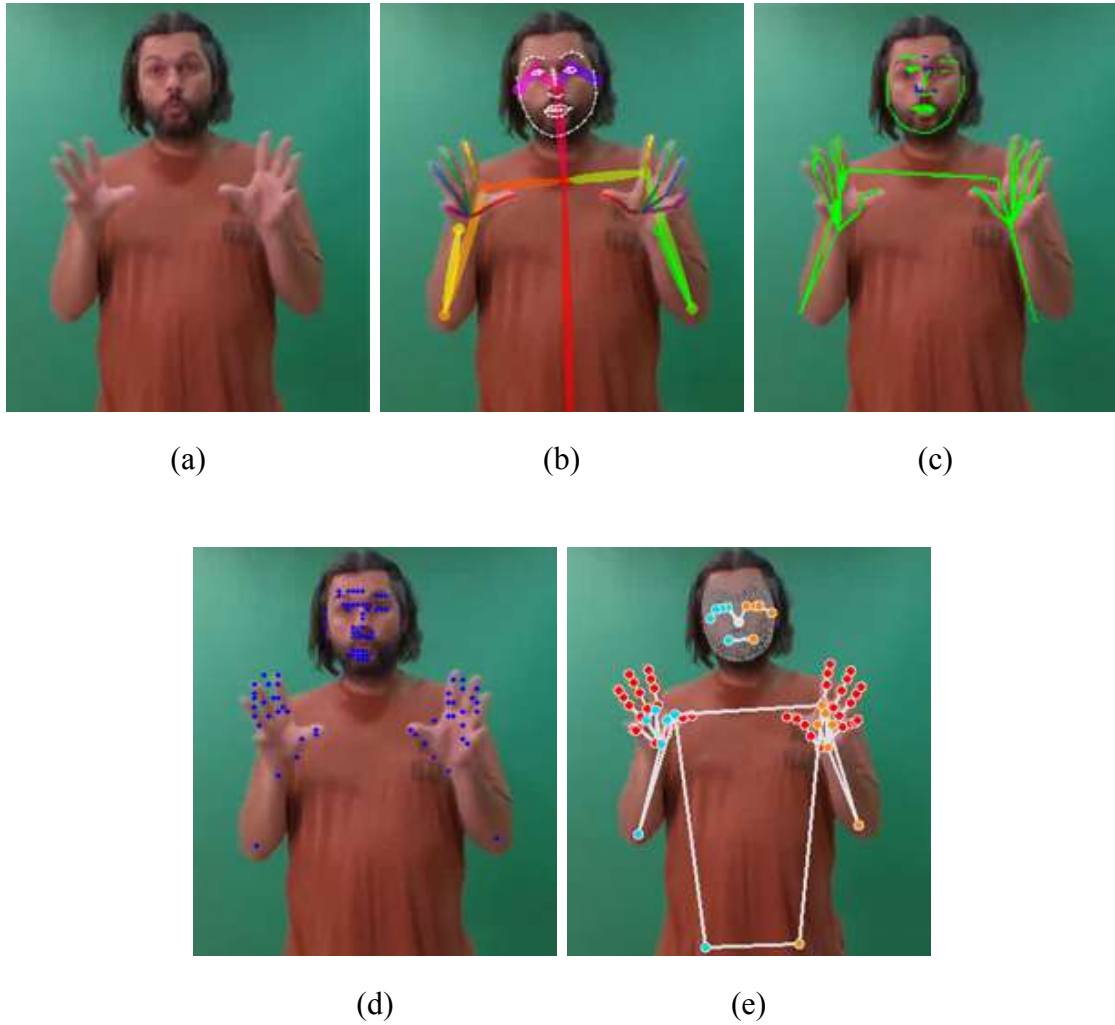


Figure 3.3: (a) Sample frame from the ITI CGSL dataset of [96]; (b) 2D human pose estimation via OpenPose [108]; (c) 2D human pose regression via HRNet [109]; (d) 2D human skeleton representation using the HPRNet framework [110]; and (e) 3D human skeleton representation using the MediaPipe holistic pipeline [111].

of each pair of the detected body parts. OpenPose renders a detailed representation of the human skeleton, capturing a total of 137 descriptors for the skeletal joints. The former descriptors are provided in the form of image coordinates. In more detail, OpenPose estimates 25 keypoints for body posture, 21 joints for each hand, and 70 landmarks for the face (see also Figure 3.3(b)).

HRNet [109]: The HRNet whole-body pose estimator relies on high-resolution convolutional networks maintaining high-resolution representations by connecting high-to-low resolution convolutions in parallel with repeated multi-scale fusions across parallel convolutions. The network involves multi-resolution blocks, each connecting high-to-low resolution convolu-

tions in parallel. The parallel processing approach of HRNet allows keeping high resolution through the whole neural network and hence the representation is more precise. HRNet estimates in total 133 whole-body landmarks from the RGB images, namely 23 body-pose skeletal joints, 21 joints for each hand, as well as 68 facial keypoints (see also Figure 3.3(c)).

HPRNet [110]: HPRNet is a single-stage end-to-end trainable network that estimates 2D whole-body keypoints. In HPRNet, the input image first passes through a backbone network and the output is fed to 8 separate branches, namely: (i) person center heatmap; (ii) person center correction; (iii) person bounding box weight and height; (iv) body keypoint offsets; (v) body keypoint heatmaps; (vi) hand keypoint offsets; (vii) face keypoint offsets; and (viii) face bounding box weight and height. The network results in a total of 133 keypoints, i.e., 17 human body keypoints, 68 face, 42 hand (21 keypoints for each), and 6 foot (3 keypoints for each) joints (see also Figure 3.3(d)).

3.3.2 3D Skeleton Regression

In addition to 2D skeleton information, we capture 3D human pose information in order to incorporate the depth dimension and allow our SLR model to obtain a richer spatio-temporal representation of the input video. For this purpose, we employ the MediaPipe holistic pipeline [111], which integrates separate models of the pose, face, and hand components, optimized for their respective domains. MediaPipe holistic estimates 543 3D whole human body joints (see also Figure 3.3(e)). Specifically, 33 3D joints of these correspond to the body skeleton, 21 to the 3D joints of each hand, while the remaining 468 ones correspond to the 3D coordinates of the facial region. Note that x and y landmark coordinates are normalized to the $[0.0, 1.0]$ range, based on the image width and height respectively, while the magnitude of the depth landmark, z , uses the same scale as x , with the midpoint of hips being the origin.

3.3.3 Expressive 3D Full Body Pose and Shape via 3D Reconstruction

In addition to the above, we also employ the 3D joint rotation parameterization inferred using 3D whole body reconstruction. We obtain expressive 3D body pose and shape of the signer, operating directly on image pixels. We investigate models that reconstruct fully expressive 3D humans from RGB images without relying on any intermediate features, such as 2D or 3D skeletal keypoints, namely the following:

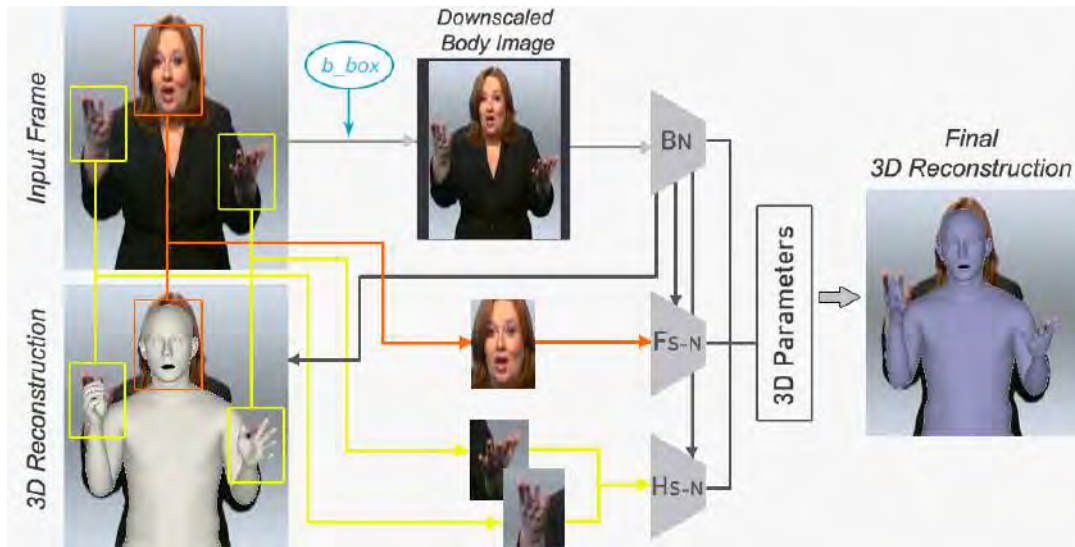


Figure 3.4: ExPose regression model [112] pipeline for 3D human body shape, pose, hand articulation, and facial parameter estimation.

ExPose [112]: The ExPose pipeline commences with taking as input a bounding of the human body retrieved through the Region-based Convolutional Neural Network (R-CNN) approach [121], which is fed to a full body neural network [122], obtaining a rough body pose estimate. To generate more reasonable representations of the hands and face that are often hard to estimate in such low-resolution images, ExPose approaches the reconstruction of the body, hands, and face separately, using part-specific models. Specifically, the recovered pose joints from the body network are projected on the original image, and they are subsequently employed for hands and face localization, as well as for generating high-resolution image patches for each Region-Of-Interest (ROI). Then, these patches are fed to hand- and face-specific sub-networks, refining the hand and face parameters (see also Figure 3.4). The ExPose framework parameterizes the human shape and pose, as well as the facial expressions through the SMPL-X contemporary parametric human body model [123]. In particular, both shape and expression are described by 10 coefficients derived from the PCA space, while the body pose includes 53 joints with 6 degrees of freedom, i.e., 22 body-pose joints, 15 joints per hand, and 1 for the jaw.

FrankMocap [113]: FrankMocap relies on an integration module that combines the outputs from two regression modules that predict 3D poses of the body and hands individually from a single RGB image input into a unified skeleton and mesh representation for both the body and hands. The body module and hand module contribute the different parts of the same output structures. The hand and body modules are designed based on deep neural network

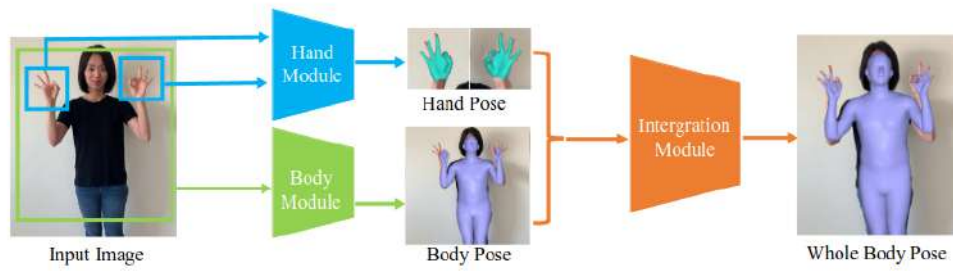


Figure 3.5: FrankMocap regression model [113] pipeline for 3D human body pose and hand articulation estimation (figure adapted from [113]).

techniques, and directly regress 3D poses from an input single RGB image. Given the 3D body and hand pose estimations from the hand and body modules, the integration is performed by direct copy-and-paste of the predicted body and hand poses to the parameterization model, achieving a near real-time performance for whole body 3D motion capture (see also Figure 3.5). Similarly to ExPose, the body pose includes 52 joints with 6 degrees of freedom, i.e., 22 body-pose joints and 15 joints per hand, yielding a 312-dim feature vector.

PIXIE [87]: PIXIE utilizes a moderator to deduce the 3D body pose and shape parameterization. PIXIE initially utilizes encoders to extract features from cropped images of the body, face, and hands. Subsequently, the moderator is trained to predict the confidence score for each individual part, which is used to calculate the weighted average of the body and part features. The fused features are then passed into separate hand and face networks to predict the parameters, which incorporate information from both whole-body and part images. The PIXIE framework estimates parameters for the human shape, pose, and facial expressions. In total, there are 250 coefficients to describe the shape and expression, while the whole-body pose consists of 55 joints with 6 degrees of freedom, i.e., 25 body pose joints including the head, jaw, and neck pose, as well as 15 joints per each hand.

3.4 Data and Experimental Framework

Performance evaluation of the investigated approaches is conducted regarding the detection and tracking accuracy of the hand articulation, using three publicly available isolated-sign GSL datasets described in detail in Section 2.2: the *Polytropon GSL corpus* [15], the *ITI GSL isolated dataset* [96], and the *Dicta-Sign database* [95]. Specifically: from the Polytropon GSL corpus, we utilize 7 videos from its single signer that include 106 frames in total

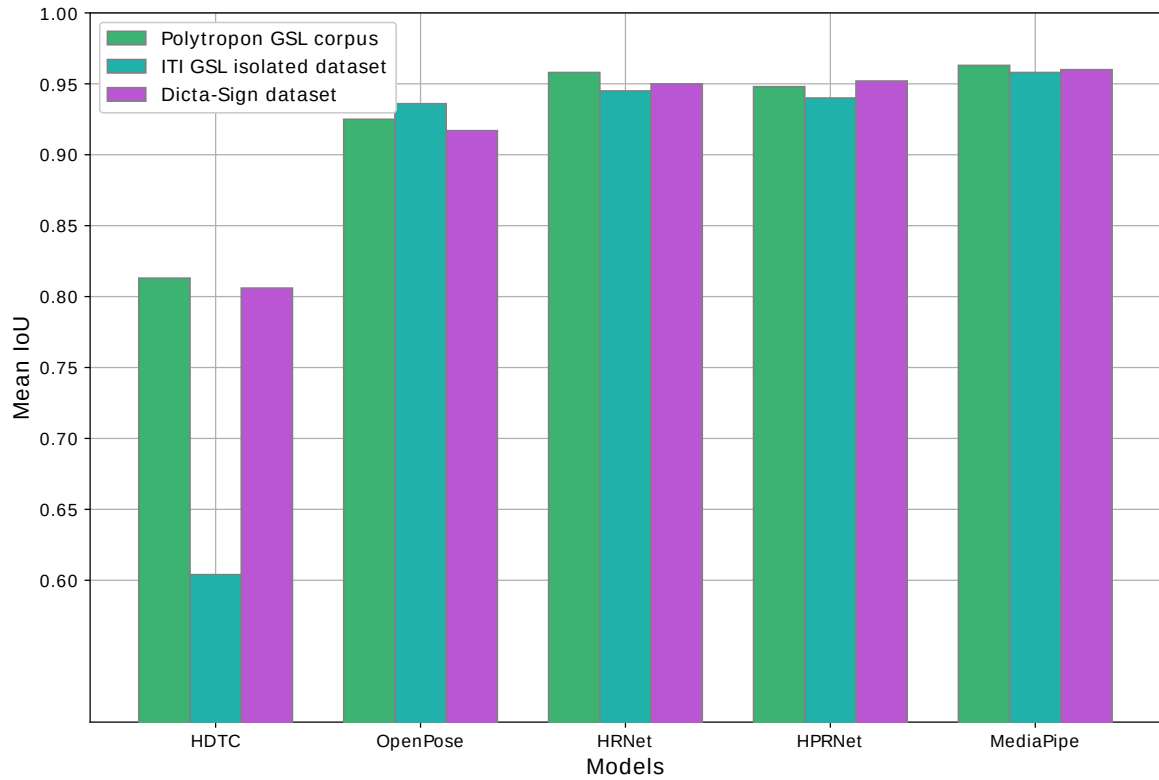


Figure 3.6: Accuracy comparison of the HDTC scheme against all 2D and 3D human skeleton regression frameworks on all three evaluation sets in terms of mean IoU between the ground-truth and the predicted hands' bounding boxes.

with resolution of 848×480 pixels; from the ITI GSL corpus, we use 7 videos (one per signer) that contain 96 frames with resolution of 648×480 ; and from the Dicta-Sign dataset, we employ 16 video snippets (one per signer) with 720×576 -pixel resolution. In total, this yields a total of 358 image frames from 24 signers, for which we manually annotate the signer hands' bounding boxes and label them as left or right hand.

3.5 Experimental Results

The set of experiments regards the detection and tracking accuracy of the investigated approaches described in Sections 3.2 and 3.3. Initially, the HDTC scheme, as well as the 2D and 3D skeleton regression models are evaluated in terms of the mean Intersection-over-Union (mean IoU) [124], a standard metric to measure the overlap ratio between the ground-truth and the predicted hands' bounding boxes. For comparison purposes, in the case of skeleton detection models we retrieve the corresponding bounding boxes of the hand region leveraging

the minimum and maximum values of x and y axis of the hand skeletal joints' coordinates. Note that the 3D pose and shape reconstruction models are not included in the particular evaluation, due to the fact that such frameworks suffer from the limitation of misalignment with the subject in the image (mesh-to-image misalignment) [87]. As shown in Figure 3.6, the MediaPipe 3D skeleton regression model reaches the highest IoUs in all three evaluation datasets, exceeding 0.95 IoU, while the HRNet and HPRNet 2D pose detectors follow. The lowest performance is achieved on the ITI GSL dataset (0.60) by the HDTC scheme, probably due to the video quality that is rather low as compared to the other two corpora, as well as the variability of standing posture of the signers. The MediaPipe holistic framework seems to outperform the considered alternatives on all three datasets, demonstrating the robustness of this deep-learning approach on signer 3D skeleton detection and tracking, providing low risk of false detections and missing hand candidate regions.

In Figure 3.7, the performance of the HDTC scheme is assessed against the OpenPose 2D skeleton detection library, as well as the MediaPipe 3D skeleton regression framework by providing the confusion matrices over all evaluation frames. As already mentioned, there are 358 frames in total where left and/or right hands participate in the signing process. In its computation, only detections with IoUs higher than 0.5 are considered as correct. Note also that the pre-trained CNN of Section 3.2 is used for the hand detection and type classification in the case of the HDTC scheme, while for the OpenPose library and the MediaPipe framework the hand type is automatically provided by the model. Clearly, the MediaPipe model performs better than the HDTC scheme and the OpenPose model, with 97.6% of its classifications being correct and only 2.4% wrong. Among the 717 detections, there are 10 cases where no hand participating is wrongly detected and classified as left, and 7 cases for the right hand one. This is primarily due to the fact that in many cases the MediaPipe holistic model returns the signer's hand skeletal landmarks, pursuing the arm and wrist joints, even when the hand region (i.e., fingers) is not visible in the image frame. On the other hand, the HDTC approach demonstrates lower accuracy performance with 91.6% correct classifications and 8.4% wrong. Here, there are 43 misclassifications between hand types, with 23 left hands being classified as right hands and 20 the other way. There are also 10 cases where no hand participating is wrongly detected as left hand, and 7 cases as right hand. Clearly, the OpenPose model performs better than the HDTC scheme, with 96.5% of its classifications being correct and only 3.5% wrong. Among the 717 detections, there are 7 cases where no hand

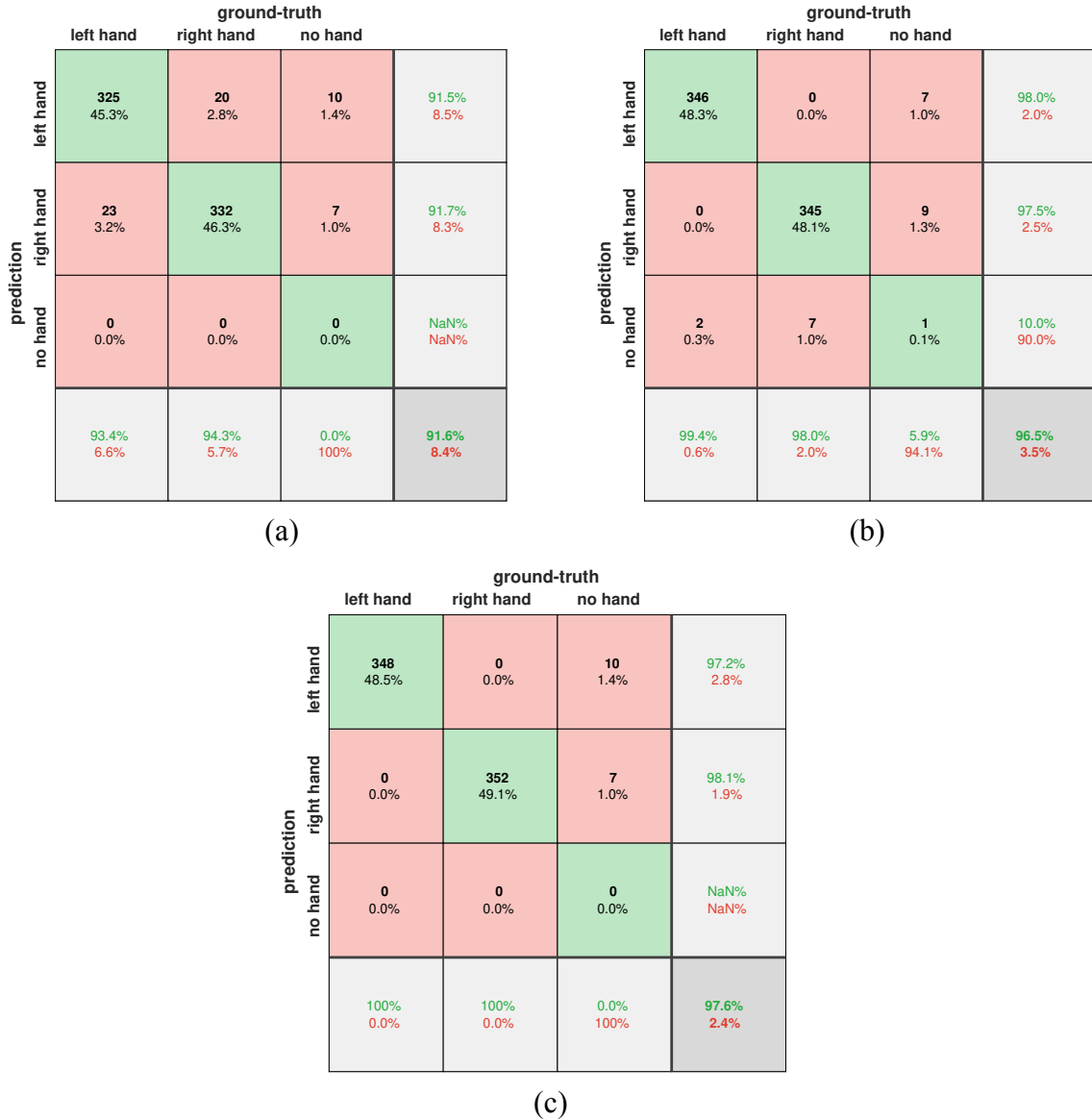


Figure 3.7: Confusion matrices of the (a) HDTC algorithm; (b) OpenPose; and (c) MediaPipe on all three evaluation sets (358 frames in total). Each row corresponds to the predicted class, while the columns represent the true class. The rate of all regions belonging to each class that are correctly and incorrectly classified are displayed in the row at the bottom, while the percentages of all the regions predicted to belong to each class that are correctly and incorrectly classified are displayed in the right column.

participating is wrongly detected and classified as left, and 9 cases for the right hand one. In addition, there are 2 cases, where left hand is wrongly detected as no hand and 7 cases, where right hand is wrongly detected as no hand. It is apparent that MediaPipe constitutes the best hand detector and tracker between the three evaluation systems, due to its detection accuracy.



Figure 3.8: 1st column: Sample frames from the Dicta-Sign dataset [95], the Polytropon corpus [15], and the ITI GSL dataset [96], imposed on 3D human body shape reconstruction through (2nd column) the ExPose regression model [112]; (3rd column) the FrankMocap framework [113]; and (4th column) the PIXIE [87] estimator.

Finally, in Figure 3.8 we evaluate the 3D body pose and shape reconstruction models of Section 3.3.3 qualitatively. As it may be observed, ExPose struggles with precise hand shapes, while FrankMocap with facial expressions and head rotations. PIXIE outperforms both with its facial expressions, as well as hand/body shapes and orientations being more realistic.

3.6 Conclusions

In this chapter, we presented our work concerning the detection and tracking of the visual articulators, which constitutes an important component in the SLR pipeline. Specifi-

cally, we performed localization and tracking of the various SL articulators of interest using a traditional approach, namely the introduced HDTC method, as well as deep learning-based approaches that capture 2D skeletal information, such as the OpenPose human joint detector, the HRNet whole-body pose estimator, and the HPRNet framework, and in addition 3D skeletal information using the MediaPipe pipeline. Further, we exploited the ExPose regression model, the FrankMocap framework, and the PIXIE estimator to obtain expressive 3D pose and shape of the signer. Our results on three isolated GSL datasets demonstrated that MediaPipe is the most accurate framework for human skeleton localization paving the way for its exploitation in the SL-ReDu prototype system discussed in Chapter 10, whereas the PIXIE regression model turns out superior to the considered alternatives for 3D body pose and shape reconstruction, as evaluated qualitatively.

Chapter 4

Extraction of Visual Features of Tracked Articulators

4.1 Introduction

In this chapter, we explore various visual features that are extracted following the SL articulator detection of Chapter 3. These emphasize articulator shape or appearance, while concentrating on static or motion patterns, thus providing a multitude of visual streams. Specifically, we first consider 2D and 3D skeletal features of the signer, as well as 3D expressive body pose and shape representations. In more detail, we explore a number of recent state-of-the-art models on this topic, such as the OpenPose [108], HRNet [109], and HPRNet [110] models for 2D human skeleton estimation, as well as the MediaPipe holistic library [111] in order to extract the signer skeletal joints in the 3D space. Further, we employ the ExPose [112], FrankMocap [113], and PIXIE [87] regression models to pursue 3D shape, pose, and facial expressions of the signer directly from single RGB images. Moreover, we investigate deep learning-based appearance and motion representations of the signing activity, exploring various 2D- and 3D-CNN image feature learners [85, 96, 125–130]. Finally, we investigate motion representations via different optical flow models [131–135].

To evaluate the investigated feature representations, we combine the resulting visual streams with a BiLSTM [78] encoder extracting spatio-temporal latent representations, before feeding them to the classifier (a single fully-connected layer, followed by a softmax) but in the case of 3D-CNNs we do not employ the BiLSTM encoder due to the 3D-CNN ability to model both spatial and temporal features (see also Figure 4.1). All systems are evaluated

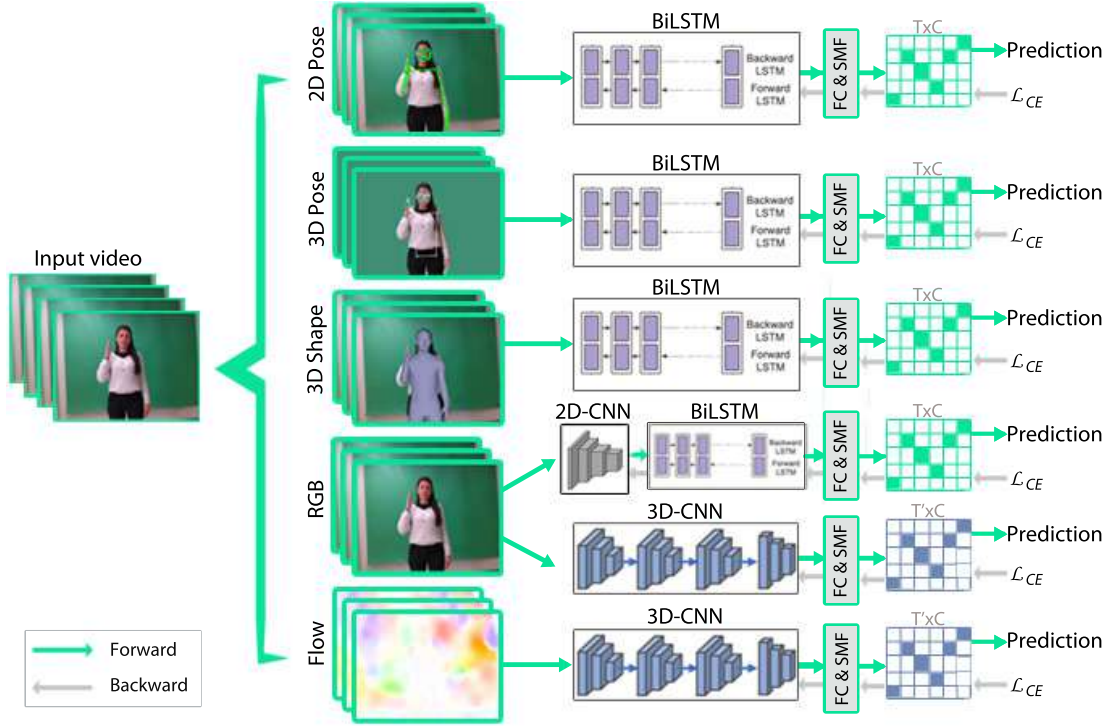


Figure 4.1: General overview of the various visual feature stream based models for isolated SLR. The following notation is employed: Fully-connected (FC), Softmax function (SMF), and Cross-entropy loss function ($\mathcal{L}_{CE}$). Gray backward arrows refer to backpropagation during training of the end-to-end models, namely the 2D-CNN model in conjunction with the BiLSTM and the SL classifier, as well as the 3D-CNN model with the SL classifier. Sample frames are from the *ITI GSL dataset* [96].

on two isolated-sign Greek SL datasets.

The remainder of this chapter is structured as follows: In Section 4.2, we describe in detail the extraction of a multitude of visual feature streams to be evaluated. In Section 4.3, we overview the data and experimental framework used in our evaluation, followed by our experimental results in Section 4.4. Finally, Section 4.5 concludes this chapter.

4.2 Extraction of Visual Features

In this section, we investigate a set of different visual modules that follow the SL articulator detection of Chapter 3, extracting visual feature representations from the input video. In more detail, a source video $\mathbf{x} = (\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_T)$ represented by T raw image frames is processed by the visual module, producing a sequence of image features $\mathbf{z} = (\mathbf{z}_1, \mathbf{z}_2, \dots, \mathbf{z}_T)$, which in turn passes through a back-end classifier module for SLR. For that purpose, we ini-

tially pursue 2D and 3D skeletal features of the signer, as well as 3D expressive body pose and shape features. Further, we explore deep learning-based appearance representations derived from the entire image frame, exploring various 2D- and 3D-CNN image feature learners. In addition, we investigate motion representations inferred from motion informative images, relying on different optical flow models. Details are provided in the next subsections.

4.2.1 Pose Features in 2D and 3D

Based on the modules discussed in Chapter 3, we consider the following skeleton-based feature representations, obtained by exploiting the upper-body and hand representations that are primarily relevant to signing:

OpenPose [108]: We employ 57 of the 137 estimated coordinates returned by OpenPose, discarding 10 human body joints associated with the invisible lower body parts of the signer, as well as the face keypoints. Specifically, we retain 15 body-pose and 42 hand-pose ones, all provided in 2D coordinates on the image plane, thus resulting in a 114-dimensional (dim) feature vector ($\mathbf{z} \in \mathbb{R}^{T \times 114}$).

HRNet [109]: We disregard 10 lower body joints, as well as 68 facial keypoints of the 133 skeletal joints returned by HRNet. Specifically, we employ 13 body-pose skeletal joints and 21 joints for each hand, all provided in 2D coordinates on the image plane, thus generating a 110-dim feature vector ($\mathbf{z} \in \mathbb{R}^{T \times 110}$).

HPRNet [110]: We retain 55 of the 133 HRPNet keypoints, namely 13 body-pose and 42 hand-pose ones, all provided in 2D coordinates on the image plane (discarding 10 lower torso keypoints and 68 facial landmarks). This yields a 110-dim feature vector ($\mathbf{z} \in \mathbb{R}^{T \times 110}$).

MediaPipe [111]: We exploit 67 of the 543 estimated 3D coordinates returned by the MediaPipe holistic framework, disregarding 8 human body joints associated with the lower body parts of the signer, as well as the 468 facial landmarks. In particular, we retain 25 body-pose and 42 hand-pose ones, all provided in 3D coordinates, thus resulting in a 201-dim feature vector ($\mathbf{z} \in \mathbb{R}^{T \times 201}$).

Note that in case of skeletal joint estimation failures, the missing features are substituted by the previous-frame existing ones. Further, to achieve translation and scale invariance, we normalize all extracted human skeletal joints by converting the image to a local coordinate system with the neck keypoint being the origin, and we apply further normalization based on the distance between the left and right shoulder keypoints.

4.2.2 Expressive Body Pose and Shape Representation in 3D

In addition to the above, we also explore the SLR contribution of the 3D joint rotation parameterization, inferred using 3D whole body reconstruction, as discussed in Section 3.3.3. We thus obtain expressive 3D body pose and shape representations of the signer, as follows: **ExPose** [112]: ExPose parameterizes shape and expression using 20 coefficients in total, while for the body pose it extracts 53 joints, namely 22 body-pose joints, 15 joints per hand, and 1 for jaw, with rotation representation dimension equal to 6. This yields 338-dim features ($\mathbf{z} \in \mathbb{R}^{T \times 338}$).

FrankMocap [113]: Similarly to ExPose, the body pose includes 52 joints with 6 degrees of freedom, i.e., 22 body-pose joints and 15 joints per hand. This results in a 312-dim feature vector ($\mathbf{z} \in \mathbb{R}^{T \times 312}$).

PIXIE [87]: PIXIE parameterizes shape using 200 coefficients of a linear shape space and expression using 50 coefficients. In addition, for the body pose it extracts 55 joints, namely 25 body-pose joints, including head, jaw, and neck pose, as well as 15 joints per hand, with rotation representation dimension equal to 6. This yields 580-dim features ($\mathbf{z} \in \mathbb{R}^{T \times 580}$).

4.2.3 Appearance Features

Appearance information of articulations plays a central role in SLR. Here, we consider such information at a global level, i.e., for the entire frame, using 2D- and 3D-CNN based feature extraction. In both cases (2D- and 3D-CNNs), after body-region cropping given a bounding-box generated via 3D skeletal coordinates estimated by the MediaPipe holistic model [111] (see Section 4.2.1), the body-cropped frame sequence is fed to the CNN for feature extraction. Specifically, we investigate five different multi-layer 2D-CNNs for this purpose:

ResNet-18 [136]: This is a 18-layer deep network pre-trained on the ImageNet database [118]. The network employs 17 convolutional layers and a single fully-connected layer, and it requires its input to be rescaled to 224×224 pixels. Feature maps are generated by taking the output of the last global average pooling layer, yielding 512-dim features ($\mathbf{z} \in \mathbb{R}^{T \times 512}$).

VGG19 [120]: This network constitutes a variant of the VGG model, consisting of 8 convolutional layers (3×3 convolutional kernels are used with stride one) and 3 fully-connected layers. VGG19 is pre-trained on the ImageNet database [118], and it requires input data rescaling to 224×224 pixels. The model yields 4096-dim output (at its FC8 layer; thus $\mathbf{z} \in \mathbb{R}^{T \times 4096}$).

ResNeXt-101 [127]: This is a neural network that reduces the number of hyperparameters required by the conventional ResNet [136]. This is achieved by the use of “cardinality”, an additional dimension on top of the width and depth of ResNet. Cardinality defines the size of the set of transformations. The network is based on the regular ResNet-101 model, substituting 3×3 convolutions inside the bottleneck block for 3×3 grouped convolutions. The network is pre-trained on the ImageNet corpus [118]. The network requires its input to be rescaled to 224×224 pixels. Feature maps are generated by taking the output of the last global average pooling layer, yielding 2048-dim representations ($\mathbf{z} \in \mathbb{R}^{T \times 2048}$).

MobileNetV2 [128]: This is a CNN, with its architecture based on an inverted residual structure, where the residual connections are between the bottleneck layers. The intermediate expansion layer uses lightweight depthwise convolutions to filter features as a source of non-linearity. As a whole, the MobileNetV2 contains the initial fully convolution layer with 32 filters, followed by 19 residual bottleneck layers. MobileNetV2 is pre-trained on the ImageNet database [118], and it requires input data rescaling to 224×224 pixels. Feature maps are generated by taking the output of the last 2D convolutional layer, yielding 1280-dim representations ($\mathbf{z} \in \mathbb{R}^{T \times 1280}$).

InceptionV3 [126]: This is a 48-layer deep CNN pre-trained on ImageNet [118], and it requires input image rescaling to 299×299 pixels. Feature maps are generated by taking the output of the last global average pooling layer, yielding 2048-dim representations ($\mathbf{z} \in \mathbb{R}^{T \times 2048}$).

In the four latter cases, in order to reduce feature dimensionality, a trainable linear layer is used on top of the extracted representations, resulting in 512-dim appearance-based visual features for each input frame ($\mathbf{z} \in \mathbb{R}^{T \times 512}$).

In addition, we examine the contribution of five 3D-CNN models to the task of SLR. Specifically, we consider the following models:

ResNet2+1D [85]: This is an 18-layer model that separates 3D convolutions to spatial 2D convolutions and temporal 1D convolutions. Specifically, 3D convolutional kernels of dimension $u \times K \times K$, where u denotes the temporal extent and K the spatial width and height, are replaced by spatial convolutional filters with dimensionality of $1 \times K \times K$, followed by temporal convolution filters of size $u \times 1 \times 1$. The model contains five (2+1)D convolutional blocks (each composing of spatial and temporal convolutions) and a global average pooling layer that operates on both space and time. The model weights are pre-trained on the Kinetics dataset [137]. The network uses 16 frames of resolution 112×112 as input, yielding a

512-dim feature vector.

Residual 3D network (R3D) [130]: This network performs 3D convolutions that are followed by batch normalization and Rectified Linear Unit (ReLU). R3D is an 18-layer model consisting of five 3D convolutional blocks followed by a global average pooling layer summing out both spatial and temporal information. R3D model weights are pre-trained on the Kinetics dataset [137]. The network uses 16 frames of resolution 112×112 as input and obtains a 512-dim feature vector.

Pseudo-3D Residual network (P3D) [138]: In this network, 3D convolutions are decoupled into $1 \times K \times K$ convolutional filters in the spatial domain and $u \times 1 \times 1$ convolutions tailored to the temporal domain. Specifically, the architecture consists of 199 layers involving different variants of pseudo-3D convolution blocks. In the first one, temporal 1D convolutional filters follow spatial 2D convolutional filters in a cascaded manner, while in the second one both filter types operate in a parallel fashion. The last one is a compromise between the first and the second block variant, by simultaneously building the direct influences among spatial and temporal convolutional filters, as well as the final output. The network uses 16 frames of resolution 160×160 as input, resulting in 512-dim features.

Inflated 3D ConvNet (I3D) [139]: This model builds upon the InceptionV1 model [140], but inflates the 2D convolutional filters and 2D pooling kernels to 3D, by endowing them with an additional temporal dimension. Particularly, filters are converted from square to cubic. Thus, the I3D model uses a 2D model architecture and also bootstraps the model weights from 2D pre-trained models. Such weights are pre-trained on the ImageNet corpus [118]. The I3D model uses 64 frames of resolution 224×224 pixels as input and yields a 512-dim feature vector.

3D ConvNet (C3D) [125]: C3D is a 3D-CNN, containing 8 convolutional and 5 pooling layers, followed by 2 fully-connected layers. The C3D model uses 16 frames of resolution 112×112 pixels as input, and results in a 4096-dim feature vector. For dimensionality reduction, a linear layer is used, resulting in 512-dim features.

Note that in the case of 3D-CNNs the input sequence is processed using a sliding window, where the input video is split into T' subsequences equal in length with the number of frames F used as network input, using a stride $s = F$. We properly apply zero-padding to the end of the video. There is no overlapping between subsequences, thus resulting in $T' = \lceil T/s \rceil$ subsequences. Each subsequence is propagated into the 3D-CNN model, yielding 512-dim

spatio-temporal appearance features in each case, thus $\mathbf{z} \in \mathbb{R}^{T' \times 512}$ overall.

4.2.4 Optical Flow Features

In addition to the above, motion information of the articulation process plays a crucial role in SLR. Typically, such information is extracted as optical flow. Here, to acquire it, we investigate the following four approaches that combine traditional optical flow algorithms with deep-learning techniques:

TVL1 [131]. This method is based on the minimization of a functional containing a data term using the L1 norm and a regularization term using the total variation of the flow. The main feature of this formulation is that it allows discontinuities in the flow field, while being robust to noise. The algorithm is an efficient numerical scheme, which solves a relaxed version of the problem through alternate minimization.

PWC-Net [134]. This method follows three main principles: pyramidal processing, warping, and the use of a cost volume. Initially, it warps the CNN features of the second image of the pair, employing the current optical flow estimate. Subsequently, a cost volume is generated, using the warped features of the second image, as well as the features of the first one. Finally, the cost volume is further processed by a CNN for optical flow estimation.

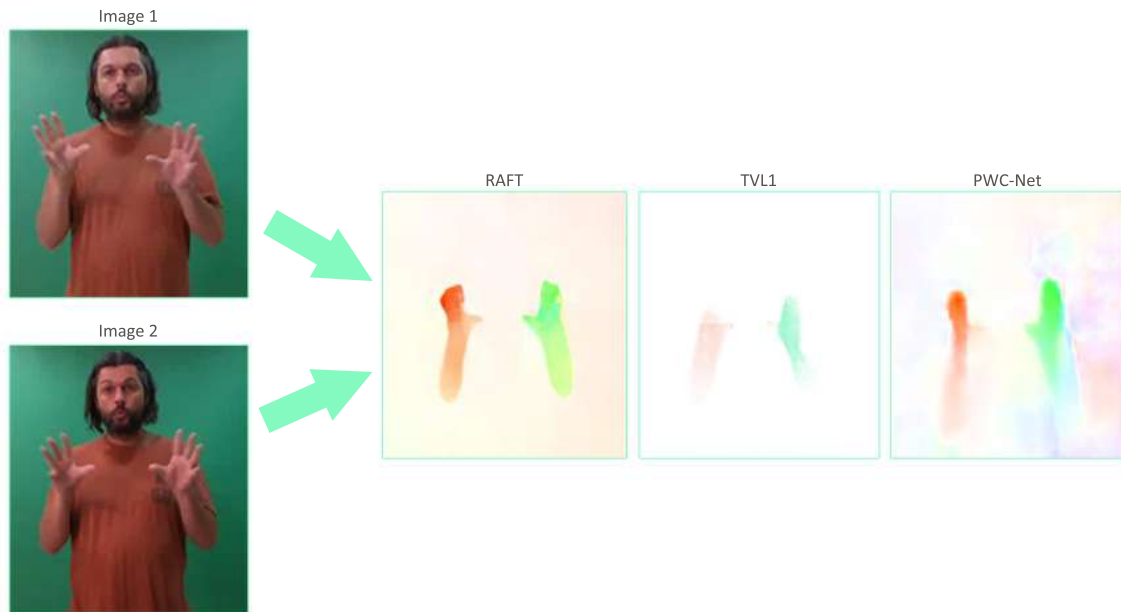


Figure 4.2: Optical flow motion informative images of two consequent sample input frames generated by RAFT [135]; TVL1 [131]; and PWC-Net [134]. Sample frames are from the *ITI GSL dataset* [96].

RAFT [135]. The network architecture consists of three main components: (i) a convolutional network based feature encoder that extracts per-pixel features from two adjacent frames, and a context encoder that extracts features from the first one; (ii) a correlation layer, which calculates the visual similarity between specific parts of the previous frame and each part of the subsequent frame image in a brute force fashion; and (iii) a Gated Recurrent Unit (GRU) based update operator that increases accuracy by iteratively performing inferences.

Once the optical flow is estimated using any of the aforementioned models, it is scaled to the input image size and stored as a two-band float image for both horizontal and vertical flow components in an optical flow file. Subsequently, motion informative images are generated by colorizing the displacement vectors acquired by the optical flow files (see also Figures 4.2).

Then, to extract motion representations, we feed sequences of colorized optical flow to the ResNet2+1D [85] model (one of the 3D-CNN models discussed in Section 4.2.3 that is pretrained on the Kinetics dataset [137]), resulting in 512-dim spatio-temporal appearance features, $\mathbf{z} \in \mathbb{R}^{F \times 512}$.

4.3 Data and Experimental Framework

The performance of the various visual modules, relying on 2D and 3D pose feature representations, 3D pose and shape parameterization, as well as appearance and motion features, in conjunction with an SL classifier back-end that is described later in this section, is assessed on two publicly available isolated-sign GSL datasets: the *Polytropon GSL corpus* [15] and the *ITI GSL dataset* [96], which are described in Section 2.2.

In the case of Polytropon, we consider an isolated small-vocabulary SLR task of vocabulary size 103 that involves signs with a sufficient number of occurrences appearing between 30 to 110 times (52.6 on average). A total of 5,414 video snippets are generated by segmenting the longer video database files through the ELAN annotation time-stamps corresponding to the glosses of interest. Evaluation is conducted using 10-fold cross-validation. For each fold, 80% of the data is used for training, 10% for validation, and 10% for testing.

In the case of the ITI GSL dataset, we employ its isolated signs part, containing 310 unique glosses, with a total of 40,785 gloss instances. For evaluation, we consider two different experimental paradigms: (i) the MS setting, where 10-fold cross-validation is employed and the training (80%), validation (10%), and test (10%) sets of each fold contain data of all 7

signers, and (ii) the SI setting, which uses 7-fold cross-validation with each fold involving data from only one test signer. The model is developed on the data from the remaining 6 signers, where 90% is used for training and 10% is used for validation.

To allow the evaluation of visual features, we build an isolated SLR system, where each feature stream of Section 4.2 is combined with a sequential module for sign classification. Specifically, both skeletal and 2D-CNN based appearance features are fed to a BiLSTM based encoder [78] for sign prediction. Such encoder consists of two BiLSTM layers (with hidden dimensionality equal to 512), followed by a SL classifier, i.e., a dense fully-connected layer coupled with a softmax layer, for class posterior generation. In particular, for the pose modality, the skeletal feature representations are extracted and, subsequently, fed into the classification model, i.e., the BiLSTM encoder and the dense fully-connected layer with softmax, which is trained for sign classification. On the other hand, for the 2D-CNN based RGB modality we build an end-to-end model, where the 2D-CNN appearance feature representations are jointly learned with the BiLSTM encoder and the SL classifier yielding sign prediction. In both cases, model training is conducted employing the Adam optimizer [141] with an initial learning rate of 0.001, decreased by a factor of 0.3 if the validation score remains consistent for 9 evaluation steps. We also employ dropout at a rate of 0.3 and a mini-batch size of 64. For training, we use the cross-entropy loss function, since isolated SLR constitutes a classification task. In the case of 3D-CNN based feature representations (appearance and optical flow), we couple the 3D-CNN model with the SL classifier, abstaining from incorporating any BiLSTM encoder due to 3D-CNN ability to model both spatial and temporal features. During training, the 3D-CNN model is jointly trained with the SL classifier, deploying the Adam optimizer with an initial learning rate of 0.0001 and weight decay of 0.001. For training, the cross-entropy loss function is applied with label smoothing [142]. The mini-batch size is fixed to 16. During inference, the predicted sign is obtained by calculating for each class the mean value of the elements of the probability distribution, derived from the softmax layer, along the temporal dimension and, subsequently, the `argmax()` operator is used calculating the class with the highest probability.

All models are implemented in PyTorch [143], and the experiments are performed on an Nvidia GeForce RTX 3090 GPU. It should be noted that OpenPose was run on Google Colaboratory [144], which is a cloud-based Jupyter notebook environment with an Nvidia Tesla K80 and Cuda 10.1 toolkit.

Table 4.1: Sign classification accuracy (%) of isolated GSL recognition on the Polytropon and ITI GSL corpora (the latter under both experimental settings), employing various networks of Section 4.2.

Visual representations		SLR Dataset →		Polytropon GSL	ITI GSL	
		Models ↓			MS	SI
Pose	2D skeleton	OpenPose	BiLSTM	84.91	86.50	89.25
		HRNet		92.58	94.22	93.46
		HPRNet		92.11	90.81	91.37
	3D skeleton	MediaPipe		93.61	92.38	93.70
	3D pose & shape	ExPose		86.77	92.14	91.73
FrankMocap		87.24	94.60	93.89		
PIXIE		95.06	95.88	93.92		
RGB	2D-CNNs	ResNet-18	BiLSTM	93.57	93.12	82.88
		VGG19		93.69	93.24	83.37
		ResNeXt-101		93.78	93.69	83.40
		MobileNetV2		92.20	90.31	82.44
		InceptionV3		93.84	93.80	83.55
	3D-CNNs	ResNet2+1D		98.52	98.25	95.89
		R3D		97.71	97.30	94.03
		P3D		96.14	96.49	92.14
		I3D		94.55	94.66	89.11
		C3D		94.07	93.17	85.66
Flow	3D-CNN	TVL1	ResNet2+1D	92.87	89.67	89.14
		PWC-Net		94.28	92.69	92.10
		RAFT		98.22	98.70	95.59

4.4 Experimental Results

We now proceed with our experiments, reporting sign classification accuracy (%) results of isolated GSL recognition on the two datasets of Section 4.3. Specifically, we first compare the recognition performance of the various feature representations of Section 4.2 in conjunction with the SLR back-end, discussed earlier in Section 4.3. Subsequently, we consider various feature combinations using late fusion.

In Table 4.1, we report results of the individual feature representations of Section 4.2, presented in three categories (pose, appearance, and optical-flow based features). In the case of the ITI GSL database, both MS and SI settings are considered. We can observe that, for both datasets of Section 4.3, the best feature sets from each category are, respectively, the PIXIE-based ones, the 3D-CNN based appearance features extracted by means of the ResNet2+1D network, and the optical flow features extracted by means of the RAFT model. Notably, the

Table 4.2: Sign classification accuracy (%) of isolated GSL recognition on the Polytropon corpus, when combining various well-performing visual features by late fusion. Note that in the upper three rows of the tabulated results, the corresponding row and column feature streams are fused. In the last row though, all five feature streams considered are combined.

Dataset →	Polytropon GSL	
Visual representations ↓	RGB (ResNet2+1D)	Flow (RAFT)
2D skeleton (HRNet)	98.60	98.35
3D skeleton (MediaPipe)	98.73	98.16
3D pose & shape (PIXIE)	98.89	98.54
All features	98.93	

latter two perform similarly, and both are significantly better than the PIXIE-based features. The best result on the Polytropon corpus is achieved by the ResNet2+1D on the RGB modality (98.52% accuracy), while on the ITI GSL dataset the best performance in the MS case is obtained by the RAFT optical flow model (98.70%) and in the SI case by the ResNet2+1D on the RGB modality (95.89%). It is notable that all 3D-CNN appearance models perform well, outperforming the respective 2D-CNNs coupled with the BiLSTM encoder, showcasing the contribution of 3D-CNN based spatio-temporal feature extraction to the SLR task. Further, in addition to the tabulated results, we evaluated the contribution of adding a BiLSTM encoder on top of the pure appearance-based ResNet2+1D. Our experiment on the Polytropon dataset showed that such architecture yields a slight accuracy improvement, reaching 98.87% (vs. 98.52% without it).

Next, we evaluate the combination of various visual feature representation streams by means of late fusion of the corresponding models. Note that, to fuse the different modalities (visual representations in conjunction with the SLR back-end) during inference we save the output of the last fully-connected layer of each stream before the softmax layer, and subsequently we sum them up. It should also be noted that in order to combine the 2D-CNN feature representations with the 3D-CNN ones, we split the 2D-CNN modality probability distributions into T' subsequences equal in length with the number of frames used as 3D-CNN network input, without overlapping, calculating the mean probabilities for each class of each subsequence along the temporal dimension. The output probability scores for each subsequence are concatenated and summed with the corresponding 3D-CNN ones. The result is passed through a softmax layer for sign prediction. In Tables 4.2 and 4.3, we report visual

Table 4.3: Sign classification accuracy (%) of isolated GSL recognition on the ITI GSL corpus (under both experimental settings), when combining various well-performing visual features by late fusion. Note that in the upper three rows of the tabulated results, the corresponding row and column feature streams are fused. In the last row though, all five feature streams considered are combined.

Dataset →	ITI GSL (MS)		ITI GSL (SI)	
Visual representations ↓	RGB (ResNet2+1D)	Flow (RAFT)	RGB (ResNet2+1D)	Flow (RAFT)
2D skeleton (HRNet)	98.31	98.74	95.82	95.55
3D skeleton (MediaPipe)	97.98	98.68	95.97	95.63
3D pose & shape (PIXIE)	98.57	98.77	96.04	95.64
All features	98.85		96.11	

feature fusion results on both Polytropon and the ITI GSL datasets. There, each cell refers to the sign classification accuracy (%) obtained by the combination of the corresponding row and column feature streams (with the exception of the last row, where all five feature sets listed in the table columns and rows are combined by late fusion). It should be noted that we employ the best performing feature stream of each category (2D skeletal features, 3D skeletal features, 3D pose and shape parameterization, appearance features, and optical flow features), as obtained in Table 4.1 (these streams are identical for all datasets and experimental settings).

We can readily observe that, in most cases, 2D and 3D skeletal features, as well as PIXIE-based ones improve results when combined with the appearance or optical-flow representations considered. In addition, fusing all feature streams significantly outperforms all individual ones of Table 4.1, as well as all other feature combinations of Table 4.2 and Table 4.3. Specifically, on the Polytropon dataset, late fusion of all five features yields 98.93% accuracy, a clear improvement over the 98.52% achieved by the best performing feature set. Similarly, on the MS (SI) setting of the ITI GSL, late fusion of all five feature streams yields 98.85% (96.11%), slightly improving over the best feature stream performance of 98.70% (95.89%).

4.5 Conclusions

In this chapter, we presented our research concerning the extraction of visual feature representations based on skeletal, appearance, and motion information of the signer. Specifically,

we performed skeletal feature extraction of the various SL articulators of interest through deep learning-based approaches that capture 2D/3D skeletal information and 3D joint rotation parameterization. In particular, for 2D human pose representation we applied the OpenPose, HRNet, and HPRNet frameworks, while for 3D skeletal features we employed the MediaPipe pipeline. Further, we exploited the ExPose, FrankMocap, and PIXIE models to obtain expressive 3D pose and shape parameterization. Moreover, we considered a number of visual feature representations to capture appearance information of the signer, employing various advanced 2D- and 3D-CNN based approaches. In addition, we leveraged the motion information of signing, extracted as optical flow through various traditional optical flow algorithms and deep-learning techniques. We reported results on two isolated GSL datasets, comparing the performance of the various visual features when combined with an SL classifier. The obtained results demonstrated that the PIXIE based skeletal features, the 3D-CNN based appearance representations derived by ResNet2+1D, and the optical flow features regressed by the RAFT model are the most robust representations, achieving the highest accuracies on both datasets. Finally, fusing all feature representations further improves SLR performance.

Chapter 5

Sign Language Recognition Architectures

5.1 Introduction

In this chapter, we build on our work on visual tracking and feature extraction, reported in Chapters 3 and 4, exploring various SLR back-end architectures operating on several visual modalities. In particular, we consider deep-learning encoder-based sequence learning models relying on a number of encoders, namely three RNN variations, i.e., the LSTM [77], BiLSTM [78], and GRU [145] architectures combined with a CTC based alignment module. In addition, we tackle SLR with sequence learning models based on attentional encoder-decoder architectures relying on four different encoding-decoding modules, namely the LSTM [77], BiLSTM [78], GRU [145], and CNN [44] models. In the last part of this chapter, we combine various visual feature streams of Chapter 4 within the investigated sequence learning frameworks via an early fusion scheme, namely feature concatenation at the model input. In addition, we apply a late fusion mechanism, where proper combination of the outputs of the sequence learning models operating independently on each stream is considered. We then evaluate all aforementioned modules on the continuous version of the *ITI GSL dataset* [96].

We devote Section 5.2 of this chapter to the development of single-stream recognition models, including sequential modules with and without attention. Specifically, we detail architectures and components for all considered schemes. In Section 5.3, we describe the two different fusion schemes across the five visual feature streams considered. We also provide the corresponding algorithmic implementation details and experimental framework in Section 5.4, as well as the evaluation results in Section 5.5. Finally, in Section 5.6 we conclude the chapter.

5.2 Single-Stream SLR

Since sequence recognition is mostly associated with RNNs, in this section, we investigate single-stream sequential modules relying on three different RNN encoding modules, namely the LSTM [77], BiLSTM [78], and GRU [145] encoders. For sequence decoding, we consider two different approaches, neither of which requires frame-level labels during training: CTC-based and attention-based decoding. Specifically, in the first case we employ an encoding module and a SL classifier (fully-connected layer and softmax) combined into a sequential model that is trained using the CTC loss function [81] (Section 5.2.1). Further, we address SLR from videos using encoder-decoder architectures equipped with attention (Section 5.2.2). For the latter, we investigate five such approaches, namely the attentional LSTM encoder-decoder, the attention-based BiLSTM encoder-decoder, the attentional GRU encoder-decoder, as well as the attentional CNN encoder-decoder. During inference, both approaches employ the CTC beam search decoder for sign prediction (Section 5.2.3). We provide more details in the next subsections.

5.2.1 Single-Stream Sequence Learning Models

As depicted in Figure 5.1, an input SL video of T image frames $\mathbf{x} = (\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_T) \in \mathbb{R}^{T \times 3 \times R \times W}$, where R and W denote the image height and width, is processed by a certain visual feature model producing a sequence of image features $\mathbf{z} = (\mathbf{z}_1, \mathbf{z}_2, \dots, \mathbf{z}_T) \in \mathbb{R}^{T \times v}$ with v denoting the feature vector size for each frame $\mathbf{z}_t$ (t denotes each time step). The visual feature representations are, in turn, fed into a sequence learning module, i.e. an encoding model that generates the sequential representations $\mathbf{h} = (\mathbf{h}_1, \mathbf{h}_2, \dots, \mathbf{h}_T) \in \mathbb{R}^{T \times v}$, followed by a fully-connected layer and a softmax activation, yielding frame-level gloss probabilities for each time step $p(s_t|\mathbf{x})$. The conditional probability of any gloss sequence $\mathbf{s} = (s_1, s_2, \dots, s_T)$, where $s_t \in \mathcal{W} \cup \{\text{blank}\}$ with $\mathcal{W}$ denoting the gloss vocabulary, is formulated as:

$$p(\mathbf{s}|\mathbf{x}) = \prod_{t=1}^T p(s_t|\mathbf{x}).$$

Thus, the conditional probability of the gloss label sequence $\mathbf{y}$ is the summation of all possible alignment paths:

$$p(\mathbf{y}|\mathbf{x}) = \sum_{\mathbf{s} \in \mathcal{B}^{-1}(\mathbf{y})} p(\mathbf{s}|\mathbf{x}),$$

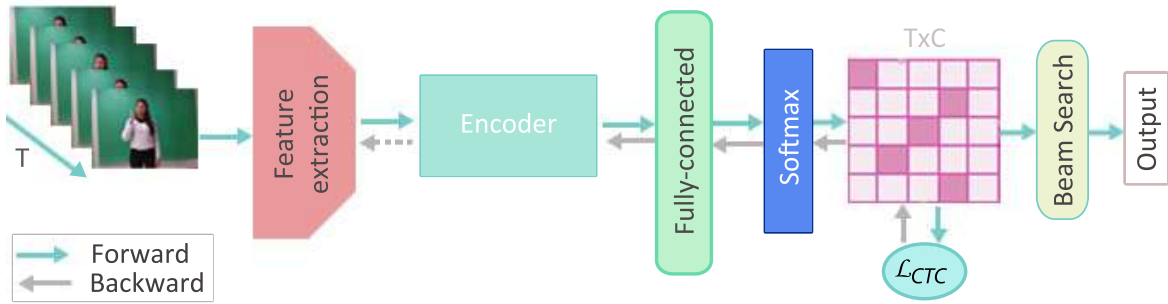


Figure 5.1: Algorithmic flow-chart of SLR relying on single-stream sequence learning models: Firstly, the signer’s video is processed by a suitable feature extraction model followed by a particular encoding module producing sequential representations. Subsequently, these representations are propagated to an appropriate classifier, i.e., a fully-connected layer coupled with softmax, for propability distribution regression, yielding the recognized output via the beam search method. During training the CTC loss function is employed. Backward arrows refer to backpropagation during training of the encoder and the SL classifier, while dashed backward arrows correspond to joint training of the 2D-CNN feature learner with encoder and SL classifier. Note that in the case of skeletal feature representations there is no end-to-end training.

The alignment path s is mapped to the target sequence y with the many-to-one mapping function $\mathcal{B}$ that removes blanks and repeated labels from s . Subsequently, an inverse operation $\mathcal{B}^{-1}(y) = \{s | \mathcal{B}(s) = y\}$ is used to represent all possible alignments corresponding to the target label sequence y . The CTC loss function is defined as:

$$\mathcal{L}_{CTC} = -\log p(y|x).$$

Next, we provide details regarding the encoding models used for sequence latent representation extraction:

LSTM [77]: Since in SLR each sign depends on previous context, we employ a stacked LSTM encoder to extract feature representations. In its general form, the LSTM encoder is fed with the feature representations z extracted from the visual module, generating hidden state $h_t = LSTM(x_t, h_{t-1})$ at time step t .

BiLSTM [78]: Instead of predicting the current output based on the past context, as in the case of LSTMs [77], BiLSTMs in addition to the previous observations access also the future ones. This is obtained by applying two LSTMs: one processing the input sequence forwards (left-

to-right) and the other backwards (right-to-left), both related to the same output, generating respective hidden state representations. These are then concatenated, generating hidden states $\mathbf{h}$.

GRU [145]: We also investigate the GRU-based encoder that has been introduced as an improved version of the recurrent block to adaptively capture various time-scale dependencies, involving fewer parameters and operations than the aforementioned encoders. Similarly to LSTM and BiLSTM based sequence learning, the GRU encoder maps an input sequence $\mathbf{z}$ to a sequence of hidden states $\mathbf{h}$.

5.2.2 Single-Stream Sequence Learning Models with Attention

In addition, we consider attention-based encoder-decoders, motivated by recent work in automatic speech recognition [146–148] and machine translation [149–151] that are equipped with a multi-step attention mechanism. Specifically, we investigate a number of alternative schemes, namely three attentional RNN variations, based on the LSTM [77], BiLSTM [78], and GRU [145] architectures, as well as the attentional CNN encoder-decoder architecture introduced in our work in [44].

In encoder-decoder architectures, the feature representations $\mathbf{z} = (\mathbf{z}_1, \mathbf{z}_2, \dots, \mathbf{z}_T)$, inferred from the input image frame sequence $\mathbf{x} = (\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_T)$ through the visual module, are passed to an encoder producing the sequential representation $\mathbf{h} = (\mathbf{h}_1, \mathbf{h}_2, \dots, \mathbf{h}_T)$. Then, $\mathbf{h}$ is fed into the decoder returning the elements of the output sequence $\mathbf{y} = (y_1, y_2, \dots, y_N)$ one by one. In attention-based architectures the context vector $\mathbf{c}_n$, which regards the likelihood of each chunk of image frames being relevant to the current output at time step n , is computed as the weighted sum of all hidden states $\mathbf{h}_t$ with $t = 1, 2, \dots, T$. Supposing that $\mathbf{d}_n$ denotes the decoder hidden state at time step n , the attention scores a_{nt} are given by a score function normalized by softmax:

$$a_{nt} = \text{softmax}(\text{score}(\mathbf{h}_t, \mathbf{d}_{n-1})),$$

For this task, several alternatives have been proposed, like the alignment function in [151]:

$$a_{nt} = \text{softmax}(\tanh(\mathbf{W}_h \mathbf{h}_t + \mathbf{W}_d \mathbf{d}_{n-1})),$$

where $\mathbf{W}_h$ and $\mathbf{W}_d$ are the model parameters (weights and biases), and the one in [152]:

$$a_{nt} = \text{softmax}(\mathbf{h}_t^\top \mathbf{d}_{n-1}). \quad (5.1)$$

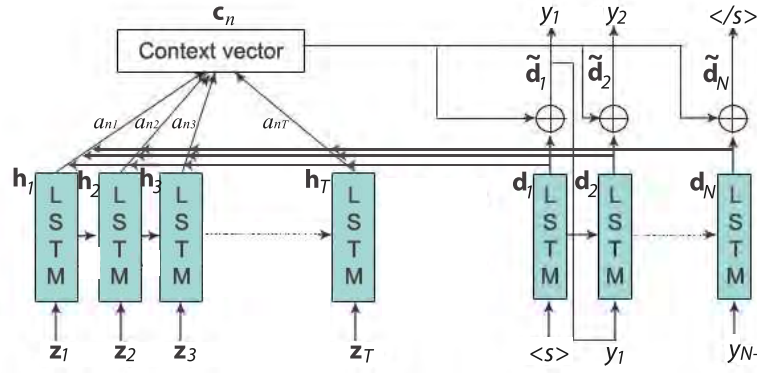


Figure 5.2: General architecture of an attention-based RNN encoder-decoder based on LSTMs, with z_t denoting the representations derived from the visual module and y being the output prediction.

The context vector c_n for $n = 1, 2, \dots, N$ derives from the weighted sum of each encoder hidden state:

$$c_n = \sum_{t=1}^T (a_{nt} \mathbf{h}_t).$$

The predictive distribution of outputting symbol y_n , $p(y_n | \mathbf{y}_{<n}, \mathbf{x})$, is given by:

$$p(y_n | \mathbf{y}_{<n}, \mathbf{x}) = \text{softmax}(\mathbf{W}_s \tilde{\mathbf{d}}_n),$$

with $\tilde{\mathbf{d}}_n$ being the attentional hidden state that derives from the concatenation of the target hidden state $\mathbf{d}_n$ and the context vector c_n :

$$\tilde{\mathbf{d}}_n = \tanh(\mathbf{W}_c [\mathbf{c}_n; \mathbf{d}_n]).$$

Thus, the loss function of the attention-based encoder-decoder models is formulated as:

$$\mathcal{L}_{Att} = -\frac{1}{N} \sum_{n=1}^N \log p(y_n | \mathbf{y}_{<n}, \mathbf{x}).$$

As already mentioned, we investigate a number of alternative attention-based encoder-decoder schemes. More details follow:

Attention-Based LSTM Encoder-Decoder: In its general form, the LSTM encoder [77] is fed with representations $\mathbf{z}$ derived from the visual feature learner, generating hidden state representations $\mathbf{h}_t = \text{LSTM}(\mathbf{x}_t, \mathbf{h}_{t-1})$. During the decoding process, hidden state sequence $\mathbf{h}$ is processed by the LSTM decoder producing the elements of the output sequence $\mathbf{y}$, one by one, incorporating the attention mechanism, as discussed above (see also Figure 5.2).

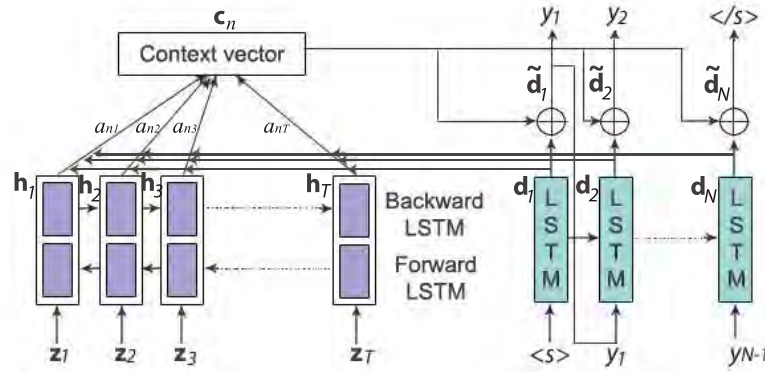


Figure 5.3: General architecture of an attention-based RNN encoder-decoder based on Bi-LSTMs, with z_t denoting the representations derived from the visual module and y being the output prediction.

Attention-Based BiLSTM Encoder-Decoder: Similarly, during encoding the sequence z derived from the visual feature learner is fed to a BiLSTM encoder [78], generating state representations h . Specifically, during encoding sequence z is fed to a forward LSTM encoder, generating state representations $\vec{h}_t = LSTM(x_t, \vec{h}_{t-1})$. In addition, the input sequence is also processed in the opposite direction $\overleftarrow{h}_t = LSTM(x_t, \overleftarrow{h}_{t+1})$. These two calculations are concatenated generating hidden state h_t . Then, as in the attention-based LSTM encoder-decoder, hidden states h are processed by the LSTM decoder, producing the predicted output (see also Figure 5.3).

Attention-Based GRU Encoder-Decoder: In the two previously discussed encoder-decoder architectures, given the input sequence z , the LSTM was adopted as encoder to estimate the corresponding sequence of hidden state h . However, LSTM training is susceptible to problems such as the vanishing gradient and the exploding gradient as described in [153]. Therefore, GRUs [145] have been introduced as an improved version of the recurrent block to adaptively capture various time-scale dependencies. Similarly to LSTMs, GRUs have a gating mechanism adjusting the information flow within them, however with fewer parameters and operations. Similar in spirit to the attentional LSTM / BLSTM encoder-decoder model, the attention-based GRU encoder-decoder comprises two processes, namely encoding and decoding, incorporating the attention mechanism and operating in a similar fashion (see also Figure 5.4).

Attention-Based CNN Encoder-Decoder: Another sequence learning architecture introduced in the literature is the attention-based CNN encoder-decoder [44], which relies on

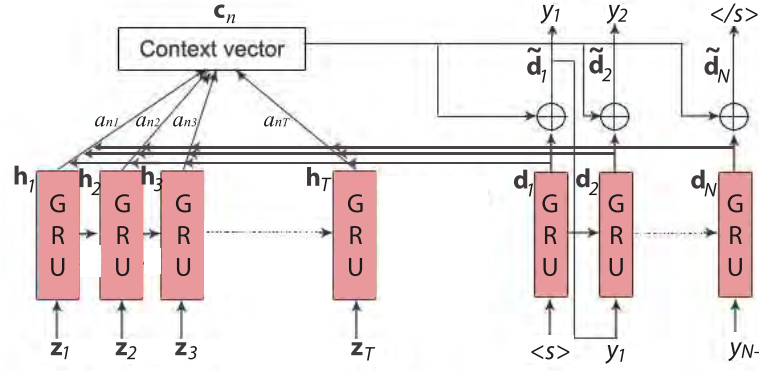


Figure 5.4: General architecture of an attention-based RNN encoder-decoder based on GRUs with z_t denoting the representations derived from the visual module and y being the output prediction.

convolutional block structures on both encoder and decoder to compute the latent state representations (see also Figure 5.5). The attention-based architecture relies on multi-layer convolutional encoder-decoder neural networks. Specifically, both encoder and decoder employ a convolutional block structure to compute the latent state representations $\mathbf{h}$ and $\mathbf{d}$. For simplicity purposes, we assume that the output of the l -th encoder's layer is expressed by $\mathbf{h}^l = (\mathbf{h}_1^l, \mathbf{h}_2^l, \dots, \mathbf{h}_T^l)$ and the l -th layer of the decoder returns $\mathbf{d}^l = (\mathbf{d}_1^l, \mathbf{d}_2^l, \dots, \mathbf{d}_N^l)$ hidden states. Considering a single-layer decoder with kernel width k , each output hidden state $\mathbf{d}_n^1$ will be associated with k inputs. Thus, if we add multiple layers on top of each other, under the assumption that later layers will process k outputs of the previous ones, the resulting state will be related to more inputs than previously. More precisely, stacking 5 layers with kernel width $k = 5$, leads each hidden state $\mathbf{d}_n^5$ output to be determined by 21 inputs. Thus, k concatenated input elements embedded in v dimensions, $\mathbf{Z} \in \mathbb{R}^{k \times v}$, are provided as inputs to a convolutional kernel $\mathbf{K} \in \mathbb{R}^{2v \times kv}$ outputting $\mathbf{H} \in \mathbb{R}^{2v}$, with twice the dimensionality of input elements.

As in [154], each layer composes of a one-dimensional convolution followed by a Gated Linear Unit (GLU) [155]), which behaves as a gating scheme that assists in dealing with the convolution output $\mathbf{H}$:

$$U(\mathbf{H}) = \mathbf{P} \otimes \sigma(\mathbf{C})$$

with $U(\mathbf{H}) \in \mathbb{R}^v$ denoting which of $\mathbf{P}$ inputs are relevant with the current target element through the sigmoid non-linear gate $\sigma(\mathbf{C})$, $\mathbf{P}, \mathbf{C} \in \mathbb{R}^v$ being the outputs of the convolution ($\mathbf{H} = [\mathbf{P}\mathbf{C}] \in \mathbb{R}^{2v}$) and $\otimes$ indicating point-wise multiplication.

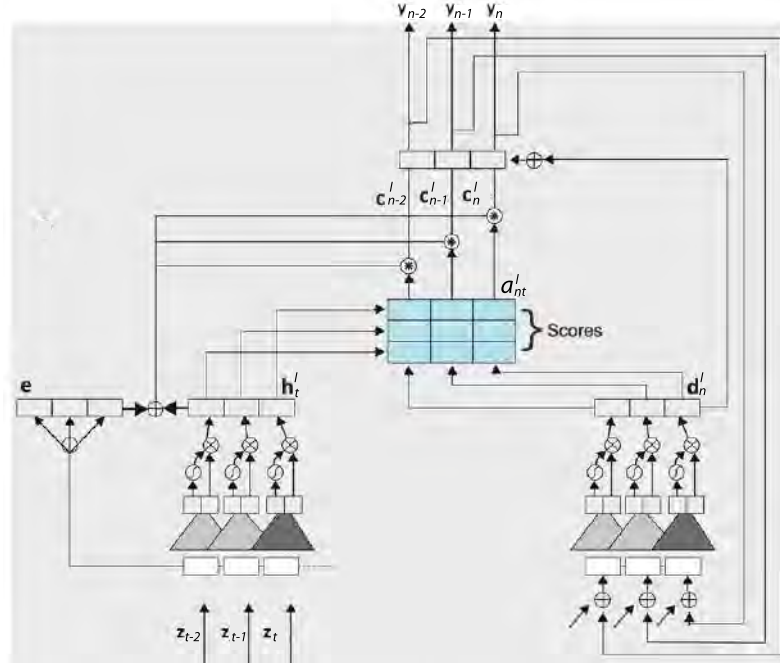


Figure 5.5: General architecture of the attention-based convolutional encoder-decoder model with attentions computed as dot products between decoder's context representations (bottom right) and encoder representations (bottom left).

To smoothly optimize and leverage the performance of deep convolutional networks, residual functions with reference to each convolution input and output are added as in [136]. Moreover, in order to retain the length of the encoder convolutional layer output as the input length, we add zero padding to the inputs of each layer. On the other hand, for the decoder inputs we add zero padding by $k - 1$ elements on both sides and then remove k elements from the output end to avoid the inclusion of future information during decoding as in [154, 156].

A multi-layer CNN encoder-decoder model implies a multi-step attention mechanism [154, 157]. More precisely, as shown in Figure 5.5, the provided variable z by the feature extractor is fed into the multi-layer convolutional encoder outputting the hidden states h_t^l for the top layer l . Consequently, the weights a_{nt}^l are computed through an alignment function conferred upon the corresponding decoder layer d_n^l and each output of the last encoder layer h_t^l . The context vectors c_n^l , which derive from the weighted sum of each output of the last encoder layer combined with the embeddings e_t of the input elements z_t in distributional space:

$$c_n^l = \sum_{t=1}^T a_{nt}^l (h_t^l + e_t),$$

are in turn fed into the next decoder layer exploiting specific information during attention calculation. Once the context vector c_n^l of the last layer has been computed, it is concatenated

with the decoder hidden state returning the attentional vector $\tilde{\mathbf{d}}_n$.

In contrast to the work in [154], we avoid the addition of input elements $\mathbf{z}_t$ position embeddings in the encoder-decoder inputs since it seems to hurt our model performance. Instead, we use an input feeding scheme [152] in which the attentional vectors $\tilde{\mathbf{d}}_n^l$ are concatenated with inputs at the following time step. As it turns out, input feeding is beneficial for our model, owing to the fact that previous attentional vectors are taken into account during the current alignment score calculation. Thus, through this modification our model turns into a fully-connected deep neural network in both directions (vertical and horizontal) that deploys substantially previous alignment information during the estimation of the new ones.

5.2.3 Inference via Beam Search

In this section, we describe the beam search strategy [158], a decoding method used by the CTC-based sequence learning models of Section 5.2.1 and the attention-based ones of Section 5.2.2 during inference. Let $\mathbf{x}$ denote the source input sequence, which is processed by the sequential module producing the sequence $\mathbf{h}$. The sequential representation $\mathbf{h}$ is further mapped to a probability distribution over all possible outputs at each time step by the classifier. Assuming that each input sequence $\mathbf{x} = (\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_T)$ is paired with a target sequence $\mathbf{y} = (y_1, y_2, \dots, y_N)$, where N is the maximum number of tokens in the output sequence, then during inference the probability of an output at time step n is conditional, represented as follows:

$$p(y_n) = p(y_n | \mathbf{y}_{<n}, \mathbf{x}).$$

Thus, we choose the sequence that maximizes the conditional probability at each time step given input $\mathbf{x}$:

$$\hat{\mathbf{y}} = \operatorname{argmax} \prod_{n=1}^N p(y_n | \mathbf{y}_{<n}, \mathbf{x}).$$

The most dominant inference decoding method is the beam search [158], which stores the top- b best scoring candidates at each time step, as well as their scores, where b denotes the beam width. In particular, at time step n_0 the beam search keeps the b highest scoring candidates, while as moving to the next time step, the b candidates are extended picking the top- b candidate expansions. The process is repeated iteratively to the next steps until the maximum number of tokens is reached. Therefore, the score of a candidate sequence $\mathbf{y}$ is

computed as follows:

$$\log p(\mathbf{y}) = \sum_{n=1}^N \log p(y_n | \mathbf{y}_{<n}, \mathbf{x}).$$

The former function has an unfavorable bias towards shorter sequences due to the small values obtained when summing logarithms of values less than 1. To tackle this, we aid longer sequences by dividing the result by the number of tokens raised to the power of a length penalty $c = 0.6$, choosing the output sequence which maximizes the following score:

$$\frac{1}{N^c} \log p(\mathbf{y}) = \frac{1}{N^c} \sum_{n=1}^N \log p(y_n | \mathbf{y}_{<n}, \mathbf{x}).$$

5.3 Multi-Stream SLR Fusing Multiple Modalities

As already mentioned in Section 5.1, we employ as visual features the most efficient approaches of Chapter 4, starting with the HRNet framework [109] to detect the human skeleton in the RGB image frame and extract the respective 2D skeletal features. Subsequently, we exploit the MediaPipe [111] holistic pipeline to obtain 3D human skeletal information, including body pose, face, and hand landmarks. Further, we employ the PIXIE regression model [87] to pursue 3D shape, pose, and facial expressions of the signer directly from RGB images. Moreover, we adopt 2D appearance and motion representations of the signing activity using the InceptionV3 [126] image feature learner and the RAFT [135] optical flow algorithm. It should be noted that the best performing approach regarding appearance representations of Chapter 4 is obtained by the ResNet2+1D model, but since its training is more computationally expensive and time-consuming compared to 2D-CNNs due to the input high dimensionality, we choose to employ in our experiments the best performing 2D-CNN appearance feature learner. Thus, we exploit the InceptionV3 for both appearance and motion feature representations. In this section, we combine the aforementioned feature streams within the investigated sequence learning frameworks via an early fusion scheme, namely feature concatenation at the model input, as well as a late fusion mechanism, where an appropriate combination of the outputs of the sequence learning models operating independently on each stream is considered.

Early fusion: In early fusion, as also illustrated in Figure 5.6, the five visual feature stream representations are fused before encoding by simple concatenation. In particular, 3D human body pose, shape, hand articulation, and facial features estimated by the PIXIE regression

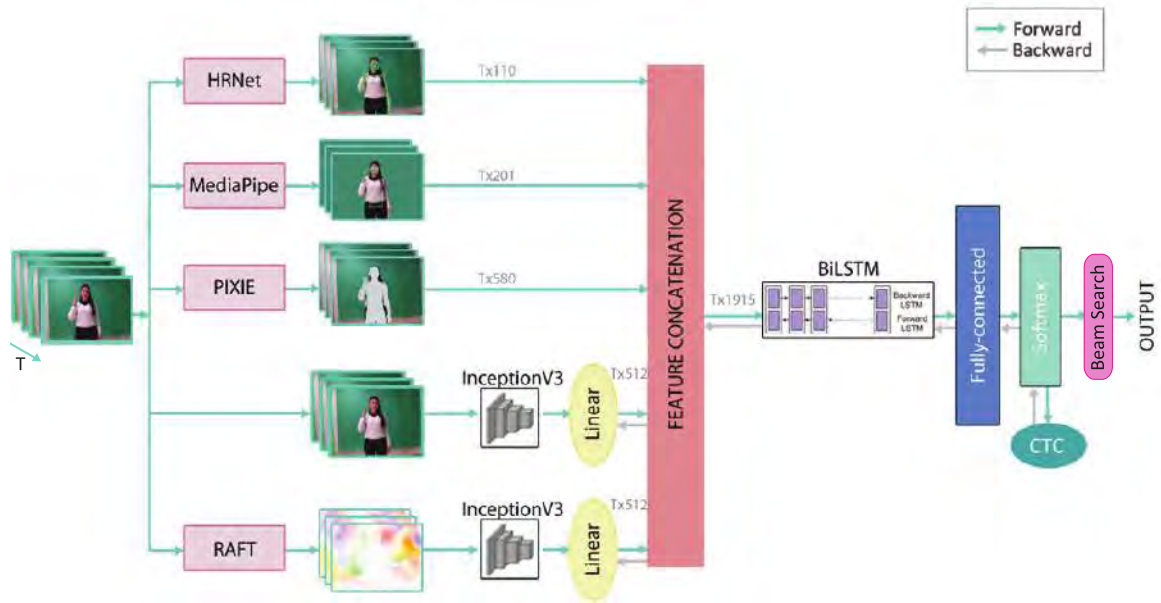


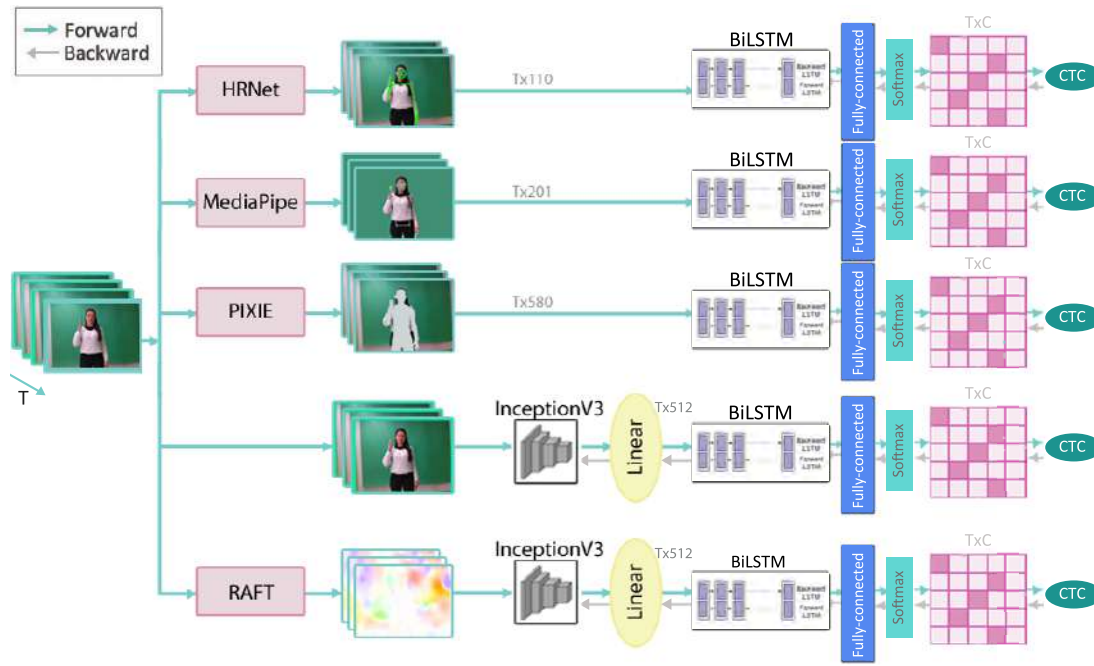
Figure 5.6: Early fusion SLR architecture for multiple (five) visual feature streams within the CTC-based encoder-only framework of Section 5.2.1. Gray backward arrows refer to backpropagation during training.

model are concatenated with 2D and 3D body, face, and hand skeletal joints, as well as with CNN-based appearance and optical flow representations. This process yields 1915-dim feature vectors (311-dim 2D and 201-dim 3D human skeletal data, 580-dim 3D body pose and shape parameters, 512-dim motion-based representations, and 512-dim appearance ones).

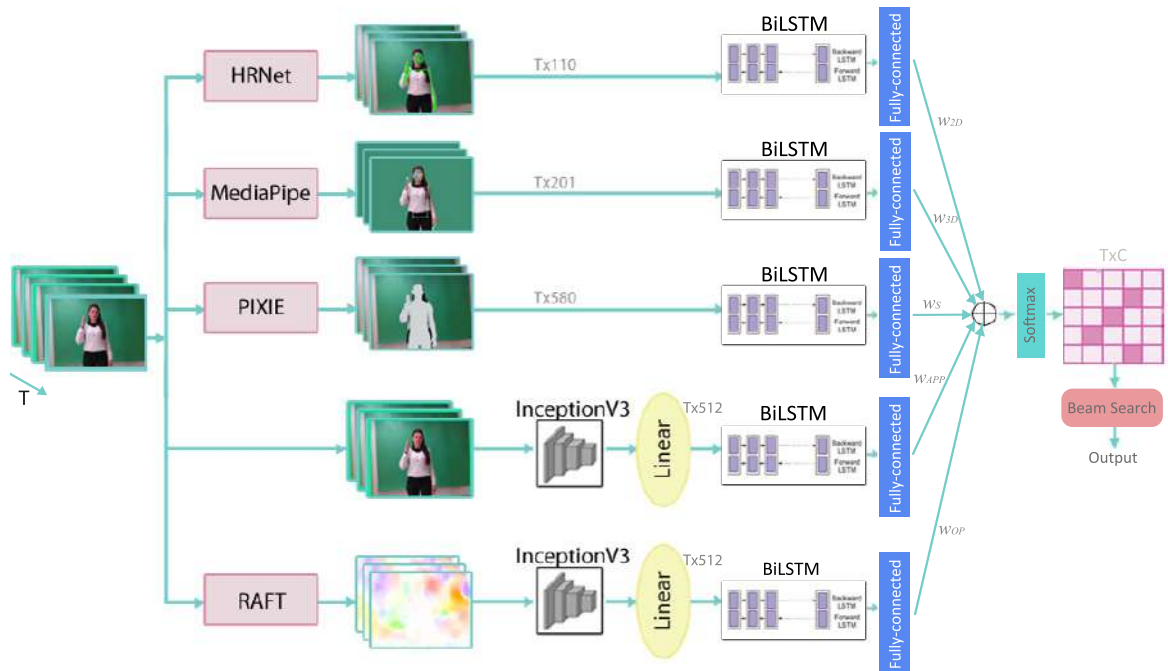
Late fusion: In late fusion, the five different feature representations are first processed separately by their respective sequence learning models (see also Figure 5.7). Subsequently, the fully-connected layer outputs of all five models (before the application of the softmax function) are combined via weighted summation, assigning different weights to each model in accordance with its accuracy on the validation set. The result is passed through a softmax layer, and then decoding is performed.

5.4 Data and Experimental Framework

As already mentioned, the aforementioned single-stream learning models with and without attention are combined with the best performing 2D/3D skeletal, 3D pose and shape, as well as 2D-CNN based appearance and motion feature streams of Chapter 4, developing individual SLR systems. The performance of the resulting systems is assessed on the *ITI continuous GSL dataset* [96] that has been presented in Section 2.2. For evaluation, the pre-defined MS split is employed, with 8,189 videos assigned to training, 1,063 to validation,



(a)



(b)

Figure 5.7: Late fusion SLR architecture for multiple (five) visual feature streams within the CTC-based encoder-only framework of Section 5.2.1, where (a) illustrates the training phase of the late fusion based system, with appearance and optical flow modalities being trained end-to-end, and (b) shows the system during inference with beam search strategy being used for output prediction.

and 1,043 to testing, across all 7 signers.

As already described in Section 5.2, we investigate four single-stream sequence learning models relying on CTC-based decoding and five single-stream sequence learning models within the attention-based decoding framework, with the following implementation details:

- The *LSTM encoder* consists of two layers with 1024 hidden units each. Its training is conducted using the Adam optimizer [141] with the initial learning rate equal to 0.001, decayed by a factor of 0.7 if the validation score remains consistent for 9 evaluation steps.
- The *BiLSTM encoder* consists of three layers with 512 hidden units each. Its training employs the Adam optimizer with an initial learning rate of 0.001 that is decreased by a factor of 0.5 if the validation score remains consistent for 9 evaluation steps.
- The *GRU encoder* consists of two layers with 1024 hidden units each. Its training uses the Adam optimizer with an initial learning rate of 0.001 that is decreased by a factor of 0.3 if the validation score remains consistent for 9 evaluation steps.
- The *attentional LSTM encoder-decoder (ALSTM)* consists of a one-layer encoder and a one-layer decoder, both with 128 hidden units. Training is conducted using the Adam optimizer with initial learning rate of 0.001 decayed by a factor of 0.3.
- The *attentional BiLSTM encoder - LSTM decoder (ABiLSTM)* comprises a one-layer BLSTM encoder and a one-layer LSTM decoder with 128 hidden units. Network training employs the Adam optimizer with an initial learning rate of 0.001 decreased by a factor of 0.3.
- The *attentional GRU encoder-decoder (AGRU)* constitutes a 2-layer GRU encoder-decoder with 128 hidden units. During training the Adam optimizer is used with an initial learning rate of 0.001 decreased by a factor of 0.3.
- The *attentional CNN encoder-decoder (ACNN)* includes 3-layer CNNs on both encoder and decoder with kernel width 5 and 128 hidden units. Training is conducted via the Adam optimizer with an initial learning rate of 0.001 decayed by a factor of 0.1.

All single-stream sequence learning models with CTC decoding are trained employing a dropout rate of 0.1 with the mini-batch size fixed to 16. During inference, the beam search

strategy [158] is used with a beam width of 5. All models are implemented in PyTorch [143], and experiments are carried out using GPU acceleration for both training and evaluation.

In the case of attention-based encoder-decoder models, during training a dropout rate of 0.3 with a mini-batch size fixed to 256 are employed. To achieve a better matching of a target element, the beam search strategy [158] with beam width of 5 during decoding is applied. In addition, label smoothing [142] rate of 0.5 is employed for reducing over-confidence in predictions. In all cases, GPU acceleration is used for both model training and evaluation. In addition, the alignment scores are calculated using the dot function of (5.1), as discussed in Section 5.2.2.

5.5 Experimental Results

We next evaluate the performance of the sequence learning models of Section 5.2 as combined with various visual representations of Chapter 4 and the fusion schemes of Section 5.3, reporting experimental results on the ITI GSL continuous corpus under its MS experimental framework described in Section 5.4. All results are provided in Word (gloss) Error Rate (WER, %), taking into account the number of substitutions, deletions, and insertions of the predicted hypotheses against the annotated ground truth.

In our first experiments, reported in Table 5.1, we employ the best performing visual representations of Chapter 4 individually, namely the HRNet [109] based 2D human skeletal features (110-dim), the 3D human skeleton representation (201-dim) deduced by the MediaPipe [111] holistic pipeline, the 3D human pose and shape parameterization (580-dim) regressed via the PIXIE model [87], the InceptionV3 [126] based appearance representation (512-dim), and the RAFT [135] optical flow features (512-dim). Note that, in order to extract motion feature representations from the optical flow informative images, instead of using the ResNet2+1D model as in Section 4.2.4 we feed them to an InceptionV3 network [126], pre-trained on ImageNet [118], yielding a 2048-dim output, which is further fed to a trainable linear layer, reducing the feature vector size to 512-dim. These streams are fed to the three sequence learning models of Section 5.2.1 that employ different encoders coupled with a SL classifier and CTC decoding for gloss sequence prediction, as well as the four attention-based sequence learning models of Section 5.2.2. As deduced from Table 5.1, in most cases the appearance feature stream yields the best results on all sequence learning models, while

Table 5.1: WER (%) of continuous GSL recognition on the ITI continuous GSL corpus, using various visual feature streams independently in conjunction with the single-stream sequence learning models of Section 5.2.

Visual representations $\rightarrow$ Models $\downarrow$	2D skeleton	3D skeleton	3D pose & shape	Appearance	Flow
LSTM	47.55	47.20	46.05	45.92	56.34
BiLSTM	47.23	43.64	43.51	43.09	45.29
GRU	49.72	46.77	47.58	46.91	47.54
ALSTM	51.47	50.78	49.12	48.30	58.96
ABiLSTM	50.12	47.89	45.74	45.57	49.50
AGRU	52.84	50.04	50.21	50.11	52.18
ACNN	52.08	51.69	51.12	48.94	50.86

incorporating 3D skeleton, as well as 3D pose and shape information seems to also benefit system performance. Further, the 3D human pose and shape derived from PIXIE seems to constitute a more powerful representation than the 2D skeletal features extracted by the HR-Net regression model. Moreover, using 2D skeleton and optical flow representations perform the worst. It can also be observed that the BiLSTM single-stream model without attention turns out superior to the considered alternatives consistently for all feature streams, revealing its suitability in SLR, while the worst results are obtained by the attention-based models. Notably, among all model and single feature stream combinations, the BiLSTM encoder applied on appearance features yields the best results (43.09% WER).

Next, in Table 5.2 we evaluate the combination of the various visual feature streams by means of the two fusion approaches of Section 5.3, namely simple concatenation (early fusion) and weighted summation (late fusion), in conjunction with the best-performing sequence learning model of Table 5.1, i.e. the BiLSTM encoder coupled with CTC decoding. Specifically, we consider all two-stream combinations (ten in total), as well as the fusion of all five streams. Note that, for late fusion, the following equations are used in combining the linear layer outputs of the individual models (denoted by $S_{C_{stream}}$) that are produced by the sequence learning models:

- In the case of two streams, $S_C = 0.5 \times S_{C_1} + 0.5 \times S_{C_2}$
- In the case of all five streams, $S_C = 0.1 \times S_{C_{2D}} + 0.4 \times S_{C_{3D}} + 0.3 \times S_{C_S} + 0.1 \times S_{C_{APP}} + 0.1 \times S_{C_{OF}}$

Table 5.2: WER (%) of continuous GSL recognition on the ITI continuous GSL corpus, using various feature stream combinations under early or late fusion in conjunction with the BiLSTM model with CTC decoding. In the upper ten rows, the first and second column features are fused.

1 st feature stream	2 nd feature stream	Early fusion	Late fusion
2D skeleton	3D skeleton	43.46	42.54
	3D pose & shape param.	43.20	42.48
	Appearance	43.03	42.37
	Flow	45.12	44.52
3D skeleton	3D pose & shape param.	43.48	42.69
	Appearance	43.11	42.74
	Flow	43.29	42.65
3D pose & shape param.	Appearance	42.67	42.14
	Flow	43.16	42.81
Appearance		42.84	42.25
All five feature streams		42.64	41.48

As it can be observed from Table 5.2, in all cases, late fusion is superior to early fusion. In addition, both fusion schemes outperform the corresponding single-stream results, as becomes clear by comparing Table 5.2 results to the BiLSTM row of Table 5.1. Not surprisingly, among the two-stream feature combinations, the one fusing the two best streams of Table 5.1 (appearance and PIXIE parameter features) yields the best results under both fusion schemes (WERs of 42.67% and 42.14% for early and late fusion, respectively). Finally, combining all five streams outperforms all two-stream combinations (as well as all single-stream features). The best result is the one obtained by late fusion of all streams, yielding 41.48% WER. This corresponds to a 1.61% absolute WER reduction over the best result of Table 5.1. Nevertheless, this still lags behind the state-of-the-art result on the ITI continuous GSL corpus that stands at a WER of 36.05%, reported in [159] when using generative adversarial networks.

5.6 Conclusions

In this chapter, we built on our earlier work on visual tracking and feature extraction that we have reported in Chapter 3 and Chapter 4, leveraging the visual representations of Chapter 4 and incorporating them into deep-learning based sequence models. In particular, we developed a number of single-stream sequence learning models relying on RNN (LSTM,

BiLSTM, and GRU) in conjunction with a SL classifier and CTC based decoding, while we also investigated various attentional encoder-decoder architectures counting on RNN and CNN models. The designed sequence models were applied on the considered visual features and evaluated on a large continuous SLR dataset, so as to decide on the best modeling approach. Based on our evaluation, we determined that the BiLSTM encoder coupled with CTC decoding when applied on the appearance features extracted from the InceptionV3 model constitutes the most appropriate architecture, providing the best performance on the continuous GSL dataset considered. Further, we fused the various visual feature streams within the best performing sequence learning framework (BiLSTM with CTC decoding) via an early fusion scheme, as well as a late fusion mechanism. We reported results on the continuous GSL dataset, combining the various visual features via both fusion schemes, obtaining superior results by the late fusion mechanism.

Chapter 6

A Bimodal System for Isolated Sign Language Recognition

6.1 Introduction

As already stated in Chapter 1, an important task in SLR research is this of recognizing isolated signs. The problem is considered simpler than the more challenging task of continuous signing recognition, but nevertheless important in practical applications, thus having gained significant interest in the literature. The majority of introduced ISLR systems rely on 2D and 3D appearance-based feature representations from RGB [14, 38, 160] and/or motion informative data [94, 161], combined with appropriate descriptors of the signer’s skeletal information [99, 162, 163]. For instance, in [56], an ensemble approach makes use of multi-modal information from RGB, optical flow, and depth in addition to skeletal keypoints and features. In [84], G3D, which is a unified ST-GCN, is combined with a 3D-CNN model; in [164], a multi-modal setup relies on 2D human pose landmarks and hand appearance features; and in [33] the role of optical flow, human skeletal features, as well as appearance representations of handshapes and mouthing in conjunction with an SL classifier is investigated. In addition, an inflated 3D-CNN is employed in [96] inferring spatio-temporal feature representations in order to model long-range temporal dependencies, whereas the study conducted in [82] relies on a scheme utilizing skeletal data-based ST-GCNs to effectively capture the dynamic descriptors of SL cues in both spatial and temporal domains.

In this chapter, we focus on addressing ISLR from videos. Specifically, we present our system [58] based on two main modalities: (i) a 3D-CNN model that captures the spatio-

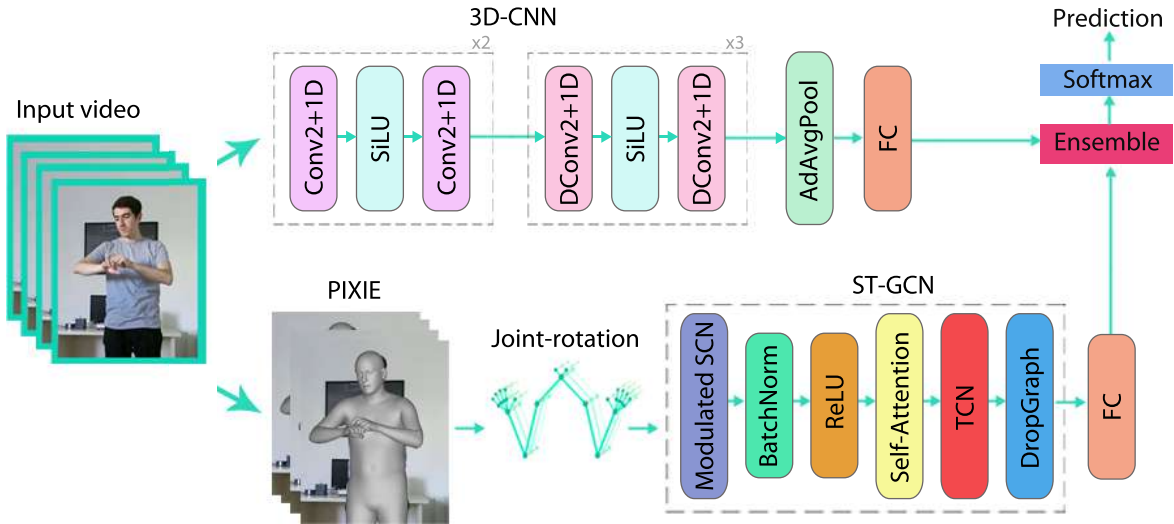


Figure 6.1: Proposed ISLR model architecture: Two groups of streams are fused, namely a deformable ResNet2+1D network applied on the RGB image frames (appearance modality) and a ST-GCN model, which incorporates modulated Spatial GCN (SCN), self-attention module, and Temporal Convolution Network (TCN) that operate on the 3D joint-rotation feature based skeleton graph (skeleton-modality). The following notation is employed: spatial and temporal convolutional block (Conv2+1D), deformable spatial and temporal convolutional block (DConv2+1D), Fully-connected (FC), and adaptive average pooling layer (AdAvgPool).

temporal SL articulation dynamics and (ii) a ST-GCN, which learns the spatial and motion correlation of the human skeletal joints (see also Figure 6.1). The former modules are trained separately, and their results are fused by an ensemble module that leverages the outputs of the modalities last fully-connected layer. The main contribution in this chapter is the investigation of ST-GCNs that rely on modulated spatial GCNs, incorporating weight modulation and an affinity modulation strategy to learn motion patterns that extend beyond the inherent connections of the body's joints. Further, we complement the spatial GCNs with an attention mechanism and temporal convolutions. Regarding graph construction, our system employs the 3D joint-rotation parameterization of the human skeleton. In particular, we utilize the PIXIE 3D human pose and shape regression scheme [87] that has demonstrated exceptional performance in various benchmarks (see also Section 3.3.3). An additional innovation comprises the use of the ResNet2+1D network [85] as the backbone (see also Section 4.2.3), replacing the spatial and temporal convolutions with their deformable counterparts, which are capable of modeling complex geometric transformations and inter-frame motions by augmenting the receptive field of the convolution.

To summarize, the contributions of this chapter encompass the following aspects: (i) a novel version of the ResNet2+1D network, which integrates deformable spatial and temporal convolutions, is introduced; (ii) an innovative ST-GCN unit that integrates modulated GCNs has been devised; and (iii) 3D joint-rotation parameterization, derived by the PIXIE 3D pose and shape regression model, is leveraged for graph construction. Thus far, none of the aforementioned components have been investigated in SLR.

The remainder of the chapter is structured as follows: In Section 6.2, we overview the adopted bimodal ISLR system, including the appearance stream (Section 6.2.1) and the skeleton-based one (Section 6.2.2); In Section 6.3, we provide details regarding the model training data, i.e., two well-known ISLR benchmarks, namely the *AUTSL Turkish SL* [14] and *ITI GSL* [96] corpora, as well as the *SL-ReDu isolated GSL* corpus [29]. We also describe in detail the adopted experimental framework; In Section 6.4, the presented approach is evaluated on the aforementioned datasets, conducting in-depth ablations to emphasize the innovative aspects of our approach; and the chapter is concluded in Section 6.5.

6.2 The Isolated SLR System

In this section, we present our deep-learning system for isolated SLR, which makes use of multi-modal appearance- and skeleton-based information. In particular, we next introduce an overview of the system, first discussing its overall architecture, including the modified ResNet2+1D network that captures spatio-temporal appearance information of signing, as well as the ST-GCN that integrates the modulated GCN variant. Further, we provide a detailed presentation of the scheme deployed for appearance- and skeleton-based stream modality fusion. More details are provided next.

6.2.1 Appearance-Based Modality

The visual representation of the different articulators involved in signing is crucial in SLR due to their high spatio-temporal correlation and the unique information content each articulatory entails. To obtain spatio-temporal representation learning, a 3D-CNN model, which decouples the spatial and temporal convolutions of 3D-CNNs, is applied to the input signing video. Specifically, we adopt the 18-layer ResNet2+1D network [85] as the underlying architecture, substituting its spatial and temporal convolutions with their deformable counterparts.

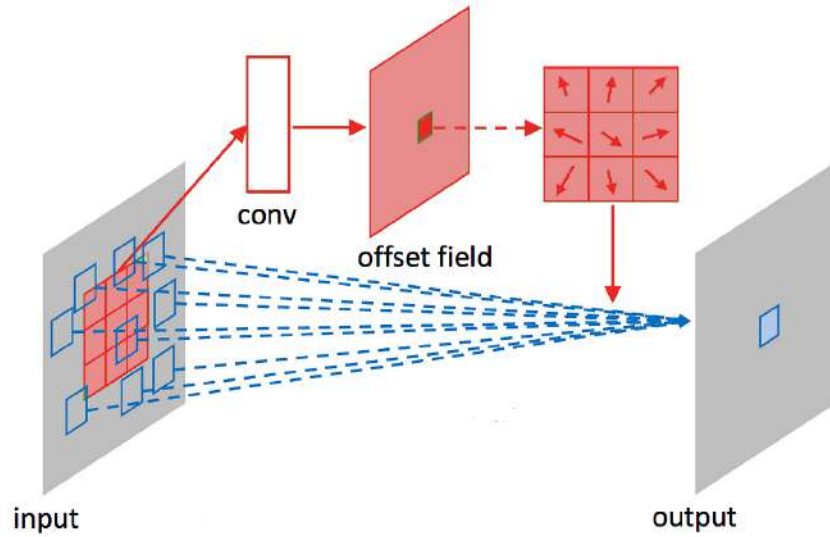


Figure 6.2: Deformable convolution with extended sampling grid (figure adapted from [165]).

Given the success of deformable CNNs in accurately representing geometric variations on challenging benchmarks in multiple domains [165], we utilize them for modeling human-related complex geometric transformations and inter-frame motions.

In ResNet2+1D, the 3D convolutional kernels with dimensions $u \times K \times K$ are substituted by spatial convolutional filters of size $1 \times K \times K$ followed by temporal convolution filters of size $u \times 1 \times 1$. Here, rather than applying traditional convolutions that operate on a fixed sampling grid, we attach learned offsets to the grid of traditional convolutional kernels, therefore expanding the convolution receptive field. More precisely, we incorporate deformable convolutions that function in two stages: (i) a traditional convolutional layer is employed to predict the positional offsets $\Delta \mathbf{p}_m$, where $m = 1, \dots, |M|$ with M representing the sampling grid of a traditional convolution, and (ii) the sampling grid is expanded by incorporating the predicted offsets $\Delta \mathbf{p}_m$ into the regular convolution operation (see also Figure 6.2).

Despite deformable convolution robustness, their computational cost is higher compared to traditional convolutions. To rectify this, we strategically integrate them in the last three layers (five layers in total), rather than utilizing them over the whole network. It is important to mention that we conducted experiments incorporating them in all layers, as well as at various layers of the ResNet2+1D model. Our findings indicate that utilizing them in the last three layers of the network yields the best SLR results.

To enhance the model's capacity, we employ the Sigmoid Linear Unit (SiLU) [166] activation function, a non-monotonic smooth function that is computed by the multiplication

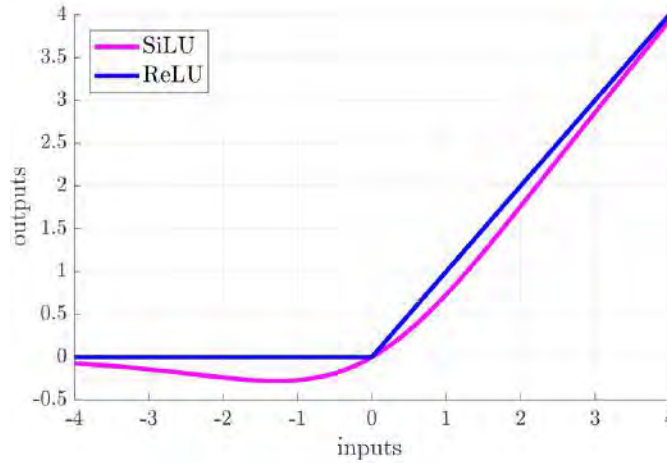


Figure 6.3: The SiLU and ReLU activation functions (figure adapted from [166]).

of the input with its sigmoid function. For inputs of large magnitude, the SiLU activation approximates well the ReLU one. As shown in Figure 6.3, unlike ReLU, the SiLU activation does not increase monotonically, having instead a global minimum value (-0.28 for input equal to -1.28) where the derivative is zero that serves as an implicit regularizer inhibiting the learning of weights with large magnitudes.

6.2.2 Skeleton-Based Modality

Here, we present an ST-GCN unit based on a modulated GCN in conjunction with a temporal convolution complemented with a self-attention mechanism. Further, the 3D joint-rotation parameterization of the human skeleton, generated by the PIXIE model, is employed for graph construction. Details are provided next.

Graph Construction: GCNs constitute a generalized variant of CNNs, with filters operating on graph-structured data, where nodes correspond to the human body joints. More precisely, a graph is defined as $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ denotes the set of nodes corresponding to J human skeletal joints and $\mathcal{E}$ represents the intra-skeleton structure edges. The edges can be represented by an adjacent matrix, denoted as $\mathbf{A} \in \{0, 1\}^{J \times J}$, where 1 regards a direct link between a pair of joints, while 0 indicates no direct connection. Each joint i is associated with a D -dimensional feature vector $\mathbf{q}_i \in \mathbb{R}^D$. Therefore, the matrix $\mathbf{Q} \in \mathbb{R}^{D \times J}$ represents the aggregation of features from all graph nodes.

Building upon our work in [80], we adopt the 3D joint-rotation parameterization of the human pose as a graph feature representation. For that purpose, in [80] we deployed the Ex-

Pose 3D human reconstruction model [112], which utilizes individual networks to infer body, face, and hand parameters. However, such regression models depend exclusively on the predictions of the distinct parts, where the respective predictions of hands and face networks can



Figure 6.4: 3D body reconstruction on sample frames of the evaluated datasets in this chapter. 1st column: Frames from the AUTSL dataset [14] (1st row), the ITI GSL database [96] (2nd row), and the SL-ReDu isolated GSL corpus [29] (3rd row); 2nd column: 3D body reconstruction via the ExPose regression model [112]; 3rd column: 3D human body reconstruction by the PIXIE estimator [87].

be inaccurate in many cases (see also Figure 6.4 (2nd column)). To mitigate this, we deploy instead a recently proposed method, the PIXIE [87] 3D pose and shape regression model, which utilizes a moderator to estimate the 3D body and shape parameterization (see also Figure 6.4 (3rd column)). As discussed in Section 3.3.3, PIXIE returns shape and expression representations, described by a total of 250 coefficients, as well as whole-body pose, including 55 joints with 6 degrees of freedom, i.e., 25 joints for body posture encompassing head, jaw, and neck, as well as 15 joints for each hand. As it can be deduced from Figure 6.4 (see also Figure 3.8), ExPose struggles with precise hand shapes, while PIXIE outperforms ExPose with its facial expressions, as well as hand/body shapes and orientations being more realistic.

ST-GCN Unit: Given input $\mathbf{Q}_{in}$, the forward propagation rule of a spatial GCN layer is given by:

$$\mathbf{Q}_{out} = \sigma(\mathbf{W}\mathbf{Q}_{in}\hat{\mathbf{A}}), \quad (6.1)$$

with $\mathbf{Q}_{out} \in \mathbb{R}^{D' \times J}$ being the updated feature vector, $\sigma()$ denoting an activation function (e.g., ReLU), $\mathbf{W} \in \mathbb{R}^{D' \times D}$ representing a learnable weight matrix, and $\hat{\mathbf{A}}$ being the symmetrically normalized affinity matrix. The ST-GCN unit consists of a GCN, which transforms and aggregates the features of the nodes and their neighboring nodes, coupled with a temporal convolution, which operates on the temporal neighborhood of nodes across consecutive frames.

One limitation concerning the vanilla graph convolution function is that the feature transformation matrix $\mathbf{W}$ is shared among all nodes in the graph, which hinders the modeling of relations between different body joints that do not belong to the same neighbor. To address this issue, many recent works apply decoupled graph convolutions, where an independent learnable weight matrix is assigned to each node for feature transformation [167]. Despite the improved performance accomplished by decoupling the weights, the size of the model increases significantly. Here, we employ a variant of graph convolution, the so-called modulated GCN [86] that is composed of two basic components: (i) weight modulation; and (ii) affinity modulation. In the case of weight modulation a learnable weight modulation matrix $\mathbf{L} \in \mathbb{R}^{D' \times J}$, which is unique for each node i , is integrated into the graph convolution function to modulate the shared weight matrix. Thus, (6.1) is modified to:

$$\mathbf{Q}_{out} = \sigma((\mathbf{L} \odot (\mathbf{W}\mathbf{Q}_{in}))\hat{\mathbf{A}}),$$

where $\odot$ denotes element-wise multiplication. In regular GCNs, the adjacent matrix $\mathbf{A}$ is

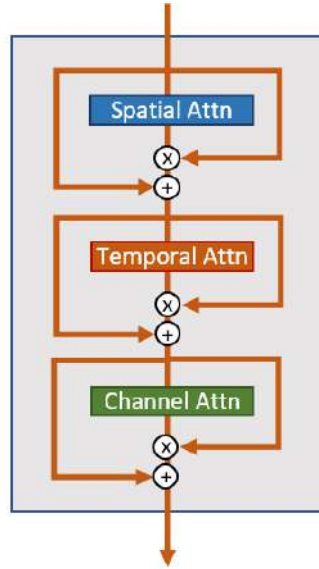


Figure 6.5: Self-attention module with spatial, temporal, and channel attention integrated into the ST-GCN unit (figure from [56]).

usually predefined according to the structure of the human skeleton, albeit SL includes motion patterns beyond the physical human skeleton structure. The affinity modulation technique addresses this limitation, by adding a learnable mask $\mathbf{B} \in \mathbb{R}^{J \times J}$ to the matrix $\mathbf{A}$, i.e. $\mathbf{A}' = \mathbf{A} + \mathbf{B}$. To prevent overfitting, we apply symmetry regularization to the affinity modulation, generating a symmetric affinity matrix as introduced in [86].

GCN is followed by self-attention with a spatial, a temporal, and a channel attention module, all three of which are cascaded to improve the context representation of the nodes (see also Figure 6.5). In addition, a temporal convolution is used to learn the relational patterns between successive frames. The dropout layer does not improve GCN performance, thus we add a DropGraph module [167] to avoid overfitting, where one node is dropped along with its neighboring nodes. In our developed ISLR model, ten such ST-GCN units are used, followed by a global average pooling layer for both spatial and temporal domains before the fully-connected layer.

6.2.3 Multi-Modal Fusion

To fuse the two modalities, i.e, the appearance and the skeleton one, we utilize an ensemble module, combining the posteriors obtained from the last fully-connected layer of the deformable ResNet2+1D and the modulated ST-GCN based sequence learning models in a

appropriate manner. In more detail, we assign proper weights to each modality output regarding their individual performance and sum them up, producing the final probability scores.

In particular, the two modalities are trained for 100 epochs, with the checkpoint of each modality with the lowest validation error being stored. Subsequently, the output of the fully-connected layers before the softmax layer are weighted and summed as follows:

$$\mathbf{p}_{\text{fused}} = 1.0 \mathbf{p}_{\text{app}} + 0.9 \mathbf{p}_{\text{skel}}.$$

Then, the output $\mathbf{p}_{\text{fused}}$ is passed through a softmax layer. The final predicted class is estimated using the mean value of the probabilities of all classes among all frames and, then the $\text{argmax}()$ operator is used estimating the class with the highest probability.

6.3 Experimental Framework

The performance of the proposed multi-modal system described in Section 6.2 is evaluated on three publicly available isolated SLR corpora that were described in detail in Section 2.2, i.e., the *AUTSL dataset*, the *ITI GSL isolated database*, and our collected studio *SL-ReDu isolated GSL corpus*. Specifically: (i) For our experiments on the AUTSL dataset, we follow the official SI data split, which consists of 28,142 training videos from 31 signers, 4,418 validation videos from 5 signers, and 3,742 test videos from 7 signers; (ii) For our experiments on the ITI GSL dataset, we employ an SI paradigm, based on a 7-fold cross-validation approach, where video data are divided into 7 subsets, each containing videos of a single signer. For each of the 7 folds, one subset is assigned to testing, and the remaining 6 constitute the training set (split between development and validation at an 80-20% ratio); and (iii) Finally, for the SL-ReDu Isolated GSL corpus we consider both MS and SI experimental frameworks, adopting the official splits for each case. In the MS case, data from all signers are split between training (12,820 videos), validation (2,263 videos), and testing (7,549 videos) with a single fold used, while in the SI case a 7-fold cross-validation framework is adopted, with each fold containing training and validation data from 18 signers, whereas testing is performed on the remaining 3. The process repeats over all 7 folds to cover all signers.

Concerning system implementation, for the 3D-CNN stream (appearance modality), we first crop the torso relying on the 3D joints provided by the MediaPipe framework [111]. Subsequently, we scale this region to 256×256 . We then use ResNet2+1D-18 with weights trained on the Kinetics dataset as the underlying model. To further improve SLR performance,

we pretrain our model on the Chinese dataset SL [162]. During fine-tuning, we apply the Adam optimizer with an initial learning rate of 0.0001 and weight decay of 0.0001. We also use the cross-entropy loss function with label smoothing [142] and a mini-batch size fixed to 16. Concerning the skeletal stream, for graph construction we employ a 46-node skeleton graph with 6-dimensional skeleton features, corresponding to the dimension of the joint-rotation representation estimated by PIXIE. Note that we neglect 9 skeletal joints out of 55 regressed joints that correspond to the invisible lower parts of the human body, retaining only 16 body pose points. The initial learning rate is set at 0.1, while the weight decay is delineated at 0.001. The Adam optimization algorithm is employed, as well as a batch size of 16.

6.4 Experimental Results

We now proceed with our isolated SLR experiments. Specifically, we first provide in-depth ablations on the system components, and finally we compare our approach to existing state-of-the-art SLR methods. The performance of our isolated SLR model is assessed on all three datasets of Section 6.3 in terms of sign recognition accuracy (%).

Initially, we present a comparison between the introduced deformable 3D-CNN model and several state-of-the-art 3D-CNN variants, as shown in Table 6.1. The accuracy of our model on AUTSL, ITI GSL, and SL-ReDu IGSL (SI setting) is 95.39%, 97.12%, and 85.95%, respectively. In all cases, our model surpasses all alternative 3D-CNN models considered. Subsequently, we conduct ablations on the various elements of the introduced modulated ST-GCN unit within the investigated skeleton-based sequence learning modality of Section 6.2.2,

Table 6.1: Sign recognition accuracy (%) in all three datasets using various 3D-CNNs on the appearance stream alone.

CNN Models	AUTSL	ITI GSL	SL-ReDu IGSL (SI)
C3D [125]	81.95	85.66	80.54
I3D [96]	87.64	89.11	80.82
P3D [138]	90.57	92.14	81.21
R3D [130]	92.04	94.03	81.97
ResNet2+1D + ReLU [168]	93.26	95.89	82.07
ResNet2+1D + SiLU	93.85	95.98	82.82
ResNet2+1D (pre-trained) + SiLU [56]	94.77	96.51	84.77
Ours	95.39	97.12	85.95

Table 6.2: Sign recognition accuracy (%) in all three datasets using ST-GCN variants on the skeletal stream only.

ST-GCN Variations	AUTSL	ITI GSL	SL-ReDu IGSL (SI)
w/o Attention	93.88	94.85	82.67
w/o Modulated GCN	94.59	95.04	83.22
w/o DropGraph	95.12	95.79	83.64
w Decouple GCN	95.17	95.84	84.44
Ours	95.32	96.14	84.73

Table 6.3: Sign recognition accuracy (%) on all three datasets using various streams for the skeleton graph construction (skeletal stream alone).

Streams	AUTSL	ITI GSL	SL-ReDu IGSL (SI)
2D Joint-position	94.96	95.46	83.07
3D Joint-position	95.10	95.68	83.37
2D Joint-motion	92.54	93.11	82.32
3D Joint-motion	93.24	93.57	82.71
Joint-rotation (ExPose)	95.15	95.74	83.12
Joint-rotation (PIXIE)	95.32	96.14	84.73

as shown in Table 6.2. Clearly, the modulated GCN and the attention mechanism are the most significant contributors. Further, the modulated GCN seems to constitute a more powerful variant than the decoupled one, since substituting the modulated GCN with the latter in our system yields lower accuracy. It can also be observed that the DropGraph module incorporation also contributes to better recognition accuracy.

Further, in Table 6.3, we assess the performance of the ST-GCN using various skeletal feature representations for graph construction. It should be noted that the 2D skeleton is obtained using the HRNet 2D pose detector [109], while the 3D skeleton is derived from the MediaPipe holistic model [111]. For comparison purposes, we retain 9 body and 30 hand (15 for each hand) points of the 133 skeletal joints returned by the HRNet, resulting in 39 2D skeletal joints in total, while for the 3D skeleton we discard 24 body, 12 hand, and 468 facial landmarks retrieved by MediaPipe, yielding 39 skeletal joints. In the case of the ExPose-based parameterization [112], we retain 16 body skeletal joints and 15 joints for each hand (53 joints retrieved via ExPose in total), resulting in 46 skeletal joints. In the joint-position case, each joint i at frame t is represented by its 2D or 3D skeletal joint-positions, i.e., $q_{it}^{(JP)} = (x_{it}^{(2D)}, y_{it}^{(2D)})$ in

Table 6.4: Ours vs. literature results on both datasets. Notation: appearance (Full Frame), hand appearance (Hand), skeleton (Skel.), optical flow (Flow).

Dataset	Model	Modalities	Acc. (%) $\uparrow$
AUTSL	VTN-PF [164]	Full Frame & Hand & Skel.	92.92
	MS-G3D [84]	Full Frame & Skel.	96.15
	Ours	Full Frame & Skel.	96.67
	SAM-SL [56]	Full Frame & Skel. & Flow	98.42
ITI GSL	I3D + BiLSTM [96]	Full Frame	89.74
	OpenHands [169]	Skel.	95.40
	Ours	Full Frame & Skel.	97.85
SL-ReDu IGSL (SI)	Ours	Full Frame & Skel.	86.78
SL-ReDu IGSL (MS)	Ours	Full Frame & Skel.	96.25

2D and $q_{it}^{(JP)} = (x_{it}^{(3D)}, y_{it}^{(3D)}, z_{it}^{(3D)})$ in 3D, while in the joint-motion instance each node is represented by its 2D and 3D joint-motion vector $q_{it}^{(JM)} = q_{it+1}^{(JP)} - q_{it}^{(JP)}$. As it can be deduced from the table, the joint-rotation parameterization obtained by the PIXIE framework clearly reaches the highest recognition accuracies.

Finally, in Table 6.4, we compare our model performance to the literature, when both modalities are taken into account and their outputs are fused in Table 6.4. Our model achieves 97.85% sign classification accuracy in the case of ITI GSL, surpassing the state-of-the-art by 2.45% absolute, which translates to a 53% relative error reduction. In addition, our model achieves 96.67% accuracy on the AUTSL dataset, trailing the state-of-the-art result (98.42%) reported in [56]. Note however that in our work we only focus on the RGB stream, while [56] incorporates several modalities at the expense of increased complexity and computational cost, such as appearance, optical flow, skeleton, and skeletal representations. Notably, we can achieve a slightly better result than [56], namely 98.45%, if we further include optical flow (motion) modality in our ensemble module, based on the RAFT model [135]. Finally, on the SL-ReDu IGSL corpus we obtain 86.78% accuracy under the SI experimental paradigm and 96.25% in the MS case. This constitutes the first experimental report on the SL-ReDu IGSL dataset.

Revisiting the tabulated SI results, one can observe that the 3D-CNN appearance module performs better than the skeletal ST-GCN one (AUTSL: 95.39% vs. 95.32%, ITI GSL: 97.12% vs. 96.14%, SL-ReDu IGSL: 82.43% vs. 82.17%), by comparing the “Ours” entries of Table 6.1 (appearance stream alone) to the corresponding entries of Tables 6.2 or 6.3 (skeletal stream only). Nevertheless, both modules perform well, with their combination fur-

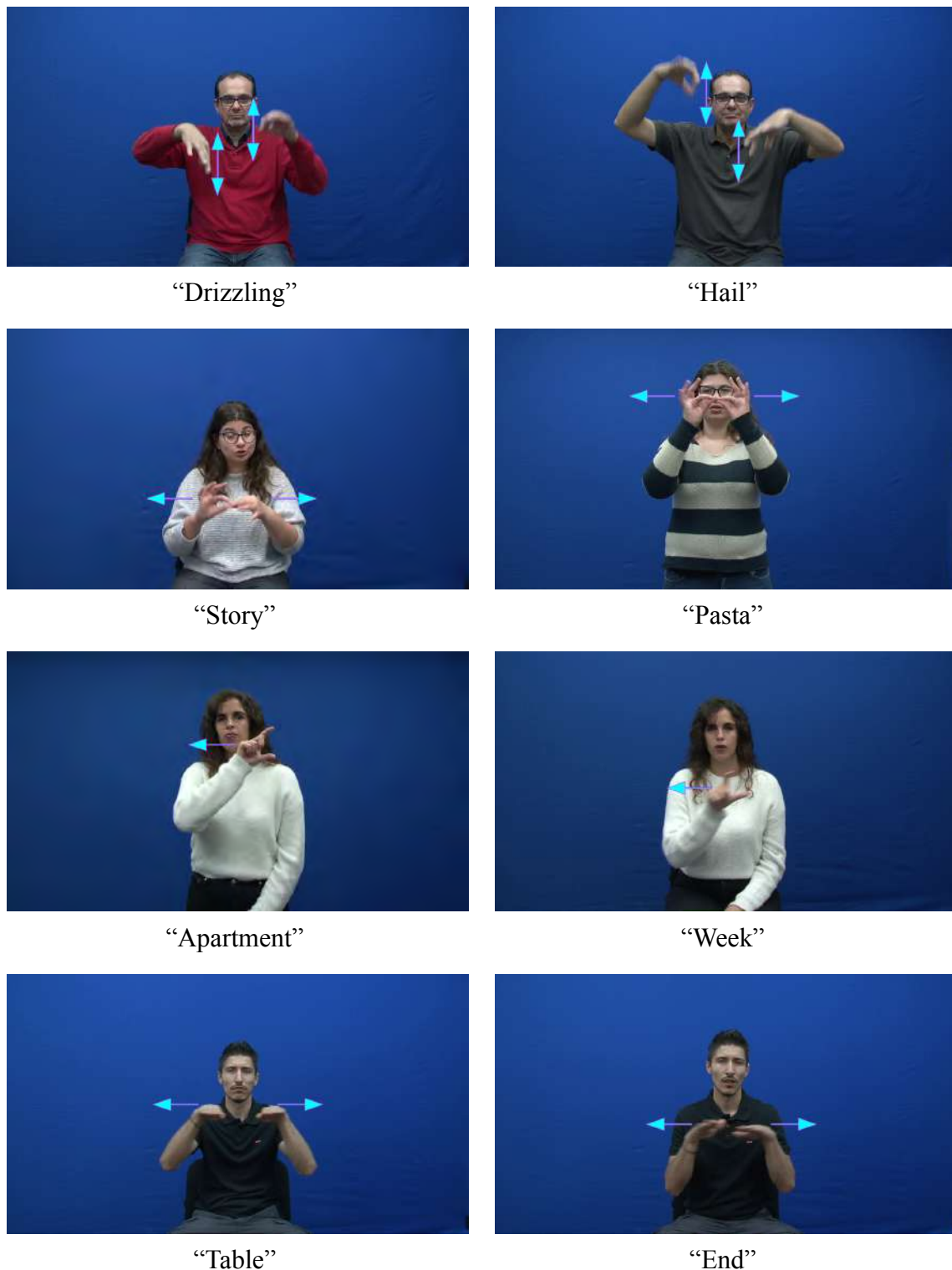


Figure 6.6: Examples of signs of the SL-ReDu isolated corpus that are frequently confused by our model due to their similar handshapes and hand orientation. Each row shows frames of different signs with almost identical signing.

ther enhancing performance (Table 6.4, “Ours”), with a relative error reduction of 28% on AUTSL, 25% on ITI GSL, and 42% on SL-ReDu IGSL (over appearance only).

It should be noted that the introduced model outperforms our earlier work on the isolated SLR task presented in [168] (Table 6.1: ResNet2+1D +ReLU). In particular, our model exhibits sign classification error relative improvement of 51% (3.41% absolute), 48% (1.96% absolute), and 19% (3.35% absolute) on the AUTSL, the ITI GSL, and the SL-ReDu IGSL (SI) datasets, respectively. It is worth noting that among the three datasets considered, the SL-ReDu corpus is the most challenging due to its larger sign vocabulary (369 vs. 226 of AUTSL and 310 of ITI-GSL), but also due to the very similar articulation of some of its signs, as also depicted in Figure 6.6.

6.5 Conclusions

This chapter primarily focused on the difficult task of SI isolated SLR. We explored the impact of combining two distinct modalities that operate on visual representations of appearance and human pose to capture the signing activity. Specifically, we investigated the benefit of incorporating spatio-temporal representations into the SLR model derived from the ResNet2+1D 3D-CNN model, albeit augmenting the convolution receptive field by substituting its spatial and temporal convolutions with their deformable counterparts. To further enhance our model we employed the SiLU activation function that outperformed the traditional ReLU. In addition, we presented an innovative ST-GCN model relying on modulated GCNs to integrate both weight and affinity modulation for modeling diverse correlations between different body joints beyond the physical human skeleton structure. The modulated GCNs were followed by an attention mechanism incorporating a spatial, a temporal, and a channel attention module connected in cascade, as well as temporal convolutions to accurately learn both local and global dynamics of the human skeletal joints. For graph construction we investigated the benefit of employing 3D human joint-rotation parameterization that was estimated through the PIXIE 3D pose and shape reconstruction approach. Finally, we fused both modalities into the proposed system, surpassing the current state-of-the-art on the ITI GSL corpus and reaching competitive performance on the popular AUTSL dataset, while also reporting the first ever results on our collected SL-ReDu isolated GSL corpus under both MS and SI experimental paradigms.

Chapter 7

A Bimodal System for Continuous Sign Language Recognition

7.1 Introduction

This chapter focuses on the task of continuous SLR, which has received considerable attention lately. The objective of continuous SLR is to predict a sequence of SL glosses from a signing video, without any prior knowledge concerning gloss-level segmentation [35,36,38].

The sequential module employed for gloss sequence prediction constitutes a crucial element in CSLR research. For that purpose, most recent schemes in the literature primarily rely on RNNs, typically the LSTM networks [77] or BiLSTM [78], in conjunction with CTC-based alignment [81]. For example, [170] tackles this challenging problem by applying 2D-CNN as the visual module followed by a BiLSTM encoder for spatio-temporal feature extraction and the CTC loss function for alignment. In [67], a 3D-CNN is employed for spatio-temporal visual feature learning, and a stacked dilated convolutional network coupled with CTC decoding is used for sequence modeling. Further, several proposed schemes use the Transformer encoder and CTC alignment as the sequence learner for CSLR, such as the approaches in [9, 63]. Moreover, some approaches consider CSLR as a task of translating videos into text representations, deploying attention-based encoder-decoders [68, 71]. In recent introduced schemes, there has been a notable emphasis on integrating temporal convolutional networks (TCNs), known for their ability to capture temporal features, with RNN encoders [35, 64, 65] or Transformers [57, 171, 172].

Although Transformers have shown impressive performance in capturing long-term de-

dependencies, they tend to neglect the local structures that are present in SL sequences. To tackle this issue, the work in [57] presents an architecture that relies on a local Transformer, incorporated with relative position encoding [90] and localness modeling via a Gaussian bias [89]. Inspired by the above and the work in [88], where an RNN is combined with a Transformer encoder, in this chapter we present our CSLR model [66] that relies on a window-based RNN (LSTM) module, followed by a Transformer encoder that incorporates multi-head local self-attention. In more detail, the RNN module operates on small segments of the input sequence, producing local contextual latent representations that are then fed as input into the Transformer, which is equipped with local self-attention arising through relative position encoding and Gaussian bias. This enables the Transformer to effectively capture short-term dependencies within the visual representations. Further, the Transformer adopts a multi-head attention mechanism to facilitate the acquisition of global structures.

In terms of visual feature streams, we once again leverage the potential complementarity of different modalities, introducing a bimodal framework that incorporates a CSLR sequence learning model that operates on two distinct streams, i.e., RGB and optical flow-based motion informative frames. To align the two streams and combine their scores, we employ the CTC guiding technique introduced in [91].

The alignment module employed in CSLR plays a crucial role in aligning the representations derived from the sequence learning model with the gloss sequence. Although the majority of literature efforts employ CTC-based alignment [35, 67, 170], relying exclusively on the CTC loss function may lead to training issues, as the training features may not be adequate to achieve accurate recognition results. To address this problem, the works in [39, 64, 67] utilize stage optimization techniques to enhance the extracted features, while the work presented in [57] incorporates auxiliary learning to improve the consistency of CSLR backbones. Here, motivated by the former we devise an alignment module that combines the CTC loss with a knowledge distillation loss [92] that aligns the probability distributions generated by the sequence learning model and the RNN module, enhancing the representation capacity of both the features derived from the visual and sequential modules.

In summary, the main contributions of our proposed architecture are: (i) a novel sequence learning model that integrates a window-based RNN module and a Transformer encoder, which is further improved by incorporating relative position encoding and a Gaussian bias to enhance localness modeling; (ii) a multi-modal framework that applies the aforementioned

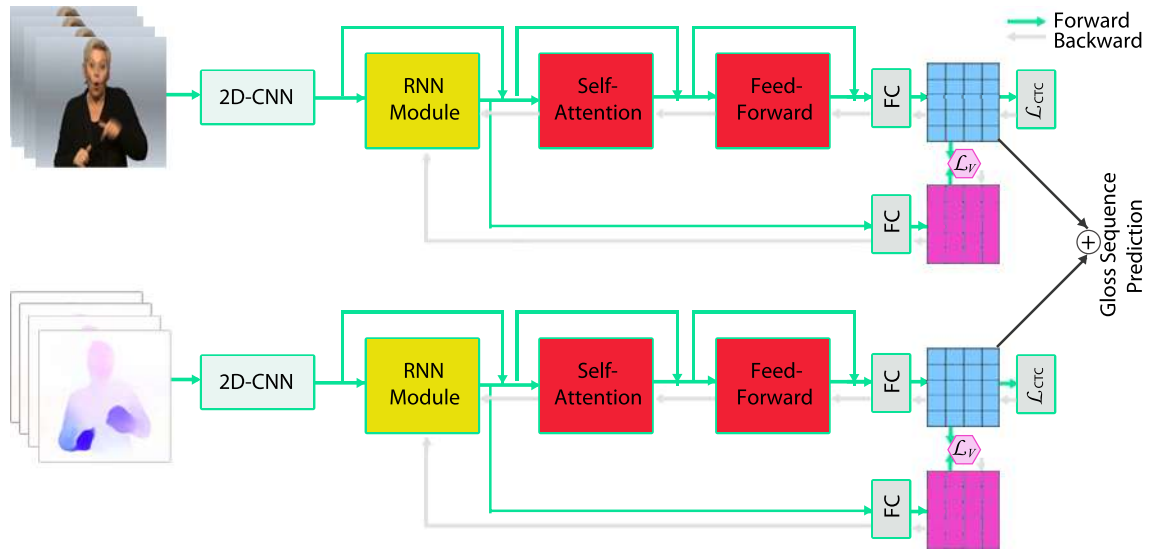


Figure 7.1: An overview of the proposed bimodal CSLR model that generates a sequence of glosses via an RNN module and a Transformer encoder.

sequence learning model to both RGB and optical flow streams which are subsequently combined using CTC guiding alignment; and (iii) the combination of the CTC loss with a visual alignment loss.

The remainder of the chapter is structured as follows: In Section 7.2, we describe in detail the adopted continuous SLR system, including the visual module (Section 7.2.1), the sequence learning approach (Section 7.2.2), the alignment module (Section 7.2.3), and the ensemble module 7.2.4; In Section 7.3, we provide details regarding the model training data, namely the *RWTH-PHOENIX Weather 2014 corpus* [13], the *RWTH-PHOENIX Weather 2014T dataset* [9], and our *SL-ReDu continuous GSL corpus* [29], as well as the adopted experimental framework; In Section 7.4, the presented approach is evaluated on the aforementioned datasets, demonstrating superior performance; and the chapter is concluded in Section 7.5.

7.2 The Continuous SLR system

Our deep-learning based approach, as illustrated in Figure 7.1, learns from two distinct modalities: RGB and optical flow. The baseline CSLR model applied to both streams comprises three components: (i) a visual module that regresses local context visual features through a 2D-CNN image feature learner and a window-based RNN module; (ii) a sequence model

that employs a Transformer encoder with multi-head local self-attention; and (iii) an alignment module that combines the CTC and knowledge distillation loss functions. In order to combine the RGB and optical flow modalities, a guiding CTC method is utilized.

7.2.1 Visual Module

To enhance the visual module we leverage both full-frame RGB and optical flow streams, capturing spatial and motion information of the SL articulation. For generating the motion informative images, we utilize the SpyNet optical flow model [132]. Motivated by its notable visual representation learning ability in the CSLR task [57, 65], we take into consideration visual representations derived from the VGG11 network [173] for both appearance and optical flow features. Specifically, a sequence of frames with length T , $\mathbf{x} = (\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_T) \in \mathbb{R}^{T \times 3 \times R \times W}$, where 3 are the number of channels for both RGB and optical flow frames, while R and W represent the height and the width of frames, is properly resized (256×256) and cropped to 224×224 , before being fed to the 2D-CNN model. The VGG11 architecture, which is pre-trained on the ImageNet corpus [118], follows a global average pooling layer, resulting in 512-dimensional features, $\mathbf{z} = (\mathbf{z}_1, \mathbf{z}_2, \dots, \mathbf{z}_T) \in \mathbb{R}^{T \times 512}$.

Before feeding spatial features into the sequence learning module, we enhance the corresponding visual representations of each frame by integrating the sequential and local information from its neighborhood. For that purpose, the visual representations deduced by the 2D-CNN image feature learner are passed through a window-based RNN module, capturing the local short-term relations between frames in the sequence. The former module operates on local windows of frames generating latent representations for each of them. Specifically, inspired by the work in [88], we deviate from the conventional approach of applying RNNs to the entire sequence, rearranging the initial frame feature sequence into multiple shorter sub-sequences through a local window of fixed size S , with each sub-sequence corresponding to one target frame. Specifically, each sub-sequence involves S consecutive frames with the last being the target frame. To obtain this, we pre-pad the beginning of the input sequence by $S - 1$. Figure 7.2 illustrates the window-based RNN module operating on a frame feature sequence of length T via a local window of size $S = 3$, generating the latent representations. More precisely, the local sequence $\mathbf{z}_t = (\mathbf{z}_{t-S+1}, \mathbf{z}_{t-S+2}, \dots, \mathbf{z}_t)$ with a window size of S is fed into the RNN unit, yielding the hidden state representation $\mathbf{h}_t$. It should be noted that the RNN module is based on the BiLSTM encoder [77] that is coupled with a normalization

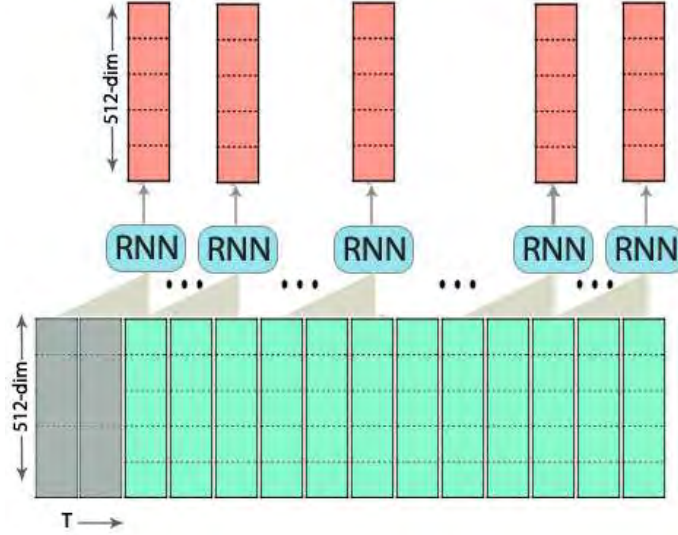


Figure 7.2: Illustration of the window-based RNN module ($S=3$).

layer.

The RNN module is analogous to a 1D-CNN layer, where each local window is processed by convolution operations. However, the convolution operation neglects the temporal data of the elements within the local window.

7.2.2 Sequence Learning

A gloss in a signing video only lasts a few frames, highlighting the significance of local contexts. For sequence learning, we thus use a Transformer encoder integrated with both global and local structural modeling. In particular, for capturing the global long-term dependencies, a multi-head attention layer coupled with a feed-forward layer is employed (see also Figure 7.1). The two layers are imposed on normalization. More precisely, three distinct linear layers are applied to the sequential representation $\mathbf{h} \in \mathbb{R}^{T \times 512}$ derived from the window-based RNN module, producing the queries $\mathbf{Q}_{tr} \in \mathbb{R}^{T \times 512}$, the keys $\mathbf{K}_{tr} \in \mathbb{R}^{T \times 512}$, and the values $\mathbf{V}_{tr} \in \mathbb{R}^{T \times 512}$. As already mentioned, multi-head attention is utilized, with $\mathbf{Q}_{tr}$, $\mathbf{K}_{tr}$, and $\mathbf{V}_{tr}$ being divided yielding $\mathbf{Q}_{tr}^\alpha, \mathbf{K}_{tr}^\alpha, \mathbf{V}_{tr}^\alpha \in \mathbb{R}^{T \times 512/n_\alpha}$, where $\alpha = 1, \dots, n_\alpha$ and n_α is the number of heads. Subsequently, the splits pass separately through a self-attention layer to produce the attention scores, which are then combined to obtain the final output. Each head attention score is calculated by the equation:

$$\text{attn}^\alpha = \frac{\mathbf{Q}_{tr}^\alpha (\mathbf{K}_{tr}^\alpha)'}{\sqrt{512/n_\alpha}} \in \mathbb{R}^{n_\alpha \times T \times T}.$$

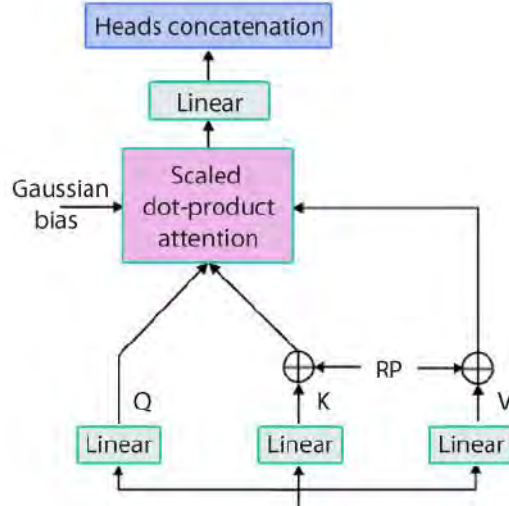


Figure 7.3: Self-attention enhancement with local modeling via Gaussian bias and relative position encoding.

We augment the self-attention layer with localness modeling as local context dependencies are crucial to the CSLR performance. To accomplish this, as Figure 7.3 illustrates, we supplement the queries $\mathbf{K}_{tr}$ and values $\mathbf{V}_{tr}$ with relative representations [90] enhancing neighboring relations. In addition, we apply a fixed window size Gaussian distribution as additive bias to mask the self-attention scores, further upgrading local context modeling. Consequently, the output of the multi-head self-attention layer is given by:

$$\mathbf{F}_{tr} = \text{concat}\{\text{softmax}(\text{attn}^a + \text{bias}^a)\mathbf{V}_{tr}^a\}_{a=1}^{n_a}.$$

7.2.3 Alignment Module

As previously stated, our CSLR model incorporates an alignment module that aligns the probability distribution produced by the sequence learning model and the one generated by the visual features of the RNN module. This is achieved by combining the CTC loss with a knowledge distillation loss [92]. Specifically, the sequence learning model probability distribution $\mathbf{D}_{SM}$ is obtained by feeding the latent representations deduced from the Transformer encoder into a linear projection layer, as well as a softmax activation. Thus, the CTC loss function (see also Section 5.2.1) is formulated as:

$$\mathcal{L}_{CTC} = -\log p(\mathbf{y}|\mathbf{x}).$$

The visual module probability distribution $\mathbf{D}_{VM}$ is produced by passing the visual features generated by the RNN module through a linear projection layer and, subsequently, a softmax activation. Next, we apply the KL-divergence loss function that was first presented in [64]. This function minimizes the distance between the probability distributions produced by the output of the sequential and visual modules, respectively. It can be expressed as follows:

$$\mathcal{L}_V = \text{KL}(\text{softmax}(\frac{\mathbf{D}_{VM}}{\tau}), \text{softmax}(\frac{\mathbf{D}_{SM}}{\tau})),$$

where τ denotes the temperature. The two loss functions are combined linearly during training, i.e: $\mathcal{L}_M = \mathcal{L}_{CTC} + \mathcal{L}_V$.

7.2.4 Ensemble Module

Direct posterior combination, i.e., late fusion, of the two individual modalities, namely the RGB- and the optical flow-based ones, has proven to be inefficient in CTC-based alignment modules due to non-aligned spike timings in the probability distributions. We address this by utilizing the two-step training methodology introduced in [91]. In particular, we choose the CSLR model trained with only the RGB modality as the guiding model, with its probability distribution being transformed into a mask $\mathbf{M}$ with ones at the output symbol with the greatest posterior and zeros at other symbols, together with the blank symbol. During the guided model training, specifically that of the optical flow stream, its probability distribution $\mathbf{D}$ is imposed on element-wise multiplication with the mask $\mathbf{M}$. Thus, the guided loss function, given by $\mathcal{L}_G = -\sum \hat{\mathbf{D}}$, is computed as the summation of masked posteriors ($\hat{\mathbf{D}} = \mathbf{M} \odot \mathbf{D}$) over time. Finally, the formula for the overall training loss is given by: $\mathcal{L} = \mathcal{L}_{CTC} + \mathcal{L}_G$. Specifically, for RGB stream training the corresponding loss function is formulated as:

$$\mathcal{L} = \mathcal{L}_{CTC} + \mathcal{L}_V,$$

while in the case of the entire system training, including optical flow stream, the following loss function is employed:

$$\mathcal{L} = \mathcal{L}_{CTC} + \mathcal{L}_V + \mathcal{L}_G.$$

7.3 Experimental Framework

System performance assessment is carried out on two popular continuous German SL corpora, i.e., the *RWTH-PHOENIX Weather 2014 multi-signer corpus* [13] and the *RWTH-PHOENIX Weather 2014T dataset* [9], as well as the collected studio *SL-ReDu continuous GSL corpus* [29] (see also Section 2.2). Specifically: (i) For the RWTH-PHOENIX Weather 2014 corpus, we consider its MS experimental setting, where a total of 5,672 video samples are assigned for training, 540 for validation, and 629 for testing; (ii) For the RWTH-PHOENIX Weather 2014T dataset, we employ the official MS split, consisting of 7,096 videos for training, 519 for validation, and 642 for testing; and (iii) Finally, for the SL-ReDu Continuous GSL corpus, we consider two experimental frameworks for evaluating performance of the continuous GSL recognition system, i.e., the MS and the SI setting. In both cases, the official MS and SI splits are employed, namely data from all signers are split between training (3,858 videos), validation (593 videos), and testing (1,479 videos) with a single fold used in the MS setting, while in the SI experimental paradigm a 7-fold cross-validation framework is applied, with each fold containing training and validation data from 18 signers, whereas testing is performed on the remaining 3. The process repeats over all 7 folds to cover all signers.

Concerning the implementation of our evaluated CSLR model, we employ two BiLSTM layers in the RNN module, each with a hidden dimensionality of 512 and subjected to nor-

Table 7.1: Ablation study for the RNN module (RNN), the Relative Position (RP) encoding, and the Gaussian Bias (GB) of single and both modalities. The evaluation is conducted in terms of word error rate (%) using all three datasets.

Modalities	RNN	RP	GB	RWTH-2014	RWTH-2014T	SL-ReDu CGSL (MS)
RGB	✓	✓	✓	27.55	25.60	15.95
				23.25	22.85	13.21
				24.05	23.12	14.10
				21.25	21.34	12.24
Optical Flow	✓	✓	✓	29.18	29.24	17.07
				26.20	26.35	15.71
				25.87	26.49	16.07
				25.07	24.13	14.01
Both	✓	✓	✓	20.89	20.73	11.21

malization. In addition, we utilize a 3-layer Transformer encoder for the sequence learning module, with hidden states of 512-dim and 8 heads. As in the work of [57], we use a fixed window size of 6.3 for the Gaussian bias in all datasets. For training, we deploy the Adam optimizer [141] with an initial learning rate of 0.0001, which is decreased by a factor of 0.0001. Additionally, a dropout rate of 0.1 and a batch size of 2 are employed. For inference, we employ beam search decoding [158] with a beam width of 5. For the alignment module, we consider a temperature τ in the $\mathcal{L}_V$ loss set at a fixed value of 8. We develop all our models in PyTorch framework [143] and conduct all our experiments on an NVIDIA GeForce RTX 3090 GPU.

7.4 Experimental Results

This section presents the experimental results and analysis of the performance evaluation conducted on the introduced continuous SLR approach. Firstly, we assess different model variants to demonstrate the contribution of each individual component of our system. In addition, we compare our model against alternative state-of-the-art CSLR methods in the literature. The performance of our CSLR model is assessed on all datasets of Section 7.3 quantitatively using the word error rate (WER, %), where the number of substitutions, deletions, and insertions in the predicted gloss sequence hypotheses are taken into account.

First, in Table 7.1 we demonstrate the contribution of the primary pillars of our model regarding both the visual and the sequential modules, i.e., the RNN module, the relative position encoding, and the Gaussian bias, when single and both modalities (RGB and optical flow) are considered. As it can be deduced, our model exhibits competitive performance when both modalities are employed, yielding a WER of 20.89%, 20.73%, and 11.21% on the RWTH-2014, the RWTH-2014T, and the SL-ReDu CGSL (MS) datasets. This demonstrates the benefit of integrating various feature representations that are complementary to each other into an SLR system. In addition, the RNN module seems to be the most robust component, while integrating relative position encoding and a Gaussian bias into the Transformer sequential module significantly enhances system performance. In addition, the RGB appearance modality constitutes a slightly more robust visual representation, achieving the lowest error rates. Note that among the three datasets considered, the lowest errors are obtained in the case of the SL-ReDu CGSL dataset. This is primary due to the fact that this

Table 7.2: Ablation study concerning the various loss functions. Evaluation comparison (in word error rate (%)) on all CSLR datasets.

Method	$\mathcal{L}_{CTC}$	$\mathcal{L}_V$	$\mathcal{L}_G$	RWTH-2014	RWTH-2014T	SL-ReDu CGSL (MS)
Ours	✓			25.64	25.67	17.01
	✓	✓		24.16	24.85	15.72
	✓		✓	21.75	21.95	13.55
	✓	✓	✓	20.89	20.73	11.21

corpus exhibits lower rate of glosses per sentence as compared to the other two corpora.

Further, in Table 7.2, we investigate the impact of the various loss functions integrated into our model. Clearly, the CTC guiding loss function demonstrates a greater contribution to the system performance compared to the visual alignment loss. Additionally, the visual alignment loss leads to a further absolute error improvement by 0.86%, 1.22%, and 2.34% in the RWTH-2014, the RWTH-2014T, and the SL-ReDu CGSL (MS) corpora, respectively. Sample model predictions generated by the proposed CSLR model, as well as several variations of it against the reference (ground truth), when applied to frame sequences of all datasets are shown in Figure 7.4. Note also, that on the SL-ReDu CGSL corpus, under the SI setting, we obtain a WER of 16.50%, which represents a reasonable degradation over the MS WER of 11.21% reported in Tables 7.1 and 7.2. It should be emphasized that these experimental results represent the first ever SLR evaluation on this particular corpus.

Finally, we compare the performance of the introduced model against the current state-of-the-art for the RWTH-2014 multi-signer corpus (Table 7.3) and RWTH-2014T (Table 7.4). Our model demonstrates competitive performance on these two CSLR datasets, achieving a WER of 20.89% on the RWTH-2014 multi-signer corpus and 20.73% WER on the RWTH-2014T dataset. In particular, in both cases, our model outperforms most results in the literature, coming very close to the state-of-the-art (19.98% WER) of [172] on the RWTH-2014 dataset, which employs a spatio-temporal based Transformer and the state-of-the-art (20.40% WER) of [57] on the RWTH-2014T dataset, which relies on local Transformer encoder and two auxiliary constraints for enhancing sequence learning.

Our model includes 36M parameters. Finally, we assess the effectiveness of our model by comparing it to a modified version that replaces the RNN module with a TCN layer. This substitution leads to a 0.78% absolute increase in WER on the RWTH-2014 multi-signer corpus.



REF: OST SEE TIEF DRUCK ZONE NORD

HYP ($\mathcal{L}_{CTC}$ only): NACHT SONNE REGEN GLATT ZONE

HYP ($\mathcal{L}_{CTC} + \mathcal{L}_V$): HEUTE NACHT TIEF SCHAUER ZONE NORD

HYP ($\mathcal{L}_{CTC} + \mathcal{L}_G$): OST AUFLOCKERUNG TIEF SCHAUER ZONE NORD

HYP ($\mathcal{L}_{CTC} + \mathcal{L}_V + \mathcal{L}_G$): OST SEE TIEF DRUCK ZONE NORD



REF: MORGEN IX MEISTENS WOLKE

HYP ($\mathcal{L}_{CTC}$ only): DANN REGEN WOLKE

HYP ($\mathcal{L}_{CTC} + \mathcal{L}_V$): MORGEN LOCH SONNE

HYP ($\mathcal{L}_{CTC} + \mathcal{L}_G$): MORGEN IX ENORM WOLKE

HYP ($\mathcal{L}_{CTC} + \mathcal{L}_V + \mathcal{L}_G$): MORGEN IX MEISTENS WOLKE



REF: Glass Dangerous

HYP ($\mathcal{L}_{CTC}$ only): Monday Tolls Blue

HYP ($\mathcal{L}_{CTC} + \mathcal{L}_V$): Dangerous

HYP ($\mathcal{L}_{CTC} + \mathcal{L}_G$): Glass Tolls

HYP ($\mathcal{L}_{CTC} + \mathcal{L}_V + \mathcal{L}_G$): Glass Dangerous

Figure 7.4: Model predictions generated by the proposed CSLR model, as well as several variations of it against the reference (ground truth), when applied to frame sequences of the *RWTH-PHOENIX Weather 2014 corpus* [13] (upper row), the *RWTH-PHOENIX Weather 2014T dataset* [9] (middle row), and the collected studio *SL-ReDu continuous GSL corpus* [29] (lower row). Note that Greek labels are provided in their English translation for readability purposes. Wrong gloss predictions are colored in red and correct predictions in green.

Table 7.3: Evaluation comparison on the RWTH-2014 dataset in terms of WER (%).

Model	Feature streams	WER (%) ↓
SubUnet [62]	Hand	40.70
SLT [10]	Full Frame	24.59
CNN-LSTM-HMM [174]	Hand/Face & Language Model	24.10
VAC [65]	Full Frame	22.30
SMKD [64]	Full Frame	21.00
STMC [35]	Full Frame	20.70
C2SLR [57]	Full Frame & Skel.	20.40
STTN [172]	Full Frame & Word Embeddings	19.98
Ours	Full Frame & Flow	20.89

Table 7.4: Evaluation comparison on the RWTH-2014T dataset in terms of WER (%). Notation: appearance (Full Frame), hand appearance (Hand), face appearance (Face), skeleton (Skel.), optical flow (Flow).

Model	Feature streams	WER (%) ↓
Re-Sign [170]	Full Frame	26.60
SFD+SGS+SFL [63]	Full Frame & Word Embeddings	26.10
Bi-ST-LSTM-A [71]	Full Frame	24.68
SLT [10]	Full Frame	24.59
CrossModal [38]	Full Frame	24.30
CNN-LSTM-HMM [174]	Hand/Face & Language Model	24.10
TDCNN [36]	Hand/Mouth & 2D skel. & Flow	23.70
SMKD [64]	Full Frame	22.40
ST-GCN [80]	Full Frame & Flow & 2D/3D Skel. & 3D Pose/Shape	21.34
STMC [35]	Full Frame	21.00
C2SLR [57]	Full Frame & 2D Skel.	20.40
Ours	Full Frame & Flow	20.73

7.5 Conclusions

In this chapter, we addressed the challenging task of CSLR, examining the contribution of learning both local signing dynamics and global structure dependencies. To achieve this, we combined a window-based RNN module, operating simultaneously on segments of the input sequence to generate local context latent representations, and a Transformer encoder, which was complemented with global structure learning obtained by multi-head attention, as well as with local context modeling through Gaussian bias and relative position encoding. To further improve model performance, we devised a bimodal system, with the proposed se-

quence learning model operating on both appearance and motion signing streams, aligning their posteriors through a guiding CTC loss function. In addition, we applied an extra auxiliary loss function, namely a knowledge distillation loss, that enhanced the window-based RNN module learning capacity by aligning its feature representations with the corresponding sequential module representations. Our experimental results on three CSLR corpora, two German and our Greek SL-ReDu one, demonstrated that the considered components, as well as the combined modalities contributed positively to the model recognition performance. In particular, the local RNN module seemed to be the most robust component, while the integration of localness into the Transformer through Gaussian bias and relative position encoding enhanced system performance. In addition, the RGB stream turned out to be the most robust modality yielding the lowest error rates. Our reported results were very competitive on the two German datasets (compared to the state-of-the-art), and they represented the first ever reported SLR results on the SL-ReDu CSLR corpus.

Chapter 8

Continuous Fingerspelling Recognition

8.1 Introduction

Among the SLR tasks introduced in the literature, limited attention has been paid to fingerspelling recognition, namely the recognition of continuous sequences of alphabet letter signs, which constitutes the focus of this chapter. Concerning this problem, it should be noted that it holds great importance to SLs, due to the fact that fingerspelling signing is frequently used for prominent words that lack dedicated signs, such as names, technical phrases, or foreign words, all of which often hold a crucial content role [44, 55].

Several studies have been proposed to address the continuous fingerspelling recognition task [43, 175–177]. For instance, the American Sign Language (ASL) fingerspelling recognition systems in [178, 179] rely on 2D-CNNs applied on depth maps and color images, while the scheme in [180] recognizes 20 out of 24 static ASL signs using PCA extracted features. In addition, a model that relies on 2D-CNNs with multiview augmentation and inference fusion from depth images has been developed in [181] for ASL alphabet recognition. In [182], a fingerspelling recognition system relying on HOG and Zernike moment features coupled with a deep belief network classifier is proposed, while in [183] a system based on a hand tracking device and a SVM classifier is introduced. Furthermore, the system in [184] exploits LBP histogram features that are derived from both color and depth data, combined with an SVM classifier. Recently, there have been proposed systems [41, 43] addressing the recognition of continuous fingerspelled sequences of alphabet letter signs, employing CNN-based features combined with semi-Markov conditional random fields and attention-based RNNs, while other studies rely on spatio-temporal feature fusion and attentional encoder-decoders.

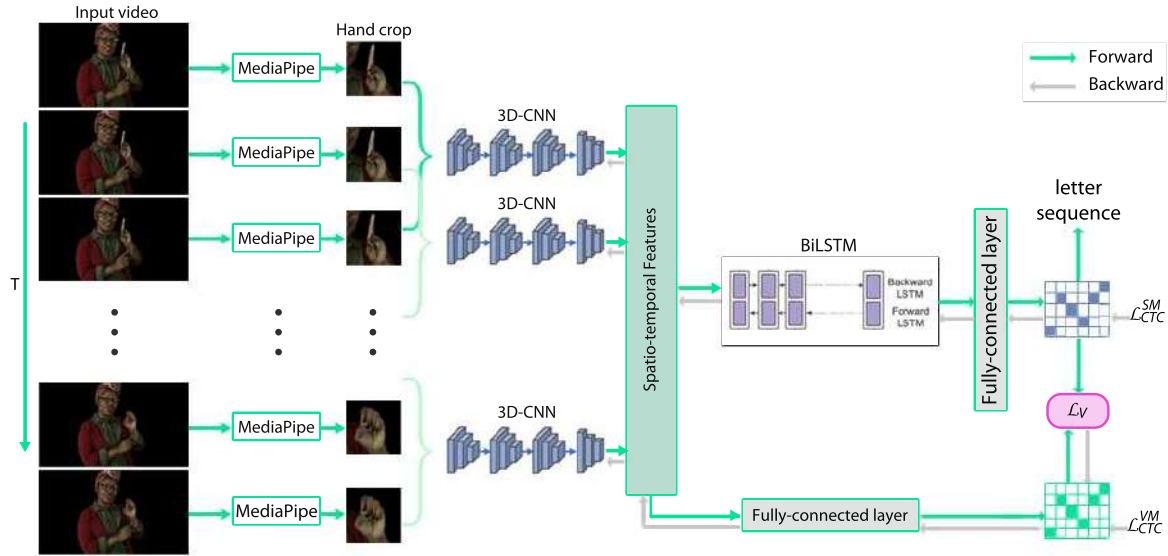


Figure 8.1: Architecture of the introduced SLR model that generates letter sequences from fingerspelling videos through spatio-temporal features produced by a 3D-CNN model (left) and a BiLSTM-based sequence learning model coupled with CTC decoding (right). Two auxiliary loss functions are also incorporated.

Further, recent schemes in the literature rely on CNNs for feature extraction combined with RNNs, typically BiLSTMs [78], followed by CTC alignment [16, 43, 185].

In this chapter, we seek to address the fingerspelling recognition task by introducing an end-to-end deep learning-based recognition scheme relying on spatio-temporal feature extraction combined with an BiLSTM encoder and a CTC alignment module. In particular, a sequence of raw image frames that corresponds to a source word (sequence of letters) is processed to produce a sequence of features, which is then passed through a BiLSTM encoder [78] producing the latent representations, followed by a CTC decoder [81] producing the sequence of letters. Our system integrates an image feature learner based on the ResNet2+1D network [85] (see also Section 4.2.3), which decouples the spatial and temporal convolutions of the 3D-CNN. The image feature learner is jointly trained with the BiLSTM encoder. Thus, the 3D-CNN model serves as a spatio-temporal visual feature learner of each video frame, while the BiLSTM encoder [78] learns their temporal relations (see also Figure 8.1). Like most sequence-to-sequence prediction problems, there is no one-to-one alignment between the input and output sequences, as a portion of frames (and their corresponding feature vectors) can be associated with a fingerspelled letter. To this end, we inte-

grate an alignment module, combining the CTC loss with a knowledge distillation loss [92] and a visual module loss function, which aligns the probability distributions generated by the sequence learning model and the 3D-CNN module, enhancing the representation capacity of both the features derived from the visual and sequential modules (see also Figure 8.1).

The main contributions of this approach lie in the design of a novel continuous fingerspelling recognition system that integrates 3D-CNNs for video feature modeling, which constitutes the first attempt to integrate 3D-CNNs into fingerspelling recognition systems in the literature. In addition, we integrate a knowledge distillation loss during training, enhancing the representation capacity of both the visual and sequential modules. We evaluate our introduced approach on the ChicagoFSWild dataset [16], as well as the SL-ReDu GSL fingerspelling corpus, demonstrating competitive performance in both cases.

The remainder of the chapter is structured as follows. In Section 8.2, we provide a description of the adopted continuous fingerspelling recognition system, including the visual module (Section 8.2.1) and the sequence learning approach (Section 8.2.2); In Section 8.3, we provide details regarding the model training data and adopted experimental framework; In Section 8.4, we report the recognition results; and the chapter is concluded in Section 8.5.

8.2 The Continuous Fingerspelling Recognition System

Continuous fingerspelling recognition aims to predict a letter sequence $\mathbf{l} = (l_1, l_2, \dots, l_N)$ based on a sequence of T image frames $\mathbf{x} = (\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_T)$. To tackle this, we introduce the approach depicted in Figure 8.1. As it can be observed, our fingerspelling recognition system comprises: (i) a 3D-CNN based visual module that operates on hand image patches cropped using the hand skeletal keypoints returned by the MediaPipe framework [111]; (ii) a BiLSTM encoder for latent visual representations modeling; and (iii) an alignment module relying on the CTC and two auxiliary visual loss functions. More details follow.

8.2.1 Visual Module

Hand ROI Cropping: A crucial component in addressing the problem of continuous fingerspelling recognition is the visual detection and tracking of the manual articulation, since typically one hand constitutes the sole articulator. To this end, we opt to employ the MediaPipe human pose detector [111] (see also Section 3.3.2). Since, the focal point of continuous

fingerspelling SLR is the manual articulation, we retain only the ROI consisting of the signing hand. In particular, here we exploit the 21 hand keypoints, which are in turn used to segment the hand ROI. Specifically, the x and y landmarks of hands with values ranging in $[0.0, 1.0]$ are normalized to the image plane based on the image width and height, respectively. Since the non-signing hand tends to be characterized by limited motion, we exploit the hand trajectory to discriminate the signing hand. Specifically, if the wrist joint remains “stable” for more than 10 consecutive frames, with x and y coordinates Euclidean distance between adjacent frames (hand position transitions) being less than 8, then this is designated as the non-signing hand. Subsequently, the maximum and minimum values of the corresponding x and y landmarks of the signing hand are employed for hand ROI cropping. It should be noted that in case of MediaPipe failure, the missing keypoints are filled with the previous existing ones.

Spatio-Temporal Appearance Features: The feature learner plays an important role in continuous fingerspelling recognition, extracting visual feature representations from image sequences. As shown in Figure 8.1, we employ a 3D-CNN model for spatio-temporal feature regression, namely the ResNet2+1D network [85] that decouples the 3D-CNNs into spatial convolutions for frame-wise feature regression and temporal convolutions to capture short-term dynamics of hand posture and motion among neighboring frames (see also Section 4.2.3). In particular, the model contains five (2+1)D convolutional blocks, encompassing both spatial and temporal convolutions that are followed by a global average pooling layer, which operates on both space and time. To further improve our model, we replace the ReLU with the SiLU [166] activation function, a non-monotonic smooth function that outperforms ReLU (see also Section 6.2.1).

Thus, a sequence of hand ROI images with length T , after appropriate rescaling to the input layer size of the ResNet2+1D network [85] (i.e., 112×112 pixels), is divided into T' subsequences using a sliding window of 8 consecutive frames and stride $s = 4$ (properly pre-padding the end of the video), where $T' = \lfloor T/s \rfloor$. Note that there is 4 frames overlap (half clip overlap) between subsequences. Thus, each hand image subsequence is propagated through the ResNet2+1D network, producing spatio-temporal features $\mathbf{z} = (\mathbf{z}_1, \mathbf{z}_2, \dots, \mathbf{z}_{T'})$. The model is pre-trained on the Kinetics dataset [137]. Feature maps are obtained by taking the output of the last spatiotemporal pooling layer, yielding 512-dim features for each subsequence.

8.2.2 Sequence Learning

The spatio-temporal features are fed into a three-layer BiLSTM [78] encoder (with hidden state dimensionality equal to 512) capturing the long-term dependencies, followed by a dense fully-connected layer and a softmax activation function producing the predicted probability scores for each letter label. Then, the CTC loss function is adopted, mapping the sequence learning model probability distribution to the sequence letter labels, using the following equation that is described in detail in Section 5.2.1:

$$\mathcal{L}_{CTC}^{SM} = -\log p(\mathbf{l}|\mathbf{x}).$$

Inspired by [65], we combine the sequence learning model CTC loss function with two auxiliary losses. Specifically, we propagate the spatio-temporal features derived from the ResNet2+1D network into a classifier based on a fully-connected layer coupled with softmax, producing probability distributions that are aligned to the letter label sequence via a CTC loss function denoted as $\mathcal{L}_{CTC}^{VM}$. Further, as in Chapter 7, the sequence learning model probability distribution $\mathbf{D}_{SM}$ is aligned with the sequence learning model probability distribution $\mathbf{D}_{VM}$ through the KL-divergence loss function, formulated as:

$$\mathcal{L}_V = \text{KL}(\text{softmax}(\frac{\mathbf{D}_{VM}}{\tau}), \text{softmax}(\frac{\mathbf{D}_{SM}}{\tau})),$$

with τ denoting the temperature. The first auxiliary loss function aligns the visual features derived from the ResNet2+1D network with the target letter sequence enhancing the visual feature learner capacity, while the KL-divergence loss function aligns short-term visual predictions and long-term context predictions through knowledge distillation. Specifically, the three loss functions are combined linearly during training as:

$$\mathcal{L}_M = \mathcal{L}_{CTC}^{SM} + \mathcal{L}_{CTC}^{VM} + \alpha \mathcal{L}_V,$$

where $\mathcal{L}$ denotes the combined loss function. In our implementation, we use temperature $\tau = 8$ and value $\alpha = 0.5$.

8.3 Experimental Framework

The performance of the introduced fingerspelling recognition model is evaluated on two publicly available datasets: (a) the *Chicago fingerspelling in the wild dataset* [16] in ASL

and (b) our collected studio *SL-ReDu fingerspelling GSL corpus* [29] (see also Section 2.2). Specifically, for the first dataset, we employ its official SI split with no signer overlap between the different sets, i.e., 5,455 videos for training performed by 87 subjects, 981 validation videos from 37 signers, and 868 test videos expressed by 36 informants. For our experiments on the SL-ReDu Fingerspelling GSL (FGSL) corpus, we deploy two experimental paradigms, namely the MS one, where data from all 21 signers being split between training (853 videos), validation (155 videos), and testing (546 videos) with a single fold used, as well as the SI paradigm, where a 7-fold cross-validation is used, with each fold containing training data (80% of the fold) and validation data (20% of the fold) from 18 signers, whereas testing is performed on the remaining 3 signers. The process repeats over all 7 folds to cover all signers.

Concerning the training of our developed model, it is conducted employing the Adam optimizer [141] with an initial learning rate of 0.001, decreased by a factor of 0.5 in each iteration. The mini-batch size is fixed to 128. In addition, for comparison purposes, we also consider a simpler model from our work in [37]. There, 3D hand skeletal information is combined with additional SL informative streams, namely 2D skeletal data derived from the OpenPose [108] detector, as well as ResNet-18 [186] based hand- and mouth-region appearance representations. These are then concatenated and propagated through the attention-based convolutional encoder-decoder of Section 5.2.2. Note that the 3D hand skeleton is acquired by projecting the 2D hand-joint coordinates, retrieved by the OpenPose framework, to the 3D space via a deep multi-layer neural network [187]. The 3D hand skeleton regression network is trained for 150 epochs using the Adam optimizer [141], a batch size of 64, an initial learning rate of 0.001 and exponential decay. In addition, the weights of the linear layers are set using Kaiming He initialization [188]. The attentional CNN encoder-decoder model involves 3 layers with kernel width 5 and 256 hidden units, and its training is carried out via the Adagrad optimizer [189] with an initial learning rate of 0.003, decreased by a factor of 1.0. We also employ dropout of 0.1 and beam search of width 5. Both models are implemented in PyTorch [143] and all experiments are carried out using GPU acceleration.

8.4 Experimental Results

In this section, we present the experimental results regarding the performance evaluation on the introduced continuous fingerspelling recognition approach. Firstly, we compare the

Table 8.1: Ablation study concerning the auxiliary loss functions and the video feature learner. The evaluation is conducted in terms of letter accuracy (LAcc, %) on both datasets (for SL-ReDu FGSL in the SI setting).

Backbone Models	$\mathcal{L}_{CTC}^{SM}$	$\mathcal{L}_{CTC}^{VM}$	$\mathcal{L}_V$	ChicagoFSWild	SL-ReDu FGSL (SI)
ResNet2+1D & BiLSTM	✓			47.27	87.98
	✓	✓		49.85	90.15
	✓		✓	48.57	89.47
	✓	✓	✓	51.69	92.49
ResNet-18 & BiLSTM	✓			42.91	86.77
	✓	✓		45.25	89.98
	✓		✓	44.20	88.25
	✓	✓	✓	47.12	91.03

introduced against several variations of it, demonstrating the benefit of exploiting 3D-CNNs for continuous fingerspelling recognition, as well as the contribution of integrating visual enhancement auxiliary loss functions during training. Further, we benchmark the performance of our system against the state-of-the-art continuous fingerspelling recognition methods in the literature. The performance of our model is assessed on both datasets of Section 8.3 quantitatively in terms of letter accuracy (LAcc, %).

First, we demonstrate the benefits of the primary components integrated into our continuous fingerspelling recognition model, namely the 3D-CNN based visual module and the two auxiliary loss functions. As deduced from Table 8.1, our model turns out superior to the considered alternative relying exclusively on a 2D-CNN image feature learner as evaluated on both datasets, yielding letter accuracy of 51.69% on the ChicagoFSWild corpus and 92.49% on the SL-ReDu FGSL dataset. This reveals the power of leveraging short-range contextual relations modeled via the ResNet2+1D network based visual module that can boost the performance of the letter prediction task. In addition, the integration of both auxiliary loss functions into our model demonstrates that supervised training on the visual module benefits system performance, improving LAcc by 4.42% and 4.47% absolute on the ChicagoFSWild and the SL-ReDu FGSL (SI), respectively. It is also worth noting that adopting the visual alignment loss leads to smaller gains than the visual module loss function only, while integrating both yields the highest accuracies. Further, the ResNet-18 performs well, but the 3D-CNN model turns out superior to the considered alternative in both fingerspelling datasets. To emphasize this, in Figure 8.2 we illustrate the sequence prediction results obtained by the introduced



REF: M A C H

HYP ($\mathcal{L}_{CTC}^{SM}$ only): B N O H

HYP ($\mathcal{L}_{CTC}^{SM} + \mathcal{L}_V$): B M A O H

HYP ($\mathcal{L}_{CTC}^{SM} + \mathcal{L}_{CTC}^{VM}$): N A C H

HYP ($\mathcal{L}_{CTC}^{SM} + \mathcal{L}_{CTC}^{VM} + \mathcal{L}_V$): M A C H



REF: L A M P A

HYP ($\mathcal{L}_{CTC}^{SM}$ only): T E N P A

HYP ($\mathcal{L}_{CTC}^{SM} + \mathcal{L}_V$): L A N T E

HYP ($\mathcal{L}_{CTC}^{SM} + \mathcal{L}_{CTC}^{VM}$): L A M M A

HYP ($\mathcal{L}_{CTC}^{SM} + \mathcal{L}_{CTC}^{VM} + \mathcal{L}_V$): L A M P A

Figure 8.2: Model predictions generated by the proposed fingerspelling recognition model, as well as several variations of it against the reference (ground truth), when applied to frame sequences of the *Chicago fingerspelling in the wild dataset* [16] (1st row) and the collected studio *SL-ReDu fingerspelling GSL corpus* [29] in its SI experimental framework (2nd row). Note that Greek labels are provided in their English translation for readability purposes. Wrong letter predictions are colored in red and correct predictions in green.

fingerspelling recognition model, as well as several variations of it regarding the incorporation of the auxiliary loss functions, when applied to sample test videos of both fingerspelling datasets. It should be noted that performance on the SL-ReDu FGSL dataset is significantly better than on the ChicagoFSWild corpus, due to the studio-like recording setup of the former vs. the in-the-wild nature of the latter.

Next, Table 8.2 reports the evaluation comparison of the introduced continuous fingerspelling recognition model against state-of-the-art models on the ChicagoFSWild fingerspelling dataset under the SI experimental paradigm. It can be observed that the proposed outperforms alternative schemes, revealing the power of capturing spatio-temporal sign articulation dynamics. In particular, our system outperforms the state-of-the-art result obtained

Table 8.2: Letter accuracy (LAcc, %) comparison of state-of-the-art on the ChicagoFSWild dataset under SI setting.

Model	Feature streams	LAcc (%) $\uparrow$
R-CNN-Att [16]	Full Frame	45.10
CNN-Att [37]	Hand/Mouth & 2D/3D Skel.	47.93
FG-Transformer [191]	Full Frame	48.36
Siam-LSTM [192]	Full Frame	48.00
Iterative-LM [190]	Full Frame	49.60
Ours	Hand	51.69

in [190], yielding 5% relative error rate reduction (2.63% absolute). Note that the system of [190] adopts the Siamese network of ResNet-50 for feature representations extraction in conjunction with a BiLSTM encoder, and a fully-connected layer, and a softmax classifier, with alignment conducted via the CTC loss function. Further, our introduced approach exhibits significant gains over our earlier fingerspelling recognition system of [37], yielding 8% relative accuracy improvement (3.76% absolute). This demonstrates that CTC based decoding constitutes a more powerful alignment module than the attention-based one concerning the fingerspelling recognition task.

Moreover, we evaluate the performance of the proposed model on the SL-ReDu FGSL under its MS experimental framework, yielding 96.01% letter accuracy, which is a reasonable improvement over the corresponding SI result (92.49%). It is worth noting that these constitute the first ever experimental results reported in the literature regarding the SL-ReDu FGSL corpus.

8.5 Conclusions

In this chapter, we introduced a continuous fingerspelling recognition system, employing a visual module based on the ResNet2+1D model with decoupled spatial and temporal convolutions, accomplishing both prevalent spatial modeling potential and motion-aware modeling adaptability. The sequence module of the system incorporated a BiLSTM encoder coupled with a fully-connected layer and a softmax for posterior distribution generation, aligned with the target letter sequence via the CTC loss function. To improve end-to-end model training, we integrated two additional auxiliary loss functions, enhancing the model to produce superior predictions through its visual and the sequential modules. In particular, we combined the

CTC loss function with a knowledge distillation loss, aligning the probability distributions generated by the sequence and the visual modules, as well as an additional auxiliary loss function generated by the probability distributions of the visual module, enhancing the representation capacity of the 3D-CNN visual feature learner. The results on two fingerspelling corpora, one in ASL and the other in GSL collected by us, demonstrated that the considered components and the integrated loss functions positively contributed to the model recognition performance. Our model outperformed the current state-of-the-art on the ChicagoFSWild ASL fingerspelling dataset, whereas we reported the first ever results on our SL-ReDu fingerspelling GSL corpus under both MS and SI experimental frameworks.

Chapter 9

Cued Speech Recognition

9.1 Introduction

Speechreading plays a crucial role in speech perception for the hearing impaired, despite its inherent limitations caused by the visual similarity of some speech patterns. This is because of several phonemes that share identical lip movements, known as visemes, which can lead to inaccuracies in speechreading. To tackle this issue, Cornett [193] invented the cued speech communication system, which disambiguates the mouthing patterns by integrating hand positioning and gestural cues. In cued speech, the simultaneous articulation of mouth patterns, handshapes, and hand positioning in relation to the mouth offers a comprehensive visual representation of the phonological system of a spoken language. Cued speech has been adopted in several languages and dialects. For example, as it can be seen in Figure 9.1, the French cued speech consists of five distinct hand positions used to encode vowels, as well as eight handshapes that are employed to encode consonants, and eight lip contour patterns. This conjunction of hand positions, handshapes, and lip contour patterns results in a total of 34 phonemes within the French cued speech system [45]. Similarly, British English cued speech employs four distinct hand positions to represent the twelve monophthongs and four hand slips to represent the eight diphthongs. Additionally, it utilizes eight different hand shapes in combination with lip patterns to encode the twenty-four consonants, resulting in a total of forty-four phonemes. Example video frames of cued speech articulation in French and British English are shown in Figure 2.7. These frames are extracted from corpora collected for this purpose [48, 101].

Because mouthing and gestural patterns are the main means that cued speech informa-

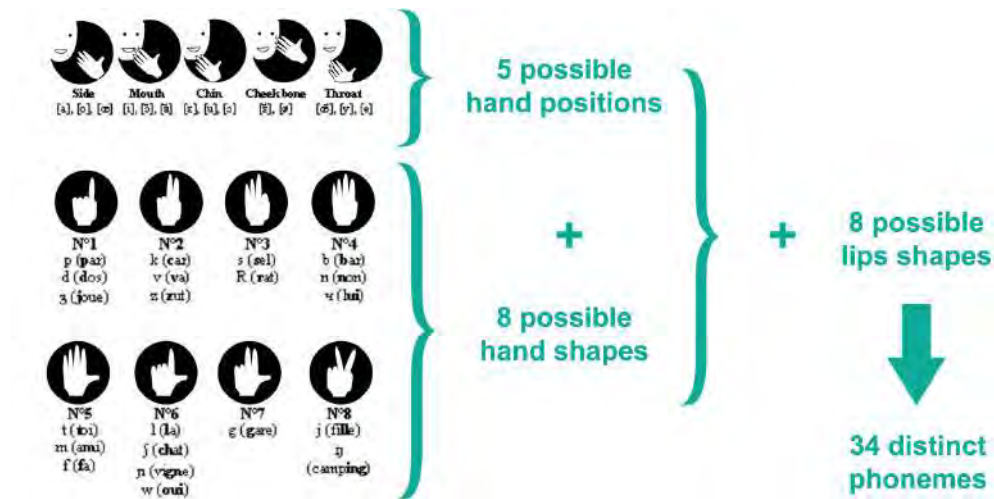


Figure 9.1: French cued speech phonetic encoding system (figure adapted from [194])

tion is delivered, lipreading [195] and sign recognition [196, 197] constitute dominant aspects in automatic cued speech recognition. Due to the availability of cued speech data resources [48, 101, 198], the problem has lately gained attention in the literature. For instance, as the visual front-end, early cued speech recognition systems used artificial markings to identify the articulators of interest [45, 199], while more recent works use deep learning techniques to segment the hand regions and track lips [48, 94]. As visual module, appearance-based visual feature extraction is primarily performed via CNNs [46, 48]. For sequence learning, most schemes use HMMs [45, 46, 194, 199]. In addition to the former aspects, the natural asynchrony between handshape and mouthing articulation is another significant component of cued speech. As described in [194], the hand-shape precedes the mouthing articulation by around one syllable. The issue is also considered in [46, 101, 200]. Recently, there have been proposed schemes relying on CNN feature learners combined with BiLSTM [201] or Transformer [202] based cross-modal learning.

This chapter focuses on the task of automatic cued speech recognition from upper-body videos, without the use of any artificial markings, building a system [94] where handshape and lip visual representations regressed through a 3D-CNN feature learner, as well as hand position embeddings relative to the mouth, are fused and fed to a TDS block structure followed by a multi-step attention-based convolutional decoder for phoneme prediction. In particular, our cued speech recognition system commences with the tracking and, by extension, the region segmentation of the mouth and hands of the cued speech interpreter through the OpenPose [108] 2D skeleton regression model. Next, we explore the potential benefits of

incorporating feature streams, which capture the positional, appearance, and structural information of the articulators of interest, such as the signer's 2D hand skeleton coordinates and 2D lip points, inferred via OpenPose, the 3D hand skeletal coordinates derived through the architecture in [187], as well as 3D-CNN based appearance feature representation. Finally, we modify the sequence learning model of [94] by considering its TDS convolution block structure [93, 94] to encode the fused visual features, in conjunction with a CTC decoding approach [81] for phonetic sequence prediction. It should be noted that our approach expects our model to learn the synchronization of the hand and mouth feature streams implicitly, rather than relying on explicit synchronization prior to their fusion.

The proposed system is evaluated on two publicly available cued speech datasets in French [48] and British English [101], each involving a single subject (see also Figure 2.7). We conduct a comparative analysis of our approach with various sequence learning models and explore the integration of several visual feature streams. The introduced system demonstrates superior performance, as compared to our earlier research on the topic [94]. Specifically, a notable decrease in the absolute Phoneme Error Rate (PER) is obtained, namely of 8.87% (from 29.12% to 20.25%) in the French cued speech corpus and 3.67% (from 36.25% to 32.58%) in the British English cued speech set.

The rest of the chapter is structured as follows: In Section 9.2, we overview the adopted cued speech recognition system, including the visual module (Section 9.2.1) and the sequence learning approach (Section 9.2.2); In Section 9.3, we provide a description of the model training data and adopted experimental framework; In Section 9.4, we report the recognition results; and the chapter is concluded in Section 9.5.

9.2 The Cued Speech Recognition System

In this section, we present our introduced approach for cued speech video recognition, as also schematically illustrated in Figure 9.2. As it may be observed, the cued speech recognition system comprises three distinct pillars: (i) a visual module addressing the detection and tracking of the various articulators, as well as the extraction of various visual feature representations concerning the hand and mouth articulation; (ii) the ensemble module used for the early fusion of the extracted feature streams; and (iii) the sequence module for phoneme prediction. Next, a detailed overview of all system modules is provided.

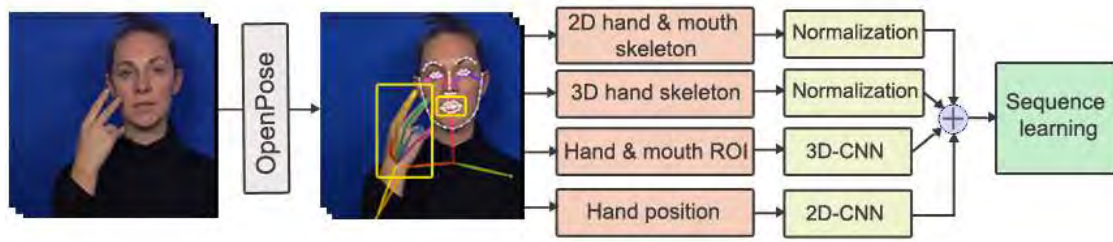


Figure 9.2: Architecture of the introduced cued speech recognition system that generates phonemes from cued speech videos, following the detection of hand and mouth articulators (left), the extraction and fusion of various feature streams (middle), and sequence learning for phoneme prediction (right).

9.2.1 Visual Module

Hand and Mouth Detection: As already mentioned in Section 9.1, cued speech exclusively relies on manual articulation along with mouthing patterns. Hence, it is obvious that a core component of a cued speech recognition system regards the detection and tracking of the two articulators in space and time. To this end, we exploit the OpenPose library [108] (see also Section 3.3.1), which infers 2D human body skeletal joints and facial keypoints from single RGB images using deep convolutional pose models (see also Figure 9.3(a),(d)). Based on the hand and mouth keypoints, we further localize and segment the respective ROIs of the signer hands and mouth, as also shown in Figure 9.3(c),(f). Note that in case of OpenPose failure, most likely because only part of the signer’s body is visible in the corpora considered here (see also Figure 2.7), we resort to the detection and ROI extraction approach from our previous work [36], described in detail in Section 3.2.

2D Hand and Mouth Keypoint Features: Since only manual and mouthing articulation participates in cued speech, we retain 41 OpenPose keypoints in total, namely 21 hand-pose joints of the signing hand (the right hand in both considered datasets) and 20 facial landmarks corresponding to the lip region, all provided in 2D coordinates on the image plane, thus resulting in a 42-dim feature vector for the hand and 40-dim features for the mouth (82-dim in total). To obtain scale invariance, countering the natural variability in signer-camera relative positioning, the retained keypoint coordinates are normalized before being passed to the ensemble module. In particular, the 2D points of interest are converted to a local coordinate system with the wrist keypoint and the upper-middle lip keypoint being the respective ori-

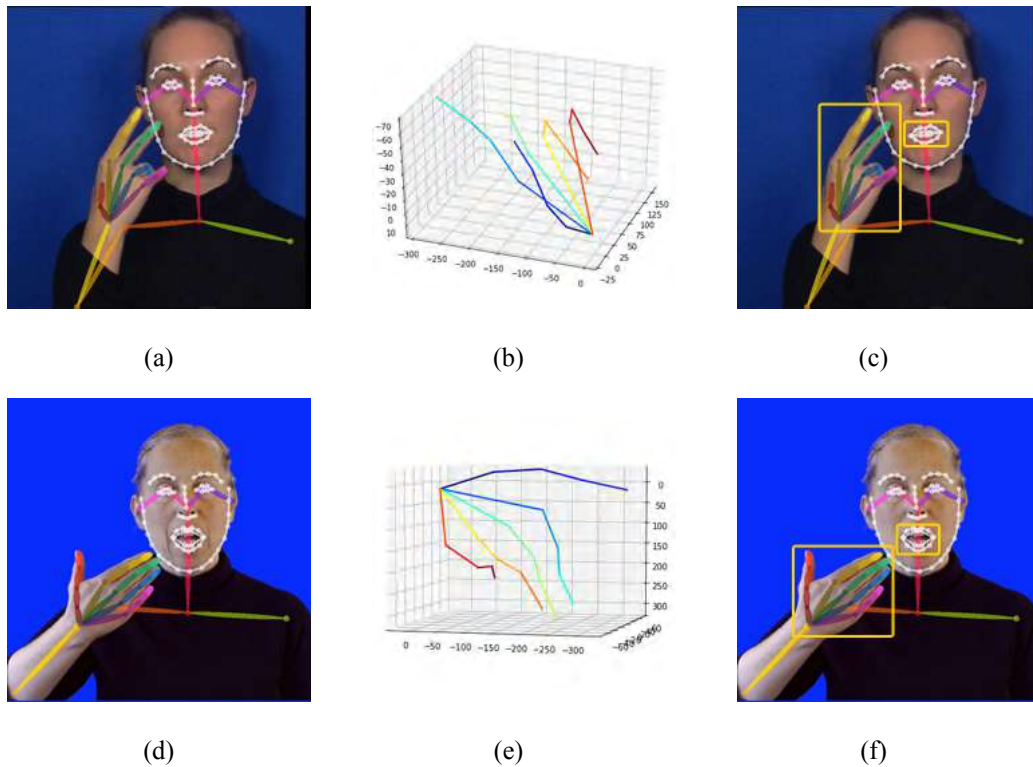


Figure 9.3: Examples of articulator detection and keypoint feature extraction on the French cued speech dataset (upper row) and the British English cued speech corpus (lower row). Shown, column-wise, left-to-right: (a,d) 2D “skeletal” joints returned by OpenPose; (b,e): inferred 3D keypoints of the signing hand; (c,f): bounding boxes of the signing hand and mouth ROIs derived based on the corresponding OpenPose 2D keypoints.

gins and subsequently all coordinates are rescaled to yield unit distance between the left and right shoulders. Note that in case of OpenPose failure estimating the desired keypoints, the missing keypoints are substituted by the previous existing ones.

3D Hand Keypoint Features: In addition, we explore the contribution of incorporating more informative hand skeletal features in the cued speech recognition system that encapsulate hand trajectory information in 3D. In particular, we pursue 3D hand skeletal joint regression, by lifting 2D hand joint locations to the 3D space [37, 80]. As a first step, we use the 2D human skeleton of the hand, extracted earlier via the OpenPose framework [108]. The 21 2D hand keypoints, corresponding to each hand, are provided as input to the hand-pose regression model and the output is the coordinates of hand joints in the 3D space. The wrist is assumed as the origin in both the 2D and 3D coordinate systems, so as to ensure translation invariance. We have observed that a simple linear architecture, similar to the one proposed in [187] demonstrates high generalization capabilities. Our model is a deep multi-layer neu-

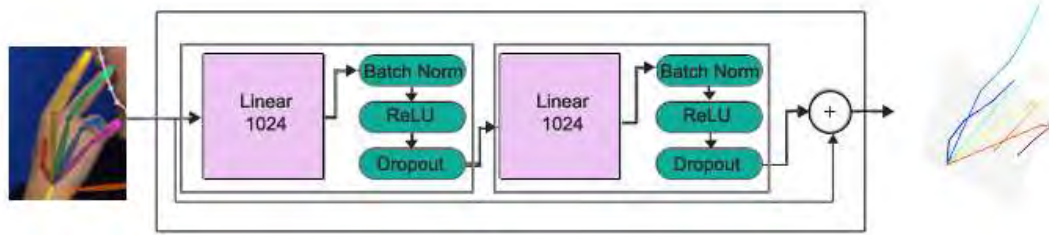


Figure 9.4: Model architecture for the generation of 3D hand skeleton from corresponding 2D information (figure adapted from [37]).

ral network (see also Figure 9.4), whose main building block is a linear layer, followed by batch normalization, dropout and ReLU activation. This unit is repeated twice and the two units share a residual connection. By including batch normalization and dropout, the model's robustness to noisy 2D detections is enhanced. Additionally, the inclusion of residual connections improves the model's generalization ability. For network training, we use the Rendered HandPose dataset [203], which constitutes a large-scale 3D hand pose set based on synthetic hand models.

The model yields 63-dim features, corresponding to the 21 hand 3D joints (see also Figure 9.3(b),(e)). For translation and scale invariance, as in the case of 2D skeletal features, the keypoints are normalized based on the distance between the hand shoulder and elbow joints of the hand, and regarding the wrist as the system origin.

Hand and Mouth Appearance Features: Appearance information of the articulation process plays a central role in cued speech recognition. Here, we consider such at a local level of the two main cued speech articulators, namely the signing hand and the lip region, as in our earlier work [36]. Thus, we obtain appearance visual representations of the hand and mouth ROIs. In particular, after proper rescaling of each ROI (96×96 pixels), we apply a 3D-CNN feature learner on three temporally adjacent ROIs of the signing hand or mouth. For this purpose, we deploy the 3D ResNet-34 network [129], which incorporates 3D convolutions with dimensions of $3 \times 3 \times 3$ and 3D pooling. The network is pre-trained on the Kinetics dataset [139], yielding feature maps (512-dim) from the global average pooling layer output.

Hand Position Detection and Representation: As mentioned in Section 9.1, the location of the hand in relation to the mouth is of utmost importance in the field of cued speech. To acquire this, we utilize the 2D coordinates of the upper skeletal joint of the signing hand to obtain the hand's relative location. Subsequently, we pass the former through a five-layer 2D-CNN, which consists of three fully-connected layers, yielding 64-dim hand positional

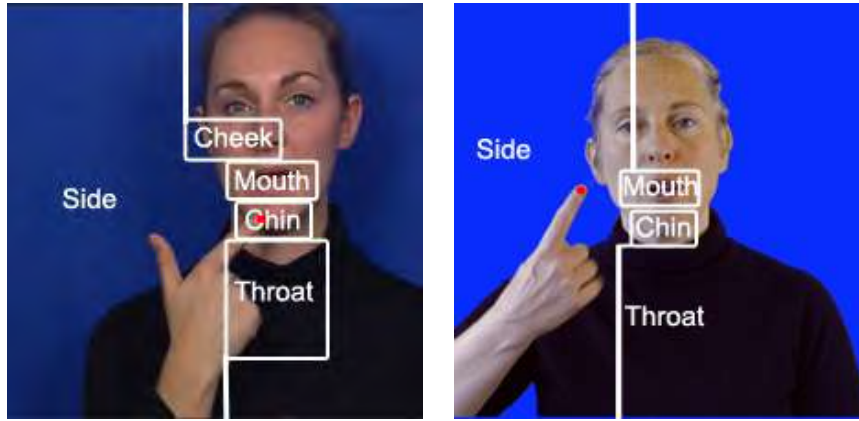


Figure 9.5: Example frames of the French cued speech (left) and British English cued speech (right) datasets, showing the location-based area division of possible signing hand positions relative to the mouth, which we use to obtain hand position embeddings in Section 9.2.1.

embeddings. The CNN is a multi-class model that categorizes different location areas of the signing hand in relation to the mouth. There are a total of five classes, known as location areas for French cued speech, while in the case of British English cued speech, there are four classes (see also Figure 9.5).

Feature Fusion: The regressed feature streams are combined by early fusion via simple vector concatenation, resulting in a 1233-dim feature vector for each video frame. This vector consists of the following components: 42 2D hand skeletal features, 40 mouth keypoints, 63 3D hand skeletal joints, 512-dim hand appearance features, 512-dim mouth appearance representation, and 64-dim hand positional embeddings. The combined vectors are subsequently fed into the sequence learning module to predict the phonetic sequence of the cued speech video.

9.2.2 Sequence Learning

In our approach, we treat continuous phoneme recognition in cued speech videos as a sequence-to-sequence prediction task, adopting a scheme relying on a TDS convolutional encoder [36, 93] in conjunction with a CTC decoder [81]. In particular, the latent representations derived from the ensemble module are fed into a TDS convolutional encoder, which involves two blocks, namely a 2D convolution over time followed by a fully-connected layer. The TDS encoder starts with a 2D convolutional layer being fed with an input of size $T \times w \times p$, where T is the number of time-steps, w represents the input width, and p denotes the number

of channels. The network employs $k \times 1$ convolutional kernels, which consist of kp^2 parameters. Additionally, downsampling is performed using a stride of 2. The convolution layer, complemented with a ReLU non-linearity, is followed by a fully-connected layer consisting of two 1×1 convolutions with a stride of 2, also coupled with ReLU non-linearity. The convolutional block output is reshaped into a $T \times 1 \times wp$ before being fed into the fully-connected layer. To enhance model efficiency, residual connections [186] and layer normalization are incorporated after convolutional and fully-connected blocks. Since the output of each TDS layer is compressed in time, we increase the number of output channels of each convolutional layer by multiplying it by a factor equal to the input features dimension divided by the number of channels of input features. Further, dropout is applied after each ReLU layer.

The output of the TDS convolutional encoder is further subjected to a linear projection layer and a softmax function, resulting in a probability distribution over all phoneme labels used to calculate the CTC loss (see also Section 5.2.1).

9.3 Experimental Framework

As mentioned in the Section 9.1, the performance of the cued speech recognition model is evaluated on two single-subject continuous cued speech corpora described in Section 2.2, i.e., the *French cued speech dataset* [48] and the *British English cued speech database* [101]. In our experiments, on both databases, we employ 10-fold cross-validation, where 80% of each fold is used for training, 10% for validation, and 10% for testing.

Concerning system implementation, the 3D hand-pose network is trained using the Rendered HandPose dataset [203], which is a large-scale dataset of 3D hand poses created using synthetic hand models [203]. This dataset employs 3D human models with corresponding animations from Mixamo 2 [204]. In addition, the software Blender 3 [205] is utilized for the image rendering. The dataset consists of 41,258 images for training and 2,728 images for inference at a resolution of 320×320 pixels. The network is trained for 150 epochs through the Adam optimizer [141], with a batch size of 64, an initial learning rate of 0.001, and exponential decay. The weights of the linear layers are determined using Kaiming He initialization [188]. Further, to extract appearance features of the hand and mouth regions, we utilize the 3D ResNet-34 model [129]. The model is trained using stochastic gradient descent with momentum set to 0.9. The initial learning rate is set to 0.1 and is decreased by a factor

of 0.001. We perform 500 full passes over the data with a mini-batch size of 256 frames. For the sequential module, we utilize a TDS convolutional encoder consisting of two 3-channel, three 5-channel, and six 7-channel TDS blocks, each having a kernel size of 3×1 . In addition, we evaluate our method against various sequence models, which vary in the type of encoder used. To assess our system, we employ a one-layer LSTM encoder and a one-layer GRU encoder, each having 256 hidden units. Moreover, we utilize a Transformer encoder with a hidden dimensionality of 512.

Further, we evaluate our system against our previously introduced multi-modal system [94], where deep learning based feature extraction from three visual streams, i.e., handshape, lip, and hand position, is combined with a sequence learning model relying on TDS encoder and an attention-based convolutional decoder for cued speech phonetics prediction. Hand region is deduced through the HDTC model (see also Section 3.2), while hand position features are obtained using the hand coordinates relative to the lip location. The lip region is extracted using a cascade object detector based on the Viola-Jones algorithm [52].

Training of all sequence learning models is performed using the Adam optimizer [141] with a learning rate of 0.003, which is decaying by a factor of 0.85. Additionally, a batch size of 128 is utilized. We utilize a dropout rate of 0.1 and apply label smoothing [142] at a rate of 0.1. For inference, we utilize the beam search approach [158] with beam width equal to 3. The PyTorch framework [143] is utilized for the implementation of our system, and its training is performed with GPU acceleration.

9.4 Experimental Results

Table 9.1 reports the performance of our introduced approach for continuous cued speech recognition from videos. There, the PER (%) acquired on both cued speech corpora by the sequence learning model that operates on different feature stream combinations and relies on the TDS convolutional encoder and CTC decoding, is shown. It is apparent that the best results are achieved when all feature streams are concatenated, showcasing the benefit of incorporating multiple feature representations into the cued speech recognition system.

Regarding the different feature combinations taken into consideration in Table 9.1, we find that removing hand positional embeddings yields the poorest PERs on both datasets, indicating the significance of these embeddings in cued speech recognition. Notably, PERs

on both datasets are much higher when hand and mouth appearance representations are substituted by the corresponding skeleton data than their combination. This demonstrates that skeletal data are descriptive representations that convey significant information that might enhance the corresponding appearance representations; thus, it is imperative to employ them in tandem. Furthermore, when added to the fusion module, the 3D hand skeleton outperforms the corresponding 2D hand skeleton, suggesting that it is a robust representation. Adding it on top of the other streams also improves system performance. Additionally, it can be seen that the 2D mouth keypoints representation as additional mouth articulatory information reduces PER on both datasets. This is primarily because OpenPose detects face keypoints more reliably than the hand joints that are often occluded. Finally, it can also be observed that the two datasets' PERs differ significantly, most likely due to the limited size of the British English set.

Further, we examined the performance of our model under multiple variations in the appearance feature learner and the number of TDS blocks in the TDS convolutional encoder. Specifically, we replaced the 3D CNN with a 2D CNN (ResNet-18 [136]) for extracting the appearance features of the hand and mouth ROIs. This model uses 3×3 convolution kernels, downsampling with stride 2, and is pre-trained on the ImageNet corpus [118]. This change significantly degraded PER, by over 2% absolute (from 20.25% to 22.74% PER) for

Table 9.1: Phoneme error rate (PER, %) on the French and British English cued speech datasets, employing various feature stream combinations in conjunction with the sequence learning model of Section 9.2.2.

Feature Streams						Datasets	
Hand 3D-CNN (512-dim)	Mouth 3D-CNN (512-dim)	Positional 2D-CNN (64-dim)	2D Hand Skeleton (42-dim)	2D Mouth Keypoints (40-dim)	3D Hand Skeleton (63-dim)	French	English
✓	✓					38.50	44.11
✓	✓	✓				25.64	35.13
		✓	✓	✓		38.87	47.10
		✓		✓	✓	37.54	45.89
✓	✓	✓	✓			25.58	33.68
✓	✓	✓		✓		24.93	33.29
✓	✓	✓			✓	25.06	33.50
✓	✓	✓	✓	✓		22.17	32.91
✓	✓	✓	✓	✓	✓	20.25	32.58

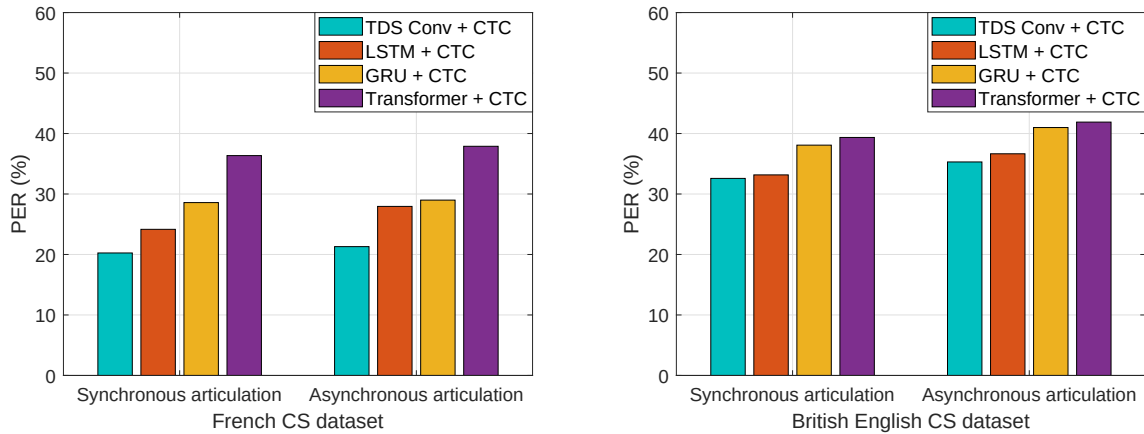


Figure 9.6: Comparative evaluation of various sequence learning models on both datasets in terms of phoneme error rate (PER, %) using all feature streams fused synchronously or with a fixed delay (asynchronously). The following notation is employed: cued speech (CS).

French cued speech and by about 3.5% (from 32.58% to 36.12% PER) for British English cued speech. For the TDS convolutional encoder, we increased the number of channels at the same receptive field from (3, 5, 7) to (10, 12, 14), (10, 14, 18), and (10, 10, 14), but in all cases we obtained worse PERs for both corpora.

Next, we explore the performance of the different sequence learning models on both cued speech datasets using all feature streams (1233-dim vectors), in Fig. 9.6. Due to the asynchrony between mouth and hand articulation, with the former occurring approximately one syllable earlier [194], we examine two feature fusion schemes: one that concatenates all features without taking into account this asynchrony, as in our system, known as “synchronous articulation” in Fig. 9.6, and another where the hand-related feature streams are artificially delayed by a fixed amount in time in the hope of better matching the mouth-related streams (referred to in the figure as “asynchronous articulation”). Specifically, we utilize a delay of 15 frames for the British English set and 12 frames for French cued speech (as suggested in [194]). The introduced sequence learning model (TDS convolutional encoder and CTC decoder), as shown in Fig. 9.6, performs well on both sets when features are concatenated directly without imposed time shift. It can also be seen that the Transformer encoder-based model produces the lowest results for both datasets, whereas the LSTM encoder outperforms the corresponding GRU model by a considerable margin, but still lagging our approach. Furthermore, when compared to synchronous fusion, imposing a temporal delay on the hand features consistently produces inferior results across all models.

Table 9.2: Evaluation comparison on both datasets in terms of phoneme error rate (PER, %).

Model	French cued speech ↓	British English cued speech (%) ↓
CNN-HMM [48]	38.00	-
Att-TDS-CNN [94]	29.12	36.25
Cross-modal CNN-BiLSTM [201]	25.80	35.10
Cross-modal CNN-Transformer [202]	24.90	33.60
Ours	20.25	32.58

In addition, we achieve considerable improvements on both datasets when we compare the best findings of Table 9.1 to our earlier work [94]. In particular, we obtain an 8.87% absolute PER reduction (from 29.12% to 20.25%) for French cued speech and a 3.67% one (from 36.25% to 32.58%) on British English cued speech. These advancements can be attributed to the redesign of both visual feature extraction and sequence learning modules of our system. Indeed, it is deduced from Table 9.1 that the considerably richer feature representation introduced here outperforms the appearance and hand positioning features that were previously employed. Additionally, the sequence learning model that has been introduced outperforms our previous model, which was based on the TDS convolutional encoder and attentional convolutional decoder. For French cued speech, this results in a PER reduction of 3.48% (from 29.12% to 25.64%), and 1.12% (from 36.25% to 35.13%) for British English cued speech, revealing the power of CTC decoding in cued speech recognition. It should be noted that these results pertain to the combination of appearance and hand positional features, as only these were taken into consideration in [94]. Finally, in Table 9.2 we compare the introduced with the state-of-the-art cued speech recognition schemes in the literature. As it can be observed, our model outperforms the state-of-the-art result obtained in [202], yielding 19% relative error rate reduction (4.66% absolute). Note that the system of [202] adopts a Transformer based cross-modal mutual learning framework that operates on three feature streams, i.e., mouth and hand shape, as well as hand position.

9.5 Conclusions

In this chapter, we explored the integration of multiple representation streams into a state-of-the-art deep-learning-based sequence learning model for the recognition of cued speech from upper body videos. In particular, our cued speech recognition system relied on spatio-

temporal features that incorporate hand positioning embeddings and CNN based appearance features of the mouth region and signing hand, as well as 2D skeletal data of the hand and lip region derived from the OpenPose library. In addition to the latter, we also included 3D skeletal representations of the signing hand using a 2D-to-3D hand-pose regression model. Subsequently, the latter representations were imposed on early fusion (simple concatenation) and propagated to a TDS convolution block structure based encoder, followed by CTC decoding, without explicit stream synchronization, for phonetic sequence prediction. We investigated the contribution of the above to recognition, evaluating our model on a French and a British English video dataset. The experimental results revealed that the inclusion of skeletal data, in particular the derived 3D skeletal data of the hand, in the feature fusion module improved the cued speech recognition when added to all other spatio-temporal data streams. The evaluation performed on two cued speech datasets demonstrated that the introduced model outperformed other sequence learning architectures and surpassed the state-of-the-art in the field.

Chapter 10

Real-Time Sign Language Recognition for a GSL Education Platform

10.1 Introduction

Learning an SL as L2 constitutes a difficult and time-consuming process for both students and their instructors [206]. Crucially, students need frequent and consistent feedback on their SL productions during learning. However, their instructors may not always be available to provide such feedback. This fact leads to a need for the development of appropriate automated systems that may assist with self-monitoring and objective assessment in SL L2 acquisition. Not surprisingly, there has been an increasing research activity to address such needs [207–214]. However, most such systems require “enrolled” signers (known to the system) and/or utilize specialized visual depth sensors or gloves for signing data input.

Motivated by the above and the absence of learning tools in GSL, we have recently concluded the SL-ReDu project [34], which aimed to pioneer a GSL learning and assessment system in an attempt to address the need for standardized L2 teaching. For this purpose, we have developed the SL-ReDu prototype system [168, 215, 216]. This system includes the theory presentation of appropriate GSL learning material [12, 215] and supports both self-monitoring and objective assessment of receptive and productive GSL learning (see also Figure 10.1). Productive learning relies on a GSL recognition module that operates in an SI fashion on video input by a typical web-cam, providing binary feedback regarding on the correctness of the signed productions.

In this chapter, we present the SL-ReDu prototype system with emphasis on the inte-

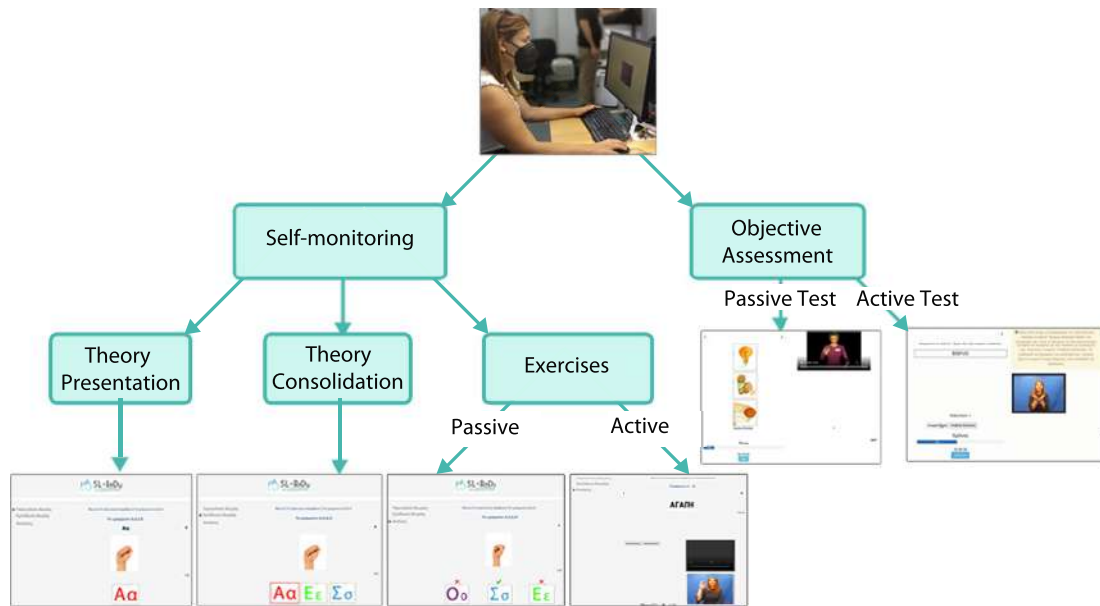


Figure 10.1: Illustration of the SL-ReDu prototype system design.

grated Greek SLR technology, which should be efficient/“light-weight” enough to allow such recognition to occur with small latency, accommodating three recognition tasks, namely: (i) isolated signs within a large-scale vocabulary (a vocabulary size of 337 GSL signs is supported); (ii) continuous signing (within a set of 695 unique GSL phrases); and (iii) continuously fingerspelled letter sequences (with a vocabulary of the 24 alphabet letters). To this end, our GSL recognition module in the case of isolated SLR is based on a 3D-CNN, namely the ResNet2+1D network [85], whereas in the case of continuous signs and continuous fingerspelling it relies on an efficient 2D-CNN, namely MobileNet [128], in conjunction with a BiLSTM encoder [78]. In particular, for isolated SLR, the system relies on the 18-layer ResNet2+1D model, also discussed in Section 4.2.3. During testing, we also incorporate a class-based Language Model (LM) into the isolated SLR system to further boost system performance. In the case of continuous signing, the SLR module consists of two parallel paths that are appropriately fused: The first involves a visual feature learner applied on each video frame, which adopts the 2D-CNN MobileNet architecture that was discussed in Section 4.2.3, which is subsequently followed by a sequence model relying on a BiLSTM encoder for learning temporal relations. The second path involves a window-based 3D-CNN module for local context classification. To further boost system performance, we leverage a class-based LM in conjunction with the SLR module. Finally, for continuous fingerspelling, the SLR module is based on the combination of the 2D-CNN MobileNet and a BiLSTM. In the adopted ap-

proach, the CNN serves as the visual feature learner of each video frame, while the BiLSTM learns temporal relations. To enhance system performance, we integrate an RNN character-level LM within the beam search decoder [158].

Further, we report the successful completion of the evaluation campaign for the aforementioned prototype system, performed by both students and experts involved in GSL teaching and learning at UTH. The process resulted in both quantitative and qualitative feedback, harvested in the form of objective measurements from the platform log files, as well as from subjective questionnaires returned by the participants, where they expressed their views on the platform objectivity, validity, usability, and functionality, as well as the overall experience, user trust, and ease of navigating it.

The remainder of the chapter is structured as follows: The SL-ReDu prototype is described in Section 10.2; In Section 10.3, we overview the adopted GSL recognition approach, presenting the visual pre-processing stage (Section 10.3.1) and each model in more detail (isolated SLR in Section 10.3.2, continuous SLR in Section 10.3.3, and continuous fingerspelling in Section 10.3.4), followed in each case with a description of the model training data and adopted experimental framework, and concluding with corresponding experimental results. Further, in Section 10.3.5 we provide a summary of the best recognition results for all three GSL recognition tasks considered. Finally, the SL-ReDu prototype system user evaluation findings are discussed in Section 10.4; and the chapter is concluded in Section 10.5.

10.2 The SL-ReDu Prototype System

The SL-ReDu platform integrates our GSL recognition module, together with an appropriate human-machine interface, into an innovative environment for evaluating the signing performance of GSL learners. As shown in Figure 10.1, the system involves two main modalities: self-monitoring and objective evaluation of receptive and productive GSL learning, thus attempting to address the shortcomings of traditional practice and assessment procedures in GSL L2 learning. In the case of productive learning, the integrated SLR technology provides a binary correctness assessment of the articulated signs, ensuring an objective, fast, and reliable testing process. The prototype allows two different modes of operation for the SLR module, performing such recognition either locally at the learner's device or remotely at a server, thus enhancing user flexibility.

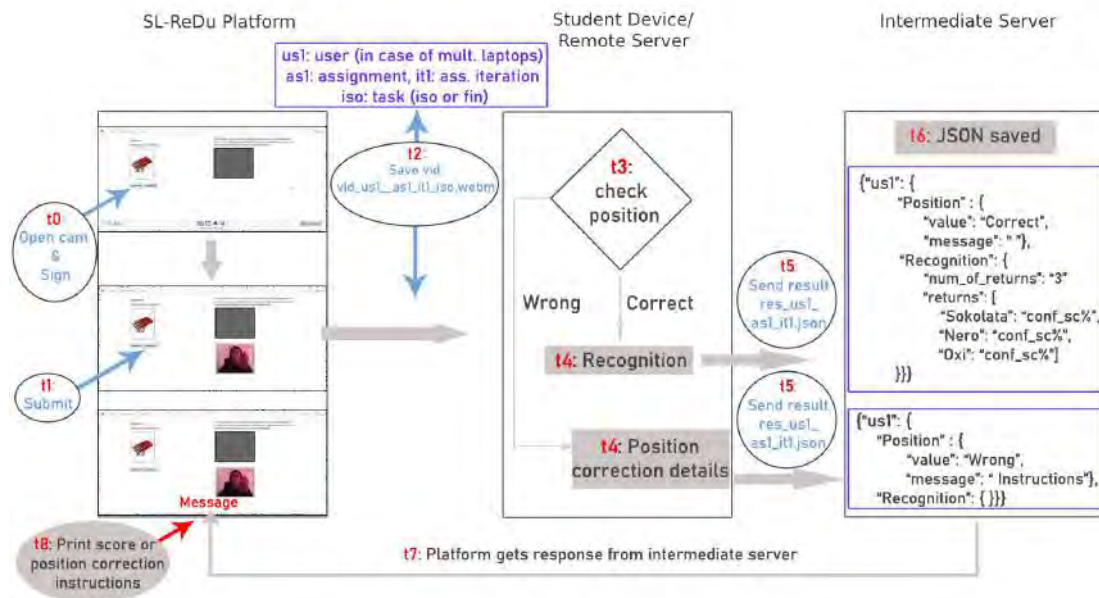


Figure 10.2: Architecture diagram illustrating the adopted SL-ReDu communication between the web application, recognition device (student device or remote server), and intermediate server to allow active SL production exercises by the learner.

Specifically, recognition of active signing can be performed at the learner's device, necessitating a number of technical requirements for such device and requiring the local installation of the module, but alternatively the module can also run standalone remotely at a server without requiring a higher-end learner's device. Appropriate communication between the GSL recognition device and the server hosting the web application is required by the selected architecture. As shown in Figure 10.2, such communication will occur between the server, the learner's device and/or a remote server, and an intermediate server (website) that serves as a "communication repository" to enable it. The communication protocol starts with the recording and submission of the learner's video containing the signed content. The file-name of the video is given by the web application and contains specific parameters such as user ID, assignment ID, assignment iteration and SL task (fingerspelling, isolated or continuous). Once submitted, the signing video is stored on the learner's device or on the remote server where the GSL recognition module is running locally. The module repeatedly pings a download folder to obtain the recorded video for processing, and at the same time a website path is cleared at the intermediate server. The recognition module uses appropriate computer vision algorithms (3D position estimation of the learner [111]) in a first phase to check the relative position of the learner with respect to the device camera's field of view and, in case of

incorrect positioning, to give instructions for placement rectification (see also 10.3.1). If no problems are detected, the video file is further processed and subjected to GSL recognition by our pipeline for extracting visual features and statistical recognition models corresponding to the specified GSL task. In both cases, the results of this module are uploaded to the intermediate server in the form of a JSON file containing both positioning and recognition result information. The corresponding filename is predefined to reflect the name of the processed video file so that the web application can access the results by pinging the intermediate server.

The learner's video, which includes the signed information, must be recorded and submitted before the intended communication protocol may begin, as seen in the figure. The web application predetermines the video filename and provides information about the user, assignment, iteration, and SL job (fingerspelling, isolated, or continuous). Following submission, the signature video is stored locally by the GSL recognition module on the learner's device or remotely on the server. The module clears a webpage route at the intermediary server while continuously pinging a download folder in order to finally obtain the recorded video for processing. In order to inspect the learner's relative position with respect to the device camera's field of view, the recognition module first uses appropriate computer vision algorithms (3D pose estimation of the learner [111]). If the placement of the learner is incorrect, the module provides instructions for placement rectification (see also Section 10.3.1). In the event that no issues are found, our visual feature extraction pipeline and statistical recognition models that are appropriate for the given GSL assignment are used to process the video clip further and subject it to GSL recognition. In both scenarios, the module's output is sent to the intermediate server as a JSON file that contains details on the recognition outcome as well as placement. The web application may obtain the results by pinging the intermediate server since the matching filename is predefined to mirror the name of the processed video file.

10.3 Overview of the GSL Recognition Approach

We next introduce an overview of our GSL recognizer, first discussing its overall architecture, followed by a more detailed presentation of its visual front-end pre-processing pipeline and the specific models for each of the three GSL recognition tasks.

The SLR algorithmic flow-chart is provided in Figure 10.3. It first employs the MediaPipe

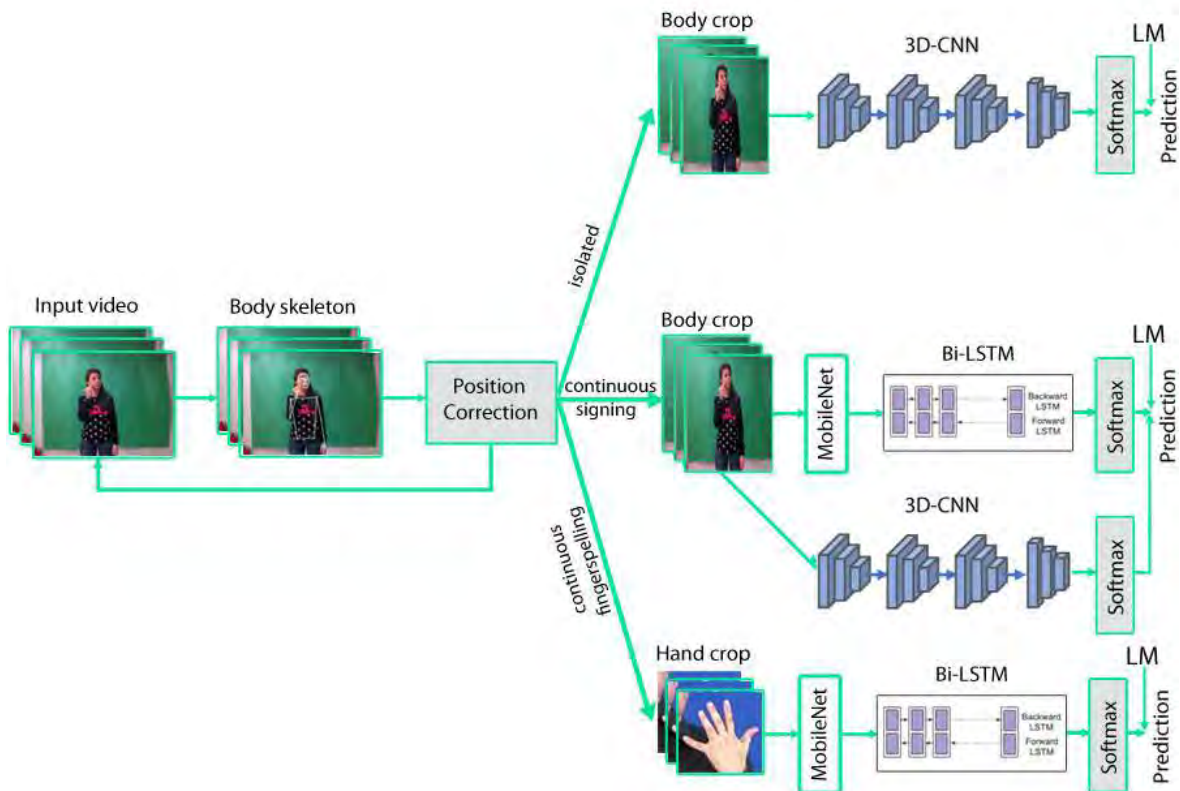


Figure 10.3: Algorithmic flow-chart of the GSL recognition module of the SL-ReDu platform: It first employs the MediaPipe framework [111] estimating the signer’s 3D pose to ensure correct positioning with respect to the camera and prompt for signer repositioning if necessary. Subsequently, the signer’s video is further processed for ROI extraction and then fed to a suitable recognition model depending on the GSL recognition task (isolated signs, continuous signs or continuous fingerspelling).

framework [111], estimating the signer’s 3D pose to ensure correct positioning with respect to the camera and prompt for signer repositioning if necessary. Subsequently, the signer’s video is further processed for ROI extraction, and it is then fed to a suitable recognition model depending on the GSL recognition task at hand (isolated signs, continuous signing, or continuous fingerspelling). An LM is also incorporated for each task to improve recognition performance. Details of each model are provided in Section 10.3.2 for isolated SLR, Section 10.3.3 for continuous SLR, and Section 10.3.4 for continuous fingerspelling, followed by corresponding experiments to benchmark GSL recognition performance.

10.3.1 Visual Pre-Processing

As illustrated in Figure 10.3, the SL-ReDu recognition module commences with the detection of the signer in order to inspect the relative position with respect to the field-of-view of the device camera, providing instructions for rectification in case of wrong positioning (e.g., occluded manual articulators). The latter functionality is necessary for the prototype system, as the user may not be correctly positioned within the camera field-of-view. For signer detection and tracking, we employ the MediaPipe holistic framework [111] (see also Section 4.2.1 and Figure 10.4).

When the articulators are occluded, namely the detected landmarks of the two hands, face, and upper torso are missing for more than a specific number of frames (12 frames), the system prompts the signer to reposition, informing of incorrect hands, face, and body positioning with respect to the camera. If the positioning is acceptable, the detected landmarks are used to generate the ROI for the next stage, i.e. the visual representation generation (see also Figure 10.3). Particularly, in the case of isolated and continuous signing, where multiple articulators may be involved in signing, the entire upper body is cropped in order to produce the ROI. This is then normalized to the subsequent CNN input layer size, i.e., 256×256 pixels of ResNet2+1D [85] in the isolated GSL recognition case and 224×224 pixels of the MobileNet [128] CNN used in the continuous GSL recognition case, as discussed later. In the case of fingerspelling though, where typically one hand constitutes the sole articulator, the ROI consists of the signing hand only (see also Figure 10.3), which is determined based on the motion of the landmarks (3D skeletal keypoints) of each hand in the video. The ROI is then normalized to the input layer size of the MobileNet CNN (i.e., 224×224 pixels), which

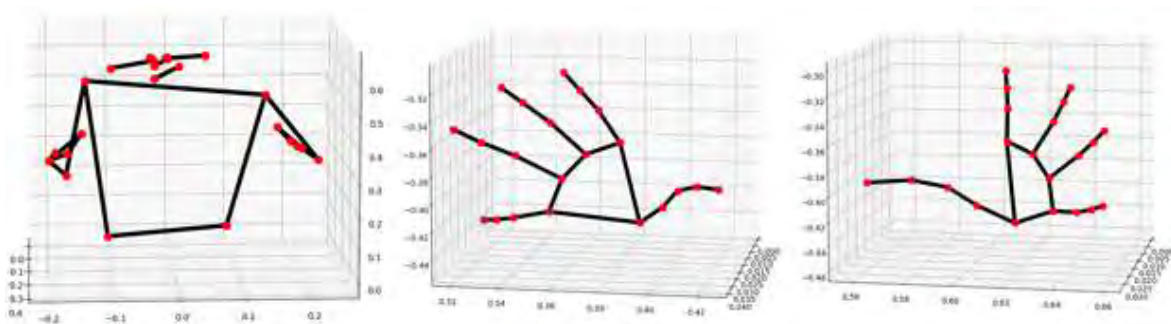


Figure 10.4: Examples of (a) 3D body, (b) left hand, and (c) right hand skeleton representations, produced by the MediaPipe holistic pipeline [111].

constitutes the visual front-end of the continuous fingerspelling sub-system. Note that we use the estimated landmarks exclusively for ROI extraction, thus minimizing the impact of occasional MediaPipe failures [217].

10.3.2 Recognition of Isolated GSL Signs

Following the general overview of our SLR module, we initiate its more detailed presentation for the isolated GSL case. Specifically, we discuss the model details, followed by its training data and conducted experiments that benchmark its performance.

The Recognition Module

A 3D-CNN is employed for isolated sign recognition (see also Figure 10.3), namely the 18-layer ResNet2+1D model [85] with weights trained on the Kinetics dataset [137]. We also pretrain the model on the Chinese SL dataset [162] to further enhance recognition performance (see also Section 4.2.3). To improve the model, we replace the ReLU activation function with the SiLU one [166] (see also Section 6.2.1). In addition, to boost system performance, we integrate a class-based LM during testing. Specifically, information regarding the sign thematic category that the user is currently navigating is exploited during decoding, reducing the number of candidate predictions. Thus, we keep only predictions belonging to the same category with the thematic category, picking the best probability score prediction. Note that the isolated SLR model serves a large-scale vocabulary of size 337, split to 33 thematic categories. Model training (fine-tuning) is conducted via the Adam optimizer [141] with initial learning rate set to 0.001 and weight decay 0.0001. During training, the cross-entropy loss is used with label smoothing [142]. The mini-batch size is fixed to 16.

Data and Experimental Framework

Model development and evaluation is based on our GSL dataset, collected as part of the SL-ReDu project, containing studio data from 21 signers (see also Section 2.3.2). In the case of studio data, each signer (native in GSL) performed the 337 signs 3 consecutive times, resulting in a total of 21,231 videos. In the case of non-studio data, signers (both native and non-native in GSL) performed from 1 to 3 times a number of signs (from 18 to 90 signs) resulting in a total of 4,300 videos. Thus, the dataset consists of 25,531 signing videos in total.

Based on these data, we adopt two experimental frameworks:

- An MS setting, where data from all signers are used for both training, validation, and test sets. Specifically, we employ ten-fold cross-validation, with 80% of all videos used for training (20,425), 10% for validation (2,553), and the remaining 10% for testing (2,553).
- An SI setting, where test signers have not been employed during training, remaining unseen. Here, we conduct our testing only on non-studio video data, since this is the scenario of interest in the deployment of the SL-ReDu platform (see below). Specifically, we consider as test signers the 14 subjects that have recorded both studio and non-studio data (testing on the latter), employing 2-fold cross-validation with each fold containing 7 “unseen” test signers, while the model is trained on all the others.

Note that we are interested primarily in SI recognition, since learner users of the SL-ReDu platform are typically “unseen” during GSL model training. Moreover, we focus on non-studio video data, since the platform evaluation took place in non-studio environments with cluttered backgrounds and ordinary lighting conditions. In addition to the above experimental paradigms, GSL models are also trained to be used by the SL-ReDu platform. For this purpose, we allocate about 90% of the available videos to training (22,978 videos) about 10% to validation (2,283 videos), and the remaining ones to testing (270 videos).

Recognition Results

In Figure 10.5, we report the recognition performance of the isolated GSL model on the aforementioned dataset under both MS and SI training/testing paradigms. Results are reported in word (sign) accuracy (WAcc, %). Specifically, we compare our 3D-CNN model with and without the inclusion of the class-based LM. Our model achieves 98.54% WAcc in the MS case and 95.32% WAcc in the SI case when the class-based LM is included in decoding. Clearly, the inclusion of the class-based LM boosts model performance (compare to the results without it). Finally, it is obvious that accuracies in the MS case are higher than those in the SI case, which is expected, since in the SI case we use only non-studio data from unseen signers for testing. Nevertheless, the WAcc remains satisfactory in isolated SLR, showing the potential of utilizing the module in learning platforms like SL-ReDu.

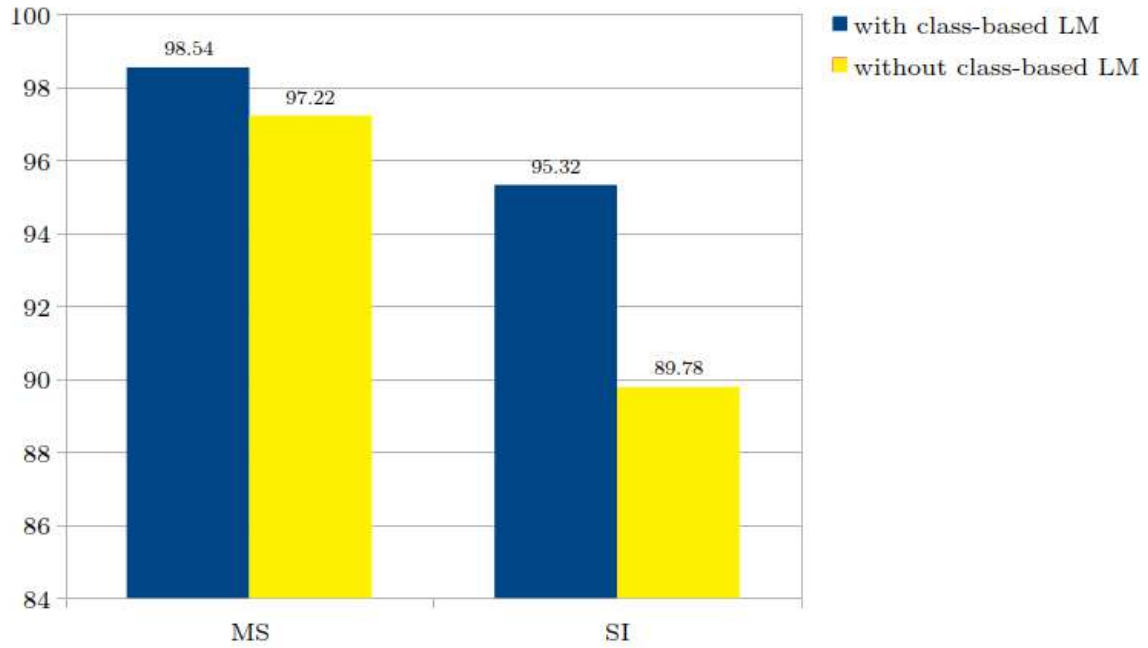


Figure 10.5: Isolated GSL recognition performance (in WAcc, %) with the integration of the class-based LM, as well as without it, under both MS and SI experimental frameworks.

10.3.3 Recognition of Continuous GSL Signing

We next proceed with a more detailed presentation for the case of continuous SLR. Specifically, we discuss the model details, followed by its training data and conducted experiments that benchmark its performance.

The Recognition Module

The continuous GSL recognition module composes of: (i) a visual feature learner of each video frame, which adopts the 2D-CNN MobileNet architecture [128], followed by a sequence model relying on a BiLSTM encoder [78] for learning temporal relations; and (ii) a window-based 3D-CNN module for local context classification. Specifically, as it can be seen in Figure 10.3, sequences of video frames are cropped using the motion of the 3D skeletal keypoints extracted via the MediaPipe regression model [111], resulting in sequences of images including the signer's body, which are then fed to the 2D-CNN model, namely the MobileNet CNN, which is pre-trained on the ImageNet corpus [118], yielding 1024-dim feature maps. These features are then passed to a linear projection layer for size reduction, resulting in 512-dim features. Subsequently, a three-layer BiLSTM encoder is employed with 512-dim

hidden states, followed by a softmax layer, yielding the probabilities distribution for aligning the signing videos to word sequences, modeled via the CTC decoding model [81]. The model's linear projection layer is jointly trained with the BiLSTM encoder. Model training is conducted using the Adam optimizer [141] with initial learning rate equal to 0.001, decayed by a factor of 0.1 if the validation score remains consistent for 9 steps. In addition, a dropout rate of 0.1 is used, and the mini-batch size is fixed to 16. During testing, beam search decoding [158] with beam width 7 is adopted. Finally, we also add a label smoothing [142] term equal to 0.2, in order to penalize low-entropy distributions.

In parallel, we split the initial sequence of video frames into overlapping short sequences (every 15 frames) of the same length using a local causal window of 60 frames, such that each short sequence includes 60 consecutive frames with the last being the target frame (properly pre-padding the video beginning). Subsequently, we apply the 18-layer ResNet2+1D model [85] of Section 10.3.2 (vocabulary size of 337 GSL signs) to these short video frame sequences producing a list of predictions. From this list we remove the predictions that appear less than 2 times, as well as duplicates, resulting in a list of P predictions. Note that the 18-layer ResNet2+1D model is trained on the data of Section 10.3.2, using the appropriate experimental framework (MS or SI).

During inference, we employ a “constrained” beam search decoder [158], forcing the recognition process to generate sequences with the highest probabilities under the constraint of needing to include a set of provided tokens, namely these P predictions extracted from the 18-layer ResNet2+1D model. In more detail, during testing, a class-based LM is integrated into the continuous GSL recognition module decoder, which involves 16 different thematic categories of sentences (695 unique sentences). Thus, we exploit information regarding the thematic category to which the input signing video's label (sentence) belongs, keeping only paths from the list of top- b paths of the last beam search decoding step that correspond to sentences (candidate sequences) of the target category. Then, we select from the set of the resulting paths the best-score one that satisfies all constraints, namely the first one in score that includes the set of these P predictions extracted from the 18-layer ResNet2+1D model. If there is no candidate sentence that involves all P predictions, then we select the first one in score that includes $P-1$, with the process repeating until there is a path satisfying at least one constraint. In case there is no candidate sentence that involves at least 1 from the P predictions, then we select the best one from the resulting paths. In addition, if there exists

no path in the list of top-b paths of the last beam search decoding step that corresponds to a sentence of the target class, then we select the best-score path from the top-b paths.

Data and Experimental Framework

As already discussed in Section 2.3.1, continuous GSL signing data in both studio and non-studio conditions have been collected. Specifically, 799 unique sentences (with a vocabulary size of 294 glosses) were divided to 3 separate groups. For the studio data collection, 21 native signers performed the content of only one group once, thus resulting in a total of 5,930 videos. For the non-studio data, each of the 3 groups was further split to 4 sub-groups, resulting in 12 total groups of phrases. Each of 17 signers (both native and non-native) performed one only group once, generating 906 videos in total. The whole set of unique sentences has been recorded at least 8 times each, resulting in 6,836 videos in total. Note that, for our system development, we have selected only the sentences consisting of up to 3-gloss sequences, which correspond to 4,762 signing videos (695 unique sentences with a 255-size vocabulary).

Similarly to Section 10.3.2, for continuous SLR evaluation we adopt two experimental frameworks:

- In the MS setup, we employ 10-fold cross-validation, with 80% of videos used for training (3,810 videos), 10% for validation (476 videos), and 10% for validation (476 videos).
- In the SI setup, for testing, only non-studio data are used of the 14 signers that produced both studio and non-studio data. For this purpose, we employ 14-fold cross-validation with each fold containing 1 “unseen” test signer, while the model is trained on all others.

Finally, to produce the GSL model that is integrated to the SL-ReDu system, we allocate about 90% of the available videos to training (4,252 videos), about 10% to validation (476 videos), and the remaining few (34 videos) for testing.

Recognition Results

Our continuous GSL recognition results are displayed in Figure 10.6. As expected, compared to isolated SLR WAcc (see Figure 10.5), it is natural that performance suffers at the level of sequence accuracy (SAcc, %), due to the sequence / continuous nature of the recognition task. Our recognition results demonstrate that the employment of LMs boosts per-

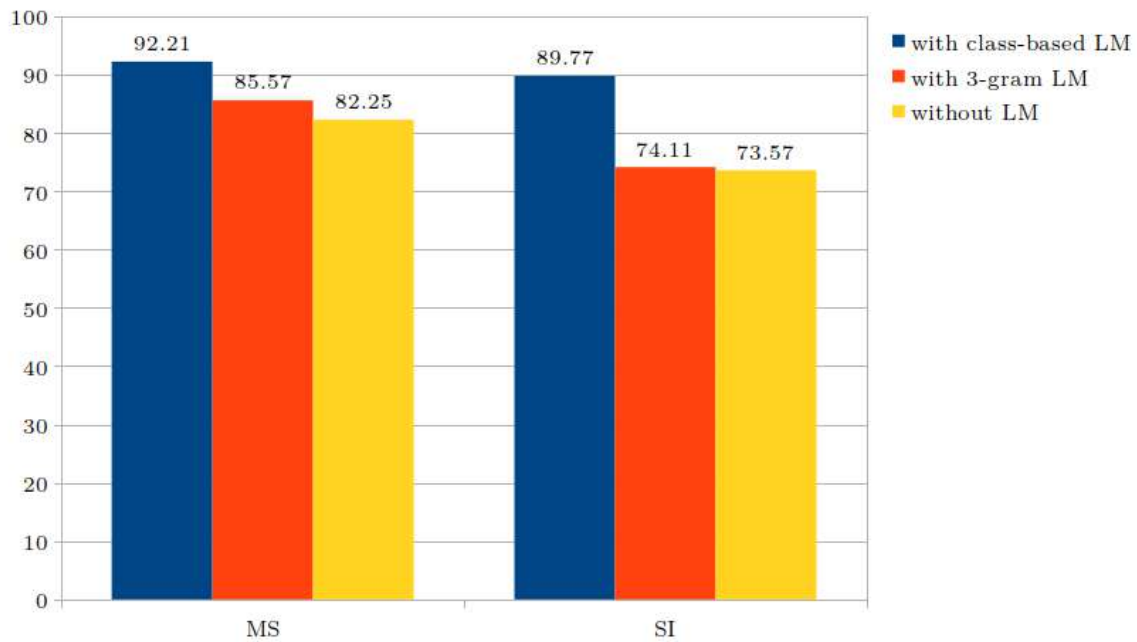


Figure 10.6: Continuous GSL recognition performance (reported in terms of sentence accuracy (SAcc, %)) when no LM is used, as well with the integration of the class-based LM or the 3-gram LM, under both MS and SI experimental frameworks.

formance. Two such LMs are considered: In addition to the class LM mentioned above a statistical 3-gram LM [218,219] is also employed. This model is trained using the modified Kneser-Ney smoothing strategy [218,220] on a very small set of 910 Greek gloss sentences, containing a total of 4,961 glosses within a 963-size vocabulary. As shown in Figure 10.6, the best results are obtained with the class-based LM, yielding 92.21% SAcc in the MS case, and (as expected) a lower SAcc of 89.77% in the SI case.

10.3.4 Recognition of Continuous GSL Fingerspelling

We finally provide a more detailed presentation for the task of continuous fingerspelling of letter sequences in GSL. Specifically, we discuss the model details, followed by its training data and conducted experiments that benchmark its performance.

The Recognition Module

A CNN-BiLSTM combination is employed for recognizing continuously fingerspelled sequences of the 24 Greek alphabet letters. In the adopted approach (see also Figure 10.3),

the CNN serves as visual feature learner of each video frame, while the BiLSTM learns their temporal relations. Specifically, the CNN uses the MobileNet architecture [128], pre-trained on the ImageNet corpus [118]. Feature maps are generated by taking the output of the last fully-connected layer, yielding 1024-dim features. These are then fed to a linear projection layer for size reduction, resulting in 512-dim features. Subsequently, a two-layer BiLSTM encoder [78] is employed with 512-dim hidden states, followed by CTC decoding [81] for letter sequence prediction. The model's linear projection layer is jointly trained with the BiLSTM. To enhance system performance, we integrate an RNN character-level LM [221] in the beam search CTC decoder, following the formula:

$$P_{final} = a \times P_{model} + b \times P_{LM}, \quad (10.1)$$

for fusing their scores, with $a = 0.5$ and $b = 1.0$. The RNN character-level LM [221] comprises a 2-layer LSTM model [77] with hidden-state size of 256 and embedding dimensionality of 64, followed by a fully-connected layer and softmax generating the probability distribution. Each LSTM layer is followed by dropout of 0.2. Training is conducted using the cross-entropy loss function with label smoothing [142]. The vocabulary comprises 24 letters and the model is trained using a dictionary consisting of 46,928 Greek words of 386,703 characters in total.

Training of the GSL fingerspelling recognition model is conducted using the Adam optimizer [141] with initial learning rate equal to 0.001, decayed by a factor of 0.1 if the validation score remains consistent for 9 steps. In addition, a dropout rate of 0.1 is used, and the mini-batch size is fixed to 16. Finally, during inference, beam search decoding [158] is adopted with beam width 2.

Data and Experimental Framework

As detailed in Section 2.3.1, continuous fingerspelling GSL data in both studio and non-studio conditions have been collected. Specifically, in the case of studio data, 21 informants signed once the 24 GSL alphabet letters in isolation, as well as a set of 50 fingerspelled words composed of 3-6 letters, which was unique to each signer. The process yielded 1,554 videos in total. In the case of non-studio data, 12 signers produced once the 24 Greek alphabet letters in isolation, as well as 50 fingerspelled words (unique to each signer) composed of 4-5 letters. Further, 7 informants from the group signed an extra 16 words (common to all)

composed of 3-7 letters, and 3 signers produced an extra 71 words (in total) of 4-5 letters each. This process resulted in 1,071 videos of non-studio quality. Note that in both studio and non-studio recordings (a total of 2,625 videos), each informant has signed each letter at least 4 times (in isolation or in a sequence). Similarly to Sections 10.3.2 and 10.3.3, for the SLR evaluation of continuous fingerspelling we adopt two experimental frameworks:

- In the MS case, we use ten-fold cross-validation, with 80% of all videos being used for training (2,101 videos), 10% for validation (262 videos), and the remaining 10% for testing (262 videos).
- In the SI case, for testing, we employ only the non-studio data generated by 12 informants. For this purpose, we use 12-fold cross-validation, with each fold containing one test signer (only non-studio video data are used for testing), while the model is trained on all data of all remaining signers.

Finally, as in Sections 10.3.2 and 10.3.3, in order to produce the GSL model that is integrated to the SL-ReDu system, we allocate about 90% of the available videos to training (2,289 videos), about 10% to validation (262 videos), and the remaining few to testing (74 videos).

Recognition Results

Concerning continuous fingerspelling, it is natural that performance suffers at the WAcc level, since letter recognition errors (including insertions and deletions) accumulate at the word level, especially for longer letter sequences. In Figure 10.7, we evaluate the performance of our fingerspelling SLR system when no LM is incorporated, as well as when the 5-gram statistical LM [218, 219] and the RNN LM [221] are incorporated under both experimental paradigms, namely MS and SI, in terms of WAcc (%) and letter accuracy (LAcc, (%)). As it can be readily seen, integration of either LM helps SLR performance (over the no LM case), with the RNN LM resulting in higher WAcc and LAcc than the statistical one in both MS and SI settings. This is not surprising, given that neural models address some of the issues of traditional statistical LMs, enabling conditioning on increasingly large context sizes with only a linear increase in the number of parameters and supporting generalization across different contexts.

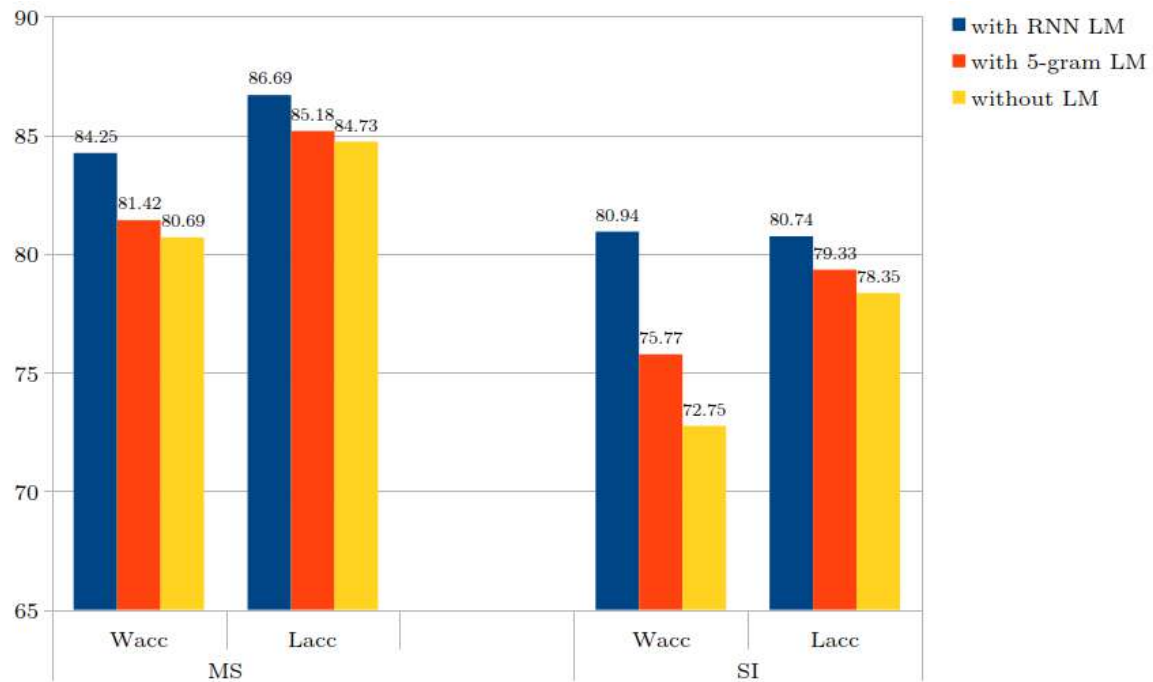


Figure 10.7: GSL recognition performance for continuous fingerspelling (reported in terms of word and letter accuracies, %) when no LM is used, as well as with the integration of the RNN LM or the 5-gram LM, under both MS and SI experimental frameworks.

10.3.5 Summary of GSL Recognizer Performance

To help summarize the above reported performance of the GSL recognition models of Sections 10.3.2, 10.3.3, and 10.3.4, we produce Table 10.1, where results in the appropriate accuracy metrics are provided for the three recognition tasks (isolated signing, continuous signing, and continuous fingerspelling) when employing the best language modeling techniques. As noticed earlier, performance in the MS case is superior in all tasks, but for the SL-ReDu application it is the SI setting that is of interest. In that case, results remain at satisfactory levels, with the isolated task performing significantly better than the continuous ones.

10.4 GSL Education Platform Evaluation

In this section, we present the evaluation results of the SL-ReDu prototype system. We first present objective metrics extracted from the log files of the SL-ReDu platform during the evaluation campaign for the GSL production exercises (“active” type), and subsequently

Table 10.1: Summary of GSL recognition results of the previous experimental sections for the three tasks considered, under both the MS and SI experimental frameworks, reported in the appropriate accuracy metrics.

GSL recognition task	Accuracy metric (%)	Experimental framework	
		MS	SI
Isolated signing	WAcc	98.54	95.32
Continuous signing	SAcc	92.21	89.77
Continuous fingerspelling	WAcc	84.25	80.94
	LAcc	86.69	80.74

we report subjective evaluation results based on questionnaires filled by the evaluation volunteers.

10.4.1 Objective Assessment of GSL Production

The GSL production exercises in the SL-ReDu platform involve active signing by the learner in front of a computer with an RGB camera of sufficient quality (sustaining a 25 Hz frame rate and at least a 640×480-pixel frame resolution video). The GSL recognition module runs and provides feedback to the learner through the SL-ReDu platform, as well as feedback in case of incorrect user placement with respect to the camera. The evaluation of GSL production involves a large vocabulary of isolated signs (337 glosses), a large number of continuous signs (a set of 695 phrases with a 255-gloss vocabulary), as well as continuously fingerspelled letter sequences. A sufficient number of tests are available for each task for the evaluation participants to choose from (a total of 40 different tests), each including production exercises, specifically:

- 20 tests for isolated signs, each containing 5 exercises;
- 10 tests for continuous signs (phrases), each containing 6 exercises;
- 10 tests for continuous fingerspelling, each containing 4 exercises with letters and words.

The participants were allowed to sign for each exercise up to twice (the second time in case of negative feedback by the system) within the time duration constraints of the selected tests (one per task), and the system automatically graded their efforts providing cumulative test scores.

In total, there were 44 participants, namely 6 GSL experts, 36 A1-level students, and 2 A2-level students. Each participant carried out two tests for isolated signs (10 exercises total), one test for continuous signs (6 exercises), and one test for fingerspelling (4 exercises), performing 20 exercises overall (880 in total over the 44 volunteers). Results are summarized in Table 10.2, cumulatively and per GSL task (over all users). As it can be observed, among the 880 exercises in total, in the first user attempt, 818 GSL productions were recognized correctly by the system ($779 + 39$) and 62 were recognized incorrectly ($60 + 2$). In the second user attempt, 71 were recognized correctly ($56 + 15$), with the remaining 29 were recognized incorrectly. These involved 7 isolated signs, 10 continuous signs, and 12 fingerspelling exercises, showcasing that the phrases and fingerspelling recognition tasks are the most difficult ones due to their continuous nature. It is interesting to note that there exists a number of cases that the user has signed correctly, but the system has recognized an incorrect sign, while there exist two cases where incorrect user signing was recognized by the system as correct. Note that to enable tabulation, we had a GSL expert view the recorded videos of the produced signs to gauge their correct or incorrect formation.

In addition to the overall results of Table 10.2, Table 10.3 summarizes the SLR results for all three GSL tasks, also reporting them for the expert users subgroup (6 of the 44 users). Results are reported in WAcc (%) for the isolated and fingerspelling tasks and in SAcc (%) for the continuous task. As it can be observed, the recognition accuracy achieved by the expert

Table 10.2: Summary (over all 44 users) of the results of GSL production exercises in the system evaluation, depicted cumulatively and per task for both the first and the second (if necessary) sign production attempt.

Attempt (880 first, 100 second)		First attempt		Second attempt	
		System response		System response	
Task	User sign	correct	incorrect	correct	incorrect
all 3 tasks	correct	779	60	56	29
	incorrect	2	39	0	15
isolated	correct	405	16	24	7
	incorrect	1	18	0	3
continuous	correct	230	20	16	10
	incorrect	1	13	0	8
fingerspelling	correct	144	24	16	12
	incorrect	0	8	0	4

Table 10.3: Comparison of objective evaluation recognition results for each GSL recognition task, reported in word accuracy (WAcc, %) for the isolated and fingerspelling tasks, while in sentence accuracy (SAcc, %) for the continuous signing task. Results for the expert user group are also reported.

Task	all users	experts users only
isolated	94.94	98.41
continuous	89.60	100.00
fingerspelling	82.69	100.00

users is 100% in the continuous signing and fingerspelling recognition tasks, and 98.41% in the isolated sign task, surpassing the cumulative evaluation scores of the entire user population. This is likely due to the more careful and clearer signing performed by the GSL experts. It is also worth noting that the GSL recognition results of all 44 users in Table 10.2 are in close agreement with those of Table 10.1, which are (for the SI setting) 95.32%, 89.77%, and 80.94% for isolated, continuous signing, and continuous fingerspelling, respectively.

10.4.2 Subjective Assessment of the GSL Education Platform

At the end of the evaluation session, participants were given an anonymous subjective experience questionnaire to rate various aspects of the education platform in terms of ease of use, usefulness, design, and user confidence on a Likert scale with five discrete options ranging from one (poor) to five (excellent). At the end of the questionnaire, volunteers were also asked to provide additional opinions or comments at the bottom of the form.

The analysis of the subjective experience questionnaires submitted by the participants at the end of their evaluation sessions provided valuable input both in the form of numerical trends and in personal comments. A summary of the mean opinion score results (in the Likert scale) is depicted in Figure 10.8, where in addition to the mean, also shown are the median, maximum, and minimum scores returned by the 44 subjects for each of the eight questions of the subjective evaluation. In all cases the majority of the evaluation participants responded with the highest mark (“very much” - median value of 5). Further, for only one question, one or more answers came back as “low” (value of 2), being the lowest evaluation mark (see also Table 10.4). Based on these observations, the results of the subjective evaluation are deemed extremely satisfactory. Finally, 18 volunteers provided additional personal feedback in the

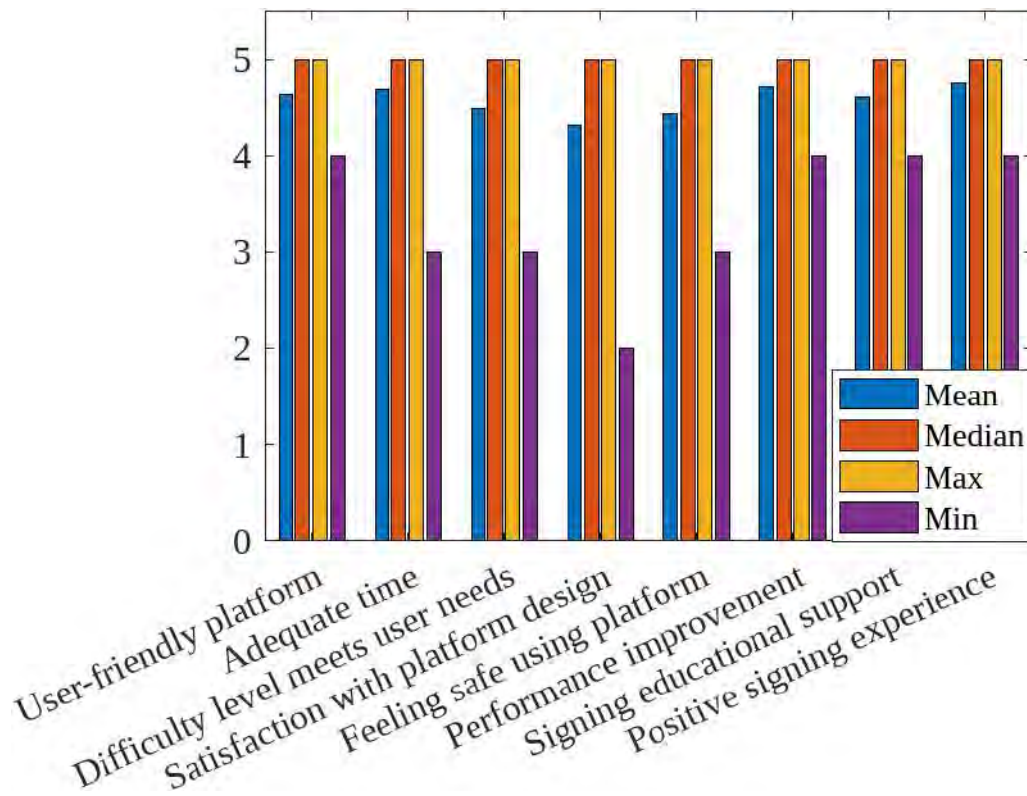


Figure 10.8: Summary of the subjective responses to 8 questions on the Likert scale over all 44 evaluation participants.

Table 10.4: Median, maximum, and minimum values of the subjective evaluation of the platform for each of the eight questions over all 44 evaluation participants.

Subjective question	Values of answers over all subjects		
	median	maximum	minimum
User-friendly platform	very much	very much	much
Adequate time	very much	very much	medium
Difficulty level meets user needs	very much	very much	medium
Satisfaction with platform design	very much	very much	low
Feeling safe using platform	very much	very much	medium
Using platform for performance improvement	very much	very much	much
Signing educational support	very much	very much	much
Positive signing experience	very much	very much	much

form of written comments at the bottom of the subjective evaluation form.

10.5 Conclusions

In this chapter, we presented the SL-ReDu education tool, which is built for both self-monitoring and objective evaluation of GSL perception and production. In particular, we overviewed the SL-ReDu prototype system, as well as the GSL recognizer that is integrated into it. The GSL recognizer is based on our research concerning visual tracking and feature extraction, the SLR modeling approach, as well as the linguistic models and training data described in Chapter 2. The developed GSL recognition module accommodates three recognition tasks, namely that for isolated signs, continuous signing, and continuous finger-spelling. Further, this chapter reports on the SL-ReDu prototype system evaluation campaign accomplished by real end-users. In particular, we presented the evaluation results, reporting objective assessments of the GSL production aspects of the system, as well as a subjective evaluation of the entire platform using a corresponding questionnaire. The obtained results are deemed extremely satisfactory, showcasing the success of the overall SL-ReDu project approach and the robust design of the prototype system, deeming it appropriate and well-suited for the education use-case.

Chapter 11

Conclusions and Future Directions

This chapter provides a concise overview of the contributions made in this Ph.D. Dissertation and highlights the importance of our research. In addition, this chapter examines potential directions for future study based on the current research findings.

11.1 Conclusions

This Thesis has led to various contributions, which are listed below with a brief summary of each:

- We collected and publicly released the SL-ReDu GSL studio corpus that constitutes a significant contribution to the SLR research community, with expected impact in both Greece and abroad. The recorded data surpass more than three times the other available pre-existing GSL dataset suitable for SLR, in both the number of recorded signers (21 informants) and its duration (almost 36 hours), while addressing three SLR tasks of large lexical content.
- We investigated various cutting-edge approaches, regarding both visual processing and feature extraction, as well as the classification and fusion methodology adapted to the SLR task. Specifically, we strived to enhance the visual module of the SLR system by exploring a number of efficient visual feature representations based on 2D/3D appearance and motion information of the signer, as well as 2D/3D skeletal data, and 3D pose and shape parameterization. In addition, we examined multiple single-stream sequence learning models with and without attention. Finally, we attempted to boost

the alignment module, experimenting with various decoding schemes and loss functions. We evaluated the developed algorithms on numerous isolated and continuous databases, demonstrating very satisfactory performance, which enabled us to proceed with the development of various SLR systems, as well as the GSL recognizer that was integrated into the SL-ReDu prototype system.

- Regarding isolated SLR, we explored the benefit of fusing two different modalities that operate on visual representations of appearance and human pose to capture the signing activity. Specifically, the introduced approach incorporates a novel version of the ResNet2+1D network, which integrates deformable spatial and temporal convolutions, operating independently on the RGB stream. In addition, an innovative ST-GCN unit that integrates modulated GCNs is applied to skeletal graphs relying on 3D joint-rotation parameterization, derived by the PIXIE 3D pose and shape regression model. The two modalities are fused through an ensemble module. The system was evaluated on two popular existing isolated SLR benchmarks, namely a Turkish and a Greek SLR corpus, as well as our recently recorded Greek isolated SLR corpus, surpassing the current state-of-the-art on one corpus and reaching competitive performance on the others in both MS and SI experimental paradigms.
- For the continuous SLR task, our research main contributions lie in the introduction of an innovative sequence learning model that relies on a window-based RNN module and a multi-head local Transformer encoder. The Transformer encoder incorporates relative position encoding and a Gaussian bias for localness modeling. To align the input frame sequence to the sequence of glosses we employ CTC decoding. In addition, in order to enhance the visual model, we combine the CTC loss with a visual alignment loss. The sequence learning model is applied to both RGB and optical flow streams, deeming the introduced approach bi-modal. The two modalities are further ensembled through CTC guiding alignment. We evaluated our model on two popular large-scale German CSLR benchmarks, as well as a Greek continuous SLR corpus, demonstrating superior performance on all three datasets.
- Further, we addressed the continuous fingerspelling recognition task, introducing a novel system that incorporates the 3D-CNN ResNet2+1D model for spatio-temporal video feature representation modeling. For sequence learning and alignment we em-

ploy a BiLSTM encoder coupled with CTC decoding. During training, in addition to the CTC loss function relying on the sequence module, we incorporate two auxiliary loss functions, i.e., a visual module CTC loss function that regards the distribution produced by the 3D-CNN based visual module, as well as a knowledge distillation loss, enhancing the visual representation capacity and the alignment between the visual and sequence modules. We evaluated our introduced approach on an ASL fingerspelling corpus, as well as the collected Greek fingerspelling corpus, reporting superior performance in both cases.

- We effectively tackled the automated recognition of cued speech from upper-body videos by integrating various visual feature streams into a cutting-edge deep-learning sequence learning model. In particular, our cued speech recognition system utilizes a combination of spatio-temporal features, such as 2D/3D-CNN appearance, hand positioning, and 3D hand skeleton that are learned using a TDS convolutional encoder. The system employs CTC decoding without the need for explicit stream synchronization. We highlighted how the inclusion of skeletal data to the feature fusion module benefits system performance. The assessment performed on two popular cued speech datasets showed that the proposed system outperforms alternative cued speech sequence learning architectures, surpassing the state-of-the-art in the field.
- We successfully developed and assessed a highly efficient GSL recognition system that enables efficient recognition with small latency. This system supports three types of recognition tasks: (i) that of isolated signs from a large vocabulary of 337 GSL signs, (ii) continuous recognition of signs from a collection of 695 distinct GSL phrases, and (iii) continuous recognition of fingerspelled letter sequences of the 24 Greek alphabet letters. The GSL recognition system was incorporated into the SL-ReDu prototype system. The system evaluation was highly successful, with both objective and subjective assessments yielding very satisfactory results. This confirms the efficiency of the proposed SLR approach and methodology, as well as the resilience of the designed platform, providing evidence that the utilization of automated technologies to assist L2 learning of GSL is feasible.

11.2 Future Directions

Real-time SLR systems are of paramount importance in enabling communication for deaf and hard-of-hearing individuals. Although there has been major progress in the field with assistive technologies improving their communication capabilities, there are still several challenges to overcome. To this end, there are various directions that can be investigated:

- The improvement of the SL-ReDu system, supplementing the GSL recognizer with the potential of providing feedback concerning the rectification of a sign in case of inaccurate SL productions. This requires the provision of adequate linguistic input that will consolidate additional knowledge on morphological patterns of GSL, as well as further exploitation of appropriate computer vision and machine learning methods.
- The development of additional novel SLR algorithms in the future, exploiting the collected SL-ReDu GSL corpus that could be utilized in multi-lingual SLR techniques and/or improved SLR model pre-training.
- Since the majority of deaf people have reading and writing difficulties, these aspects act as barrier to digital content access that is presented mostly in text form, having a detrimental interaction impact with the interface platform. For this reason, an interesting research direction could be the development of avatar-based assistive technologies for SL content presentation, providing accessibility and usability in digital platforms for this group of people.
- Finally, another essential and complex problem concerns SL translation, which involves the interpretation of the meaning in a language and the production of a rendering in another, as accurately as possible. The text/voice translation is basically achieved by a process in which the text/voice is received as input, then properly processed and finally translated to movements by the avatar. During this process we can lose essential characteristics of a language such as syntax and semantics. This might lead to wrong avatar movements and consequently incorrect translation. Thus, research on potential technology solutions for enhancing the communication process for deaf people on e-learning platforms through SL translation constitutes an important task.

References

- [1] D. F. Armstrong, M. A. Karchmer, and J. V. VanCleve. *The Study of Signed Languages: Essays in Honor of William C. Stokoe*. Washington, DC, 2002.
- [2] K. Antzakas and B. Woll. Head movements and negation in Greek sign language. In I. Wachsmuth and T. Sowa, editors, *Gesture and Sign Language in Human-Computer Interaction*, volume LNCS-2298, pages 193–196. Springer, 2002.
- [3] K. Schönström and J. Mesch. Use of nonmanuals by adult L2 signers in Swedish sign language – Annotating the nonmanuals. In *Proceedings of the International Conference on Language Resources and Evaluation*, page 153–156, 2014.
- [4] R. Rastgoo, K. Kiani, and S. Escalera. Sign language recognition: A deep survey. *Expert Systems with Applications*, 164:113794, 2021.
- [5] M. Rodda and C. Grove. *Language, cognition and Deafness*. New Jersey: Lawrence Erlbaum Associates, 1987.
- [6] M. De Meulder. *The Power of Language Policy: The Legal Recognition of Sign Languages and the Aspirations of Deaf Communities*. 2016. Ph.D. Thesis, Faculty of Humanities, University of Juväskylä, Finland.
- [7] T. Haug, W. Mann, E. Boers-Visker, J. Contreras, C. Enns, R. Herman, and K. Rowley. *Guidelines for Sign Language Test Development, Evaluation, and Use*. 2016. Unpublished document (upd. 2018), retrieved from <http://www.signlang-assessment.info/>.
- [8] D. H. Smith and J. E. Davis. Formative assessment for student progress and program improvement in sign language as L2 programs. In D. McKee, R. S. Rosen,

- and R. McKee, editors, *Teaching and Learning Signed Languages*, pages 253–280. Palgrave Macmillan, London, 2014.
- [9] N. C. Camgöz, S. Hadfield, O. Koller, H. Ney, and R. Bowden. Neural sign language translation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7784–7793, 2018.
- [10] N. C. Camgöz, O. Koller, S. Hadfield, and R. Bowden. 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*, pages 10023–10033, 2020.
- [11] V. Pitsikalis, S. Theodorakis, C. Vogler, and P. Maragos. Advances in phonetics-based sub-unit modeling for transcription alignment and sign language recognition. In *Proceedings of the IEEE Computer Vision & Pattern Recognition Workshops*, pages 1–6, 2011.
- [12] G. Sapountzaki, E. Efthimiou, S.-E. Fotinea, K. Papadimitriou, and G. Potamianos. Educational material organization in a platform for Greek Sign Language self monitoring and assessment. In *Proceedings of the International Conference on Education and New Learning Technologies (EDULEARN)*, pages 3322–3331, 2021.
- [13] O. Koller, J. Forster, and H. Ney. Continuous sign language recognition: Towards large vocabulary statistical recognition systems handling multiple signers. *Computer Vision and Image Understanding*, 141:108–125, 2015.
- [14] O. M. Sincan and H. Y. Keles. AUTSL: A large scale multi-modal Turkish sign language dataset and baseline methods. *IEEE Access*, 8:181340–181355, 2020.
- [15] E. Efthimiou, K. Vasilaki, S.-E. Fotinea, A. Vacalopoulou, T. Goulas, and A.-L. Dimou. The POLYTROPON parallel corpus. In *Proceedings of the International Conference on Language Resources and Evaluation*, pages 39–44, 2018.
- [16] B. Shi, A. Martinez Del Rio, J. Keane, J. Michaux, D. Brentari, G. Shakhnarovich, and K. Livescu. American sign language fingerspelling recognition in the wild. In *Proceedings of the IEEE Spoken Language Technology*, pages 145–152, 2018.

- [17] S. A. Mehdi and Y. N. Khan. Sign language recognition using sensor gloves. In *Proceedings of the Int. Conference on Neural Information Processing (ICONIP)*, pages 2204–2206, 2002.
- [18] C. Oz and M. C. Leu. American sign language word recognition with a sensory glove using artificial neural networks. *Engineering Applications of Artificial Intelligence*, 24(7):1204–1213, 2011.
- [19] M. Caputo, K. Denker, B. Dums, and G. Umlauf. 3D hand gesture recognition based on sensor fusion of commodity hardware. In *Proceedings of the Mensch & Computer*, pages 293–302, 2012.
- [20] A. Mittal, P. Kumar, P. P. Roy, R. Balasubramanian, and B. B. Chaudhuri. A modified LSTM model for continuous sign language recognition using leap motion. *IEEE Sensors Journal*, 19(16):7056–7063, 2019.
- [21] B. G. Lee and S. M. Lee. Smart wearable hand device for sign language interpretation system with sensors fusion. *IEEE Sensors Journal*, 18(3):1224–1232, 2018.
- [22] Z. Ren, J. Yuan, and Z. Zhang. Robust hand gesture recognition based on finger-earth movers distance with a commodity depth camera. In *Proceedings of the ACM International Conference on Multimedia*, pages 1093–1096, 2011.
- [23] A. Kurakin, Z. Zhang, and Z. Liu. A real time system for dynamic hand gesture recognition with a depth sensor. In *Proceedings of the European Signal Processing Conference*, pages 1975–1979, 2012.
- [24] S. Escalera, J. González, X. Baró, M. Reyes, I. M. Guyon, V. Athitsos, H. J. Escalante, L. Sigal, A. A. Argyros, C. Sminchisescu, R. Bowden, and S. Sclaroff. ChaLearn multi-modal gesture recognition 2013: Grand challenge and workshop summary. In *Proceedings of the International Conference on Multimodal Interaction (ICMI)*, pages 365–368, 2013.
- [25] A. A. Hosain, P. S. Santhalingam, P. Pathak, J. Kosecka, and H. Rangwala. Sign language recognition analysis using multimodal data. In *Proceedings of the IEEE International Conference on Data Science and Advanced Analytics*, pages 203–210, 2019.

- [26] I. Tashev. Kinect development kit: A toolkit for gesture- and speech-based human-machine interaction [Best of the Web]. *IEEE Signal Processing Magazine*, 30(5):129–131, 2013.
- [27] L. Keselman, J. I. Woodfill, A. Grunnet-Jepsen, and A. Bhowmik. Intel RealSense stereoscopic depth cameras. *ArXiv*, abs/1705.05548, 2017.
- [28] R. Cutler and M. Turk. View-based interpretation of real-time optical flow for gesture recognition. In *Proceedings of the IEEE International Conference on Automatic Face and Gesture Recognition*, pages 416–421, 1998.
- [29] K. Papadimitriou, G. Sapountzaki, K. Vasilaki, E. Efthimiou, S.-E. Fotinea, and G. Potamianos. SL-REDU GSL: A large greek sign language recognition corpus. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing Workshop on Sign Language Translation and Avatar Technology (ICASSPW-SLTAT)*, pages 1–5, 2023.
- [30] K. Grobel and M. Assan. Isolated sign language recognition using hidden Markov models. In *Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics*, pages 162–167, 1997.
- [31] H. Cooper, E.-J. Ong, N. Pugeault, and R. Bowden. Sign language recognition using sub-units. *Journal of Machine Learning Research*, 13:2205–2231, 2012.
- [32] A. Roussos, S. Theodorakis, V. Pitsikalis, and P. Maragos. Dynamic affine-invariant shape-appearance handshake features and classification in sign language videos. *Journal of Machine Learning Research*, 14:1627–1663, 2013.
- [33] D. Konstantinidis, K. Dimitropoulos, and P. Daras. A deep learning approach for analyzing video and skeletal features in sign language recognition. In *Proceedings of the IEEE International Conference on Imaging Systems and Techniques*, pages 1–6, 2018.
- [34] G. Potamianos, K. Papadimitriou, E. Efthimiou, S. E. Fotinea, G. Sapountzaki, and P. Maragos. SL-ReDu: Greek sign language recognition for educational applications. Project description and early results. In *Proceedings of the Pervasive Technologies Related to Assistive Environments Conference (PETRA)*, pages 1–6, 2020.

- [35] H. Zhou, W. Zhou, Y. Zhou, and H. Li. Spatial-temporal multi-cue network for continuous sign language recognition. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 13009–13016, 2020.
- [36] K. Papadimitriou and G. Potamianos. Multimodal sign language recognition via temporal deformable convolutional sequence learning. In *Proceedings of the Interspeech*, pages 2752–2756, 2020.
- [37] M. Parelli, K. Papadimitriou, G. Potamianos, G. Pavlakos, and P. Maragos. Exploiting 3D hand pose estimation in deep learning-based sign language recognition from RGB videos. In *Proceedings of the ECCV (SLRTP) 2020 Workshops*, pages 249–263, 2020.
- [38] I. Papastratis, K. Dimitropoulos, D. Konstantinidis, and P. Daras. Continuous sign language recognition through cross-modal alignment of video and text embeddings in a joint-latent space. *IEEE Access*, 8:91170–91180, 2020.
- [39] J. Pu, W. Zhou, H. Hu, and H. Li. Boosting continuous sign language recognition via cross modality augmentation. In *Proceedings of the ACM International Conference on Multimedia*, pages 1497–1505, 2020.
- [40] T. Kim, J. Keane, W. Wang, H. Tang, J. Riggle, G. Shakhnarovich, D. Brentari, and K. Livescu. Lexicon-free fingerspelling recognition from video: Data, models, and signer adaptation. *CoRR*, abs/1609.07876, 2016.
- [41] B. Shi and K. Livescu. Multitask training with unlabeled data for end-to-end sign language fingerspelling recognition. *CoRR*, abs/1710.03255, 2017.
- [42] C. K. Mummadi, F. P. P. Leo, K. D. Verma, S. Kasireddy, P. M. Scholl, and K. V. Laerhoven. Real-time embedded recognition of sign language alphabet fingerspelling in an IMU-based glove. In *Proceedings of the iWOAR*, pages 1–6, 2017.
- [43] B. Shi, A. Martinez Del Rio, J. Keane, D. Brentari, G. Shakhnarovich, and K. Livescu. Fingerspelling recognition in the wild with iterative visual attention. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 5399–5408, 2019.

- [44] K. Papadimitriou and G. Potamianos. End-to-end convolutional sequence learning for ASL fingerspelling recognition. In *Proceedings of the Annual Conference of the International Speech Communication Association*, pages 2315–2319, 2019.
- [45] P. Heracleous, D. Beutemps, and N. Aboutabit. Cued speech automatic recognition in normal-hearing and deaf subjects. *Speech Communication*, 52(6):504–512, 2010.
- [46] L. Liu, G. Feng, D. Beutemps, and X.-P. Zhang. Re-synchronization using the hand preceding model for multi-modal fusion in automatic continuous cued speech recognition. *IEEE Transactions on Multimedia*, 23:292–305, 2021.
- [47] K. Papadimitriou, M. Parelli, G. Sapountzaki, G. Pavlakos, P. Maragos, and G. Potamianos. Multimodal fusion and sequence learning for cued speech recognition from videos. In *Proceedings of the Universal Access in Human-Computer Interaction. Access to Media, Learning and Assistive Environments*, pages 277–290, 2021.
- [48] L. Liu, T. Hueber, G. Feng, and D. Beutemps. Visual recognition of continuous cued speech using a tandem CNN-HMM approach. In *Proceedings of the Interspeech*, pages 2643–2647, 2018.
- [49] H. D. Yang. Sign language recognition with the Kinect sensor based on conditional random fields. *Sensors*, 15:135–47, 12 2014.
- [50] T. Starner, J. Weaver, and A. Pentland. A wearable computer based American sign language recognizer. In *Proceedings of the International Symposium on Wearable Computers*, pages 130–137, 1997.
- [51] J. Pansare, S. Gawande, and M. Ingle. Real-time static hand gesture recognition for american sign language (ASL) in complex background. *Journal of Signal and Information Processing*, 03:364–367, 01 2012.
- [52] P. Viola and M. Jones. Robust real-time face detection. *International Journal of Computer Vision*, 57(2):137–154, 2004.
- [53] G. Awad, J. Han, and A. Sutherland. A unified system for segmentation and tracking of face and hands in sign language recognition. In *Proceedings of the International Conference on Pattern Recognition (ICPR)*, pages 239–242, 2006.

- [54] K. Papadimitriou and G. Potamianos. A hybrid approach to hand detection and type classification in upper-body videos. In *Proceedings of the European Workshop on Visual Information Processing (EUVIP)*, pages 1–6, 2018.
- [55] K. Papadimitriou and G. Potamianos. Fingerspelled alphabet sign recognition in upper-body videos. In *Proceedings of the European Signal Processing Conference (EUSIPCO)*, pages 1–5, 2019.
- [56] S. Jiang, B. Sun, L. Wang, Y. Bai, K. Li, and Y. Fu. Skeleton aware multi-modal sign language recognition. In *Proceedings of the CVPR Workshops*, pages 3408–3418, 2021.
- [57] R. Zuo and B. Mak. C2SLR: Consistency-enhanced continuous sign language recognition. In *Proceedings of the Conference on Computer Vision and Pattern Recognition*, pages 5121–5130, 2022.
- [58] K. Papadimitriou and G. Potamianos. Sign language recognition via deformable 3D convolutions and modulated graph convolutional networks. In *Proceedings of the International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5, 2023.
- [59] A. Roussos, S. Theodorakis, V. Pitsikalis, and P. Maragos. Hand tracking and affine shape-appearance handshape sub-units in continuous sign language recognition. In *Proceedings of the European conference on Trends and Topics in Computer Vision*, pages 258–272, 2010.
- [60] N. H. Dardas and N. D. Georganas. Real-time hand gesture detection and recognition using bag-of-features and support vector machine techniques. *IEEE Transactions on Instrumentation and Measurement*, 60(11):3592–3607, 2011.
- [61] O. Koller, S. Zargaran, H. Ney, and R. Bowden. Deep sign: Hybrid CNN-HMM for continuous sign language recognition. In *Proceedings of the BMVC*, pages 1–12, 2016.
- [62] N. C. Camgöz, S. Hadfield, O. Koller, and R. Bowden. SubUNets: End-to-end hand shape and continuous sign language recognition. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3075–3084, 2017.

- [63] Z. Niu and B. Mak. Stochastic fine-grained labeling of multi-state sign glosses for continuous sign language recognition. In *Proceedings of the ECCV*, pages 172–186, 2020.
- [64] A. Hao, Y. Min, and X. Chen. Self-mutual distillation learning for continuous sign language recognition. In *Proceedings of the International Conference on Computer Vision*, pages 11283–11292, 2021.
- [65] Y. Min, A. Hao, X. Chai, and X. Chen. Visual alignment constraint for continuous sign language recognition. In *Proceedings of the International Conference on Computer Vision (ICCV)*, pages 11522–11531, 2021.
- [66] K. Papadimitriou and G. Potamianos. Multimodal locally enhanced transformer for continuous sign language recognition. In *Proceedings of the INTERSPEECH 2023*, pages 1513–1517, 2023.
- [67] J. Pu, W. Zhou, and H. Li. Dilated convolutional network with iterative optimization for continuous sign language recognition. In *Proceedings of the International Joint Conference on Artificial Intelligence*, pages 885–891, 2018.
- [68] J. Huang, W. Zhou, Q. Zhang, H. Li, and W. Li. Video-based sign language recognition without temporal segmentation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, page 2257–2264, 2018.
- [69] J. Pu, W. Zhou, and H. Li. Iterative alignment network for continuous sign language recognition. In *Proceedings of the CVPR*, pages 4160–4169, 2019.
- [70] H. Zhou, W. Zhou, and H. Li. Dynamic pseudo label decoding for continuous sign language recognition. In *Proceedings of the IEEE International Conference on Multimedia and Expo (ICME)*, pages 1282–1287, 2019.
- [71] Q. Xiao, X. Chang, X. Zhang, and X. Liu. Multi-information spatial-temporal LSTM fusion continuous sign language neural machine translation. *IEEE Access*, 8:216718–216728, 2020.
- [72] T. Starner, J. Weaver, and A. Pentland. Real-time American sign language recognition using desk and wearable computer based video. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(12):1371–1375, 1998.

- [73] P. Dreuw, D. Rybach, T. Deselaers, M. Zahedi, and H. Ney. Speech recognition techniques for a sign language recognition system. In *Proceedings of the Interspeech*, pages 2513–2516, 2007.
- [74] C. Vogler and D. Metaxas. Parallel hidden Markov models for American Sign Language recognition. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 116–122, 1999.
- [75] T. Kim, G. Shakhnarovich, and K. Livescu. Fingerspelling recognition with semi-Markov conditional random fields. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1521–1528, 2013.
- [76] R. Cui, H. Liu, and C. Zhang. Recurrent convolutional neural networks for continuous sign language recognition by staged optimization. In *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1610–1618, 2017.
- [77] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural Computation*, 9(8):1735–1780, 1997.
- [78] M. Schuster and K. K. Paliwal. Bidirectional recurrent neural networks. *IEEE Transactions on Signal Processing*, 45(11):2673–2681, 1997.
- [79] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin. Attention is all you need. In *Proceedings of the Advances in Neural Information Processing Systems (NeurIPS) 30*, pages 5998–6008, 2017.
- [80] M. Parelli, K. Papadimitriou, G. Potamianos, G. Pavlakos, and P. Maragos. Spatio-temporal graph convolutional networks for continuous sign language recognition. In *Proceedings of the ICASSP*, pages 8457–8461, 2022.
- [81] A. Graves, S. Fernández, F. Gomez, and J. Schmidhuber. Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks. In *Proceedings of the International Conference on Machine Learning*, page 369–376, 2006.
- [82] C. C. Amorim, D. Macêdo, and C. Zanchettin. Spatial-temporal graph convolutional networks for sign language recognition. In *Proceedings of the ICANN*, pages 646–657, 2019.

- [83] L. Meng and R. Li. An attention-enhanced multi-scale and dual sign language recognition network based on a graph convolution network. *Sensors*, 21(4):1120, 2021.
- [84] M. Vázquez-Enríquez, J. L. Alba-Castro, L. Docío-Fernández, and E. Rodríguez-Banga. Isolated sign language recognition with multi-scale spatial-temporal graph convolutional networks. In *Proceedings of the CVPR Workshops*, pages 3457–3466, 2021.
- [85] D. Tran, H. Wang, L. Torresani, J. Ray, Y. LeCun, and M. Paluri. A closer look at spatiotemporal convolutions for action recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6450–6459, 2018.
- [86] Z. Zou and W. Tang. Modulated graph convolutional network for 3D human pose estimation. In *Proceedings of the ICCV*, pages 11457–11467, 2021.
- [87] Y. Feng, V. Choutas, T. Bolkart, D. Tzionas, and M. J. Black. Collaborative regression of expressive bodies using moderation. In *Proceedings of the 3DV*, pages 792–804, 2021.
- [88] Y. Zheng, X. Li, F. Xie, and L. Lu. Improving end-to-end speech synthesis with local recurrent neural network enhanced transformer. In *Proceedings of the International Conference on Acoustics, Speech and Signal Processing*, pages 6734–6738, 2020.
- [89] B. Yang, Z. Tu, D. F. Wong, F. Meng, L. S. Chao, and T. Zhang. Modeling localness for self-attention networks. *ArXiv*, abs/1810.10182, 2018.
- [90] P. Shaw, J. Uszkoreit, and A. Vaswani. Self-attention with relative position representations. In *Proceedings of the Conference of the NAACL-HLT*, pages 464–468, 2018.
- [91] G. Kurata and K. Audhkhasi. Guiding CTC posterior spike timings for improved posterior fusion and knowledge distillation. In *Proceedings of the Interspeech*, pages 1616–1620, 2019.
- [92] G. E. Hinton, O. Vinyals, and J. Dean. Distilling the knowledge in a neural network. *ArXiv*, abs/1503.02531, 2015.

- [93] A. Hannun, A. Lee, Q. Xu, and R. Collobert. Sequence-to-sequence speech recognition with time-depth separable convolutions. *CoRR*, abs/1904.02619, 2019.
- [94] K. Papadimitriou and G. Potamianos. A fully convolutional sequence learning approach for cued speech recognition from videos. In *Proceedings of the European Signal Processing Conference (EUSIPCO)*, pages 326–330, 2021.
- [95] S. Matthes, T. Hanke, A. Regen, J. Storz, S. Wörseck, E. Efthimiou, A.-L. Dimou, A. Braffort, J. Glauert, and E. Safar. Dicta-Sign – Building a multilingual sign language corpus. In *Proceedings of the Workshop on Representation and Processing of Sign Languages: Inter. Between Corpus and Lexicon (Satellite to LREC)*, 2012.
- [96] N. Adaloglou, T. Chatzis, I. Papastratis, A. Stergioulas, G. Papadopoulos, V. Zacharopoulou, G. J. Xydopoulos, K. Atzakis, D. Papazachariou, and P. Daras. A comprehensive study on sign language recognition methods. *IEEE Transactions on Multimedia*, 24:1750–1762, 2022.
- [97] U. von Agris, M. Knorr, and K. Kraiss. The significance of facial features for automatic sign language recognition. In *Proceedings of the IEEE Int. Conf. on Automatic Face and Gesture Recognition (FG)*, 2008.
- [98] H. R. V. Joze and O. Koller. MS-ASL: A large-scale data set and benchmark for understanding American Sign Language. In *Proceedings of the British Machine Vision Conf. (BMVC)*, 2019.
- [99] S. Albanie, G. Varol, L. Momeni, H. Bull, T. Afouras, H. Chowdhury, N. Fox, B. Woll, R. Cooper, A. McParland, and A. Zisserman. BOBSL: BBC-Oxford British sign language dataset. *CoRR*, abs/2111.03635, 2021.
- [100] J. Pu, W. Zhou, and H. Li. Sign language recognition with multi-modal features. In E. Chen, Y. Gonh, and Y. Tie, editors, *Advances in Multimedia Information Processing - PCM 2016: 17th Pacific-Rim Conference on Multimedia*, pages 252–261. Springer, 2016.
- [101] L. Liu, J. Li, G. Feng, and X. Zhang. Automatic detection of the temporal segmentation of hand movements in British English cued speech. In *Proceedings of the Interspeech*, pages 2285–2289, 2019.

- [102] T. Hanke. iLex - a tool for sign language lexicography and corpus analysis. In *Proceedings of the International Conference on Language Resources and Evaluation (LREC)*, pages 923–926, 2002.
- [103] ELAN (Version 5.8) [Computer software], 2019. Nijmegen: Max Planck Institute for Psycholinguistics, The Language Archive. <https://archive.mpi.nl/tla/elan>.
- [104] O. Crasborn and H. Sloetjes. Enhanced ELAN functionality for sign language corpora. In *Proceedings of the Workshop on the Representation and Processing of Sign Languages: Construction and Exploitation of Sign Language Corpora*, pages 39–43, 2008.
- [105] Common European framework of references for languages. learning, teaching, assessment – Companion volume, language policy programme, education policy division, education department, council of Europe, 2020. <https://www.coe.int/en/web/common-european-framework-reference-languages>.
- [106] G. Sapountzaki, E. Efthimiou, S.-E. Fotinea, K. Papadimitriou, and G. Potamianos. 3D Greek Sign Language classifiers as a learning object in the SL-ReDu online education platform. In *Proceedings of the International Conference on Education and New Learning Technologies (EDULEARN)*, pages 6146–6153, 2022.
- [107] SL-ReDu GSL corpus. <https://sl-redu.e-ce.uth.gr/corpus/>.
- [108] T. Simon, H. Joo, I. Matthews, and Y. Sheikh. Hand keypoint detection in single images using multiview bootstrapping. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4645–4653, 2017.
- [109] K. Sun, B. Xiao, D. Liu, and J. Wang. Deep high-resolution representation learning for human pose estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, page 5693–5703, 2019.
- [110] N. Samet and E. Akbas. HPRNet: Hierarchical point regression for whole-body human pose estimation. *Image and Vision Computing*, 115:104285, 2021.
- [111] C. Lugaresi, J. Tang, H. Nash, C. McClanahan, E. Uboweja, M. Hays, F. Zhang, C.-L. Chang, M. Yong, J. Lee, W.-T. Chang, W. Hua, M. Georg, and M. Grundmann.

- MediaPipe: A framework for perceiving and processing reality. In *Proceedings of the Workshop on Computer Vision for AR/VR at IEEE CVPR*, 2019.
- [112] V. Choutas, G. Pavlakos, T. Bolkart, D. Tzionas, and M. J. Black. Monocular expressive body regression through body-driven attention. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 20–40, 2020.
- [113] Y. Rong, T. Shiratori, and H. Joo. FrankMocap: A monocular 3D whole-body pose estimation system via regression and integration. *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops (ICCVW)*, pages 1749–1759, 2021.
- [114] K. B. Shaik, P. Ganesan, V. Kalist, B.S. Sathish, and J. M. M. Jenitha. Comparative study of skin color detection and segmentation in HSV and YCbCr color space. *Procedia Computer Science*, 57:41–48, 2015.
- [115] J. Jeong, T. Yoon, and J. Park. Kalman filter based multiple objects detection-tracking algorithm robust to occlusion. In *Proceedings of the SICE Annual Conference*, pages 941–946, 2014.
- [116] P. Liao, T. Chen, and P. Chung. A fast algorithm for multilevel thresholding. *Journal of Information Science and Engineering*, 17:713–727, 2001.
- [117] A. Krizhevsky, I. Sutskever, and G. E. Hinton. ImageNet classification with deep convolutional neural networks. In *Proceedings of the Advances in Neural Information Processing Systems*, pages 1097–1105, 2012.
- [118] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and Li Fei-Fei. ImageNet: A large-scale hierarchical image database. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 248–255, 2009.
- [119] A. Mittal, A. Zisserman, and P. H. S. Torr. Hand detection using multiple proposals. In *Proceedings of the British Machine Vision Conference*, pages 1–11, 2011.
- [120] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In *Proceedings of the International Conference on Learning Representations*, page 1–14, 2015.

- [121] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 580–587, 2014.
- [122] A. Kanazawa, Michael J. Black, D. Jacobs, and J. Malik. End-to-end recovery of human shape and pose. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7122–7131, 2018.
- [123] G. Pavlakos, V. Choutas, N. Ghorbani, T. Bolkart, A. A. A. Osman, D. Tzionas, and M. J. Black. Expressive body capture: 3D hands, face, and body from a single image. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 10975–10985, 2019.
- [124] M. A. Rahman and Y. Wang. Optimizing Intersection-Over-Union in deep neural networks for image segmentation. In *Proceedings of the International Symposium on Visual Computing*, pages 234–244, 2016.
- [125] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri. Learning spatiotemporal features with 3D convolutional networks. In *Proceedings of the International Conference on Computer Vision (ICCV)*, pages 4489–4497, 2015.
- [126] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna. Rethinking the Inception architecture for computer vision. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2818–2826, 2016.
- [127] S. Xie, R. Girshick, P. Dollár, Z. Tu, and K. He. Aggregated residual transformations for deep neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5987–5995, 2017.
- [128] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam. MobileNets: Efficient convolutional neural networks for mobile vision applications. *CoRR*, arXiv:1704.04861, 2017.
- [129] K. Hara, H. Kataoka, and Y. Satoh. Learning spatio-temporal features with 3D residual networks for action recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3154–3160, 2017.

- [130] R. Qian, T. Meng, B. Gong, M.-H. Yang, H. Wang, S. J. Belongie, and Y. Cui. Spatiotemporal contrastive video representation learning. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6960–6970, 2020.
- [131] C. Zach, T. Pock, and H. Bischof. A duality based approach for realtime TV-L1 optical flow. In *Proceedings of the International Conference on Pattern Recognition*, pages 214–223, 2007.
- [132] A. Ranjan and M. J. Black. Optical flow estimation using a spatial pyramid network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2720–2729, 2017.
- [133] E. Ilg, N. Mayer, T. Saikia, M. Keuper, A. Dosovitskiy, and T. Brox. FlowNet 2.0: Evolution of optical flow estimation with deep networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1647–1655, 2017.
- [134] D. Sun, X. Yang, M.-Y. Liu, and J. Kautz. PWC-Net: CNNs for optical flow using pyramid, warping, and cost volume. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8934–8943, 2018.
- [135] Z. Teed and J. Deng. Raft: Recurrent all-pairs field transforms for optical flow. In *Proceedings of the European Conference on Computer Vision*, pages 402–419, 2020.
- [136] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016.
- [137] J. Carreira, E. Noland, A. Banki-Horvath, C. Hillier, and A. Zisserman. A short note about Kinetics-600. *CoRR*, arXiv:1808.01340, 2018.
- [138] Z. Qiu, T. Yao, and T. Mei. Learning spatio-temporal representation with Pseudo-3D residual networks. In *Proceedings of the International Conference on Computer Vision (ICCV)*, pages 5534–5542, 2017.

- [139] J. Carreira and A. Zisserman. Quo Vadis, Action Recognition? a new model and the Kinetics dataset. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4724–4733, 2017.
- [140] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–9, 2015.
- [141] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *CoRR*, arXiv:1412.6980, 2014.
- [142] C. Szegedy, V. Vanhoucke, S. Ioffe, . Shlens, and Z. Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2818–2826, 2016.
- [143] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer. Automatic differentiation in PyTorch. In *Proceedings of the NIPS-W*, 2017.
- [144] E. Bisong. *Building Machine Learning and Deep Learning Models on Google Cloud Platform: A Comprehensive Guide for Beginners*. 2019.
- [145] K. Cho, B. van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio. Learning phrase representations using RNN encoder-decoder for statistical machine translation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1724–1734, 2014.
- [146] G. Hinton, L. Deng, D. Yu, G. E. Dahl, A. r. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. N. Sainath, and B. Kingsbury. Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. *Signal Processing Magazine, IEEE*, 29:82–97, 11 2012.
- [147] D. Yu and L. Deng. Automatic speech recognition: A deep learning approach. *Springer-Verlag*, 2015.

- [148] W. Xiong, J. Droppo, X. Huang, F. Seide, M. Seltzer, A. Stolcke, D. Yu, and G. Zweig. Achieving human parity in conversational speech recognition. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 2016.
- [149] K. Cho, B. van Merriënboer, D. Bahdanau, and Y. Bengio. On the properties of neural machine translation: Encoder–decoder approaches. In *Proceedings of the Workshop on Syntax, Semantics and Structure in Statistical Translation*, pages 103–111, 2014.
- [150] K. Cho, B. van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio. Learning phrase representations using RNN encoder–decoder for statistical machine translation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1724–1734, 2014.
- [151] D. Bahdanau, K. Cho, and Y. Bengio. Neural machine translation by jointly learning to align and translate. *CoRR*, abs/1409.0473, 2014.
- [152] M.-T. Luong, H. Pham, and C. D. Manning. Effective approaches to attention-based neural machine translation. *CoRR*, abs/1508.04025, 2015.
- [153] J. Zhang, J. Tang, and L.-R. Dai. RNN-BLSTM based multi-pitch estimation. In *Proceedings of the Interspeech*, pages 1785–1789, 2016.
- [154] J. Gehring, M. Auli, D. Grangier, D. Yarats, and Y. N. Dauphin. Convolutional sequence to sequence learning. *CoRR*, abs/1705.03122, 2017.
- [155] Y. N. Dauphin, A. Fan, M. Auli, and D. Grangier. Language modeling with gated convolutional networks. *CoRR*, abs/1612.08083, 2016.
- [156] Aä. Oord, N. Kalchbrenner, and K. Kavukcuoglu. Pixel recurrent neural networks. *CoRR*, abs/1601.06759, 2016.
- [157] S. Chollampatt and H. T. Ng. A multilayer convolutional encoder-decoder neural network for grammatical error correction. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2018.
- [158] M. Freitag and Y. Al-Onaizan. Beam search strategies for neural machine translation. *CoRR*, abs/1702.01806, 2017.

- [159] I. Papastratis, K. Dimitropoulos, and P. Daras. Continuous sign language recognition through a context-aware generative adversarial network. *Sensors*, 21(7), 2021.
- [160] D. Li, C. R. Opazo, X. Yu, and H. Li. Word-level deep sign language recognition from video: A new large-scale dataset and methods comparison. In *Proceedings of the WACV*, pages 1448–1458, 2020.
- [161] C. Ong, I. Lim, J. Lu, C. Ng, and T. Ong. Sign-language recognition through gesture & movement analysis (SIGMA). In J. Billingsley and P. Brett, editors, *Mechatronics and Machine Vision in Practice 3*, pages 235–245. Springer, 2018.
- [162] J. Zhang, W. Zhou, C. Xie, J. Pu, and H. Li. Chinese sign language recognition with adaptive HMM. In *Proceedings of the IEEE International Conference on Multimedia and Expo (ICME)*, page 1–6, 2016.
- [163] M. Al-Hammadi, M. A. Bencherif, M. Alsulaiman, G. Muhammad, M. A. Mekhtiche, W. Abdul, Y. A. Alohal, T. S. Alrayes, H. Mathkour, M. Faisal, M. Algabri, H. Al-taheri, T. Alfakih, and H. Ghaleb. Spatial attention-based 3D graph convolutional neural network for sign language recognition. *Sensors*, 22(12), 2022.
- [164] M. De Coster, M. Van Herreweghe, and J. Dambre. Isolated sign recognition from RGB video using pose flow and self-attention. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 3436–3445, 2021.
- [165] J. Dai, H. Qi, Y. Xiong, Y. Li, G. Zhang, H. Hu, and Y. Wei. Deformable convolutional networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 764–773, 2017.
- [166] S. Elfving, E. Uchibe, and K. Doya. Sigmoid-weighted linear units for neural network function approximation in reinforcement learning. *Neural Networks*, 107:3–11, 2018.
- [167] K. Cheng, Y. Zhang, C. Cao, L. Shi, J. Cheng, and H. Lu. Decoupling GCN with DropGraph module for skeleton-based action recognition. In *Proceedings of the ECCV*, page 536–553, 2020.

- [168] K. Papadimitriou, G. Potamianos, G. Sapountzaki, T. Goulas, E. Efthimiou, S.-E. Fotinea, and P. Maragos. Greek sign language recognition for an education platform. *Universal Access in the Information Society*, pages 1–18, 2023.
- [169] P. Selvaraj, G. Nc, P. Kumar, and M. Khapra. OpenHands: Making sign language recognition accessible with pose-based pretrained models across languages. In *Proceedings of the ACL*, pages 2114–2133, 2022.
- [170] O. Koller, S. Zargaran, and H. Ney. Re-Sign: Re-aligned end-to-end sequence modelling with deep recurrent CNN-HMMs. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3416–3424, 2017.
- [171] P. Xie, M. Zhao, and X. Hu. PiSLTRc: Position-informed sign language transformer with content-aware convolution. *IEEE Transactions on Multimedia*, 24:3908–3919, 2021.
- [172] Z. Cui, W. Zhang, Z. Li, and Z. Wang. Spatial–temporal Transformer for end-to-end sign language recognition. *Complex and Intelligent Systems*, 9(4):4645–4656, 2023.
- [173] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In *Proceedings of the International Conference on Learning Representations*, pages 1–14, 2015.
- [174] O. Koller, N. C. Camgoz, and H. Ney and R. Bowden. Weakly supervised learning with multi-stream CNN-LSTM-HMMs to discover sequential parallelism in sign language videos. *TPAMI*, 42(9):2306–2320, 2020.
- [175] P. Goh and E. Holden. Dynamic fingerspelling recognition using geometric and motion features. In *Proceedings of the International Conference on Image Processing*, pages 2741–2744, 2006.
- [176] S. Liwicki and M. Everingham. Automatic recognition of fingerspelled words in British Sign Language. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 50–57, 2009.
- [177] S. Ricco and C. Tomasi. Fingerspelling recognition through classification of letter-to-letter transitions. In *Proceedings of the European Conference on Computer Vision*, pages 214–225, 2010.

- [178] B. Kang, S. Tripathi, and T. Q. Nguyen. Real-time sign language fingerspelling recognition using convolutional neural networks from depth map. In *Proceedings of the Asian Conference on Pattern Recognition*, pages 136–140, 2015.
- [179] S. Ameen and S. Vadera. A convolutional neural network to classify American sign language fingerspelling from depth and colour images. *Expert Systems*, 34(3):12197, 2017.
- [180] S. Upendran and A. Thamizharasi. American sign language interpreter system for deaf and dumb individuals. In *Proceedings of the International Conference on Control, Instrumentation, Communication and Computational Technologies*, pages 1477–1481, 2014.
- [181] W. Tao, M. C. Leu, and Z. Yin. American sign language alphabet recognition using convolutional neural networks with multiview augmentation and inference fusion. *Engineering Applications of Artificial Intelligence*, 76:202 – 213, 2018.
- [182] Y. Hu, H. F. Zhao, and Z. G. Wang. Sign language fingerspelling recognition using depth information and deep belief networks. *International Journal of Pattern Recognition and Artificial Intelligence*, 32(06):1850018, 2017.
- [183] L. Quesada, G. López, and L. Guerrero. Automatic recognition of the american sign language fingerspelling alphabet to assist people living with speech or hearing impairments. *Journal of Ambient Intelligence and Humanized Computing*, 8(4):625–635, 2017.
- [184] C. S. Weerasekera, M. H. Jaward, and N. Kamrani. Robust ASL fingerspelling recognition using local binary patterns and geometric features. In *Proceedings of the International Conference on Digital Image Computing: Techniques and Applications*, pages 1–8, 2013.
- [185] B. Shi, D. Brentari, G. Shakhnarovich, and K. Livescu. Fingerspelling detection in American sign language. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4164–4173, 2021.

- [186] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016.
- [187] J. Martinez, R. Hossain, J. Romero, and J. J. Little. A simple yet effective baseline for 3D human pose estimation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2659–2668, 2017.
- [188] K. He, X. Zhang, S. Ren, and J. Sun. Delving deep into rectifiers: Surpassing human-level performance on ImageNet classification. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 1026–1034, 2015.
- [189] J. Duchi, E. Hazan, and Y. Singer. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research*, 12:2121–2159, 2011.
- [190] W. Kumwilaisak, P. Pannattee, C. Hansakunbuntheung, and N. Thatphithakkul. American sign language fingerspelling recognition in the wild with iterative language model construction. *APSIPA Transactions on Signal and Information Processing*, 11(1):22, 2022.
- [191] K. Gajurel, C. Zhong, and G. Wang. A fine-grained visual attention approach for fingerspelling recognition in the wild. *IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pages 3266–3271, 2021.
- [192] P. Pannattee, W. Kumwilaisak, C. Hansakunbuntheung, and N. Thatphithakkul. Novel american sign language fingerspelling recognition in the wild with weakly supervised learning and feature embedding. In *Proceedings of the International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON)*, pages 291–294, 2021.
- [193] R. O. Cornett. Cued speech. *American Annals of the Deaf*, 112(1):3–13, 1967.
- [194] V. Attina, D. Beutemps, M. Cathiard, and M. Odisio. A pilot study of temporal organization in cued speech production of French syllables: rules for a cued speech synthesizer. *Speech Communication*, 44(1):197–214, 2004.

- [195] G. Potamianos, E. Marcheret, Y. Mroueh, V. Goel, A. Koumbaroulis, A. Vartholomaios, and S. Thermos. Audio and visual modality combination in speech processing applications. In S. Oviatt, B. Schuller, P. Cohen, D. Sonntag, G. Potamianos, and A. Krüger, editors, *The Handbook of Multimodal-Multisensor Interfaces, Volume 1: Foundations, User Modeling, and Common Modality Combinations*, pages 489–543. Morgan-Claypool, 2017.
- [196] V. Bheda and D. Radpour. Using deep convolutional networks for gesture recognition in American Sign Language. *CoRR*, abs/1710.06836, 2017.
- [197] G. A. Rao, K. Syamala, P. V. V. Kishore, and A. S. C. S. Sastry. Deep convolutional neural networks for sign language recognition. In *Proceedings of the Signal Processing and Communication Engineering Systems (SPACES)*, pages 194–197, 2018.
- [198] L. Liu and G. Feng. A pilot study on Mandarin Chinese cued speech. *American Annals of the Deaf*, 164(4):496–518, 2019.
- [199] P. Heracleous, D. Beutemps, and N. Hagita. Continuous phoneme recognition in cued speech for French. In *Proceedings of the European Signal Processing Conference (EUSIPCO)*, pages 2090–2093, 2012.
- [200] L. Liu, G. Feng, and D. Beutemps. Automatic temporal segmentation of hand movements for hand positions recognition in French cued speech. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 3061–3065, 2018.
- [201] J. Wang, Z. Tang, X. Li, M. Yu, Q. Fang, and L. Liu. Cross-Modal Knowledge Distillation Method for Automatic Cued Speech Recognition. In *Proceedings of the Interspeech*, pages 2986–2990, 2021.
- [202] L. Liu and L. Liu. Cross-modal mutual learning for cued speech recognition. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5, 2023.
- [203] C. Zimmermann and T. Brox. Learning to estimate 3D hand pose from single rgb images. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 4913–4921, 2017.

- [204] Mixamo: Quality 3D character animation in minutes. [online]: <https://www.mixamo.com>.
- [205] Blender Online Community. *Blender - A 3D modelling and rendering package*. Blender Foundation, Stichting Blender Foundation, Amsterdam, 2018.
- [206] M. Kemp. Why is learning American Sign Language a challenge? *American Annals of the Deaf*, 143(3):255–259, 1998.
- [207] J. H. Bochner, V. J. Samar, P. C. Hauser, W. M. Garrison, J. M. Searls, and C. A. Sanders. Validity of the American Sign Language discrimination test. *Language Testing*, 33(4):473–495, 2016.
- [208] P. C. Hauser, R. Paludnevičienė, W. Riddle, K. B. Kurz, K. Emmorey, and J. Contreras. American Sign Language comprehension test: A tool for sign language researchers. *Journal of Deaf Studies and Deaf Education*, 21(1):64–69, 2016.
- [209] O. Aran, I. Ari, L. Akarun, B. Sankur, A. Benoit, A. Caplier, P. Campr, A. H. Carrillo, and F.-X. Fanard. SignTutor: An interactive system for sign language tutoring. *IEEE MultiMedia*, 16(1):81–93, 2009.
- [210] Z. Zafrulla, H. Brashear, P. Presti, H. Hamilton, and T. Starner. CopyCat: An American Sign Language game for deaf children. In *Proceedings of the IEEE Int. Conference on Automatic Face and Gesture Recognition (FG)*, pages 647–647, 2011.
- [211] K. Ellis, J. Fisher, L. Willoughby, and J. C. Barca. A design science exploration of a visual-spatial learning system with feedback. In *Proceedings of the Australasian Conference on Information Systems (ACIS)*, pages 1–13, 2015.
- [212] C.-H. Chuan and C. Guardino. Designing SmartSignPlay: An interactive and intelligent American Sign Language app for children who are deaf or hard of hearing and their families. In *Proceedings of the International Conference on Intelligent User Interfaces (IUI)*, pages 45–48, 2016.
- [213] S. Ebling, N. C. Camgöz, P. B. Braem, K. Tissi, S. Sidler-Miserez, S. Stoll, S. Hadfield, T. Haug, R. Bowden, S. Tornay, M. Razavi, and M. Magimai-Doss. SMILE Swiss German Sign Language dataset. In *Proceedings of the Int. Conference on Language Resources and Evaluation (LREC)*, pages 4221–4229, 2018.

- [214] J. Joy, K. Balakrishnan, and M. Sreeraj. SignQuiz: A quiz based tool for learning fingerspelled signs in Indian Sign Language using ASLR. *IEEE Access*, 7:28363–28371, 2019.
- [215] E. Efthimiou, S.-E. Fotinea, C. Flouda, T. Goulas, G. Ametoglou, G. Sapountzaki, K. Papadimitriou, and G. Potamianos. The SL-ReDu environment for self-monitoring and objective learner assessment in Greek Sign Language. In *Proceedings of the Conference on Universal Access in Human-Computer Interaction. Access to Media, Learning and Assistive Environments (HCII)*, volume LNCS-12769, pages 72–81, 2021.
- [216] K. Papadimitriou, G. Potamianos, G. Sapountzaki, T. Goulas, E. Efthimiou, S.-E. Fotinea, and P. Maragos. Greek Sign Language recognition for the SL-ReDu learning platform. In *Proceedings of the Language Resources and Evaluation Conference Workshop on Sign Language Translation and Avatar Technology (LREC-SLTAT)*, pages 79–84, 2022.
- [217] A. Moryossef, I. Tsochantaridis, J. Dinn, N. C. Camgöz, R. Bowden, T. Jiang, A. Rios, M. Müller, and S. Ebling. Evaluating the immediate applicability of pose estimation for sign language recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 3429–3435, 2021.
- [218] S. Katz. Estimation of probabilities from sparse data for the language model component of a speech recognizer. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 35(3):400–401, 1987.
- [219] F. Jelinek. Interpolated estimation of Markov source parameters from sparse data. *Pattern Recognition in Practice*, page 381–397, 1980.
- [220] K. Heafield. KenLM: Faster and smaller language model queries. In *Proceedings of the Workshop on Statistical Machine Translation (WMT)*, page 87–197, 2011.
- [221] Y. Bengio, R. Ducharme, and P. Vincent. A neural probabilistic language model. In *Journal of Machine Learning Research*, pages 932–938, 2000.

List of Publications Related to the Thesis

List of Publications Related to the Thesis

The innovation and contributions of the Thesis are supported by a series of publications in proceedings of prestigious conferences and in one scientific journal (in total, 18 publications). In addition, a number of technical reports have been produced, as deliverables of the SL-ReDu project.

A. Publications in International Journal

1. **K. Papadimitriou**, G. Potamianos, G. Sapountzaki, T. Goulas, E. Efthimiou, S.-E. Fotinea, and P. Maragos, “Greek sign language recognition for an education platform,” *Universal Access in the Information Society (Springer)*, pp. 1–18, 2023.

B. Publications at International Conferences

1. **K. Papadimitriou** and G. Potamianos, “Multimodal locally enhanced Transformer for continuous sign language recognition,” *Proceedings of the INTERSPEECH*, pp. 1513–1517, 2023.
2. G. Sapountzaki, E. Efthimiou, S.-E. Fotinea, **K. Papadimitriou**, and G. Potamianos, “Remote learning and assessment of Greek sign language in the undergraduate curriculum in COVID time,” *Proceedings of the International Conference on Education and New Learning Technologies (EDULEARN)*, pp. 5452–5459, 2023.
3. **K. Papadimitriou**, G. Sapountzaki, K. Vasilaki, E. Efthimiou, S.-E. Fotinea, and G. Potamianos, “SL-REDU GSL: A large Greek sign language recognition corpus,” *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing Workshops (ICASSPW)*, pp. 1–5, 2023.
4. **K. Papadimitriou** and G. Potamianos, “Sign language recognition via deformable

- 3D convolutions and modulated graph convolutional networks,” *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1–5, 2023.
5. G. Sapountzaki, E. Efthimiou, S.-E. Fotinea, **K. Papadimitriou**, and G. Potamianos, “3D Greek sign language classifiers as a learning object in the SL-ReDu online education platform,” *Proceedings of the International Conference on Education and New Learning Technologies (EDULEARN)*, pp. 6146–6153, 2022.
 6. **K. Papadimitriou**, G. Potamianos, G. Sapountzaki, T. Goulas, E. Efthimiou, S.-E. Fotinea, and P. Maragos, “Greek sign language recognition for the SL-ReDu learning platform,” *Proceedings of the International Workshop on Sign Language Translation and Avatar Technology (SLTAT)*, pp. 79–84, 2022.
 7. M. Parelli, **K. Papadimitriou**, G. Potamianos, G. Pavlakos, and P. Maragos, “Spatio-temporal graph convolutional networks for continuous sign language recognition,” *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 8457–8461, 2022.
 8. E. Efthimiou, S.-E. Fotinea, C. Flouda, T. Goulas, G. Ametoglou, G. Sapountzaki, **K. Papadimitriou**, and G. Potamianos, “The SL-ReDu environment for self-monitoring and objective learner assessment in Greek sign language,” *Proceedings of the Universal Access in Human-Computer Interaction. Access to Media, Learning and Assistive Environments (HCII)*, pp. 72–81, 2021.
 9. **K. Papadimitriou**, M. Parelli, G. Sapountzaki, G. Pavlakos, P. Maragos, and G. Potamianos, “Multimodal fusion and sequence learning for cued speech recognition from videos,” *Proceedings of the International Conference on Human-Computer Interaction (HCII)*, pp. 277–290, 2021.
 10. G. Sapountzaki, E. Efthimiou, S.-E. Fotinea, **K. Papadimitriou**, and G. Potamianos, “Educational material organization in a platform for Greek sign language self monitoring and assessment,” *Proceedings of the International Conference on Education and New Learning Technologies (EDULEARN)*, pp. 3322–3331, 2021.
 11. **K. Papadimitriou** and G. Potamianos, “A fully convolutional sequence learning approach for cued speech recognition from videos,” *Proceedings of the European Signal*

- Processing Conference (EUSIPCO)*, pp. 326–330, 2021.
12. **K. Papadimitriou** and G. Potamianos, “Multimodal sign language recognition via temporal deformable convolutional sequence learning,” *Proceedings of the INTERSPEECH*, pp. 2752–2756, 2020.
 13. M. Parelli, **K. Papadimitriou**, G. Potamianos, G. Pavlakos, and P. Maragos, “Exploiting 3D hand pose estimation in deep learning-based sign language recognition from RGB videos,” *Proceedings of the European Conference on Computer Vision Workshop on Sign Language Recognition, Translation and Production (ECCVW-SLRTP)*, pp. 249–263, 2020.
 14. G. Potamianos, **K. Papadimitriou**, E. Efthimiou, S.-E. Fotinea, G. Sapountzaki, and P. Maragos, “SL-ReDu: Greek sign language recognition for educational applications. Project description and early results,” *Proceedings of the Pervasive Technologies Related to Assistive Environments Conference (PETRA)*, pp. 1–6, 2020.
 15. **K. Papadimitriou** and G. Potamianos, “Fingerspelled alphabet sign recognition in upper-body videos,” *Proceedings of the European Signal Processing Conference (EUSIPCO)*, pp. 1–5, 2019.
 16. **K. Papadimitriou** and G. Potamianos, “End-to-end convolutional sequence learning for ASL fingerspelling recognition,” *Proceedings of the INTERSPEECH*, pp. 2315–2319, 2019.
 17. **K. Papadimitriou** and G. Potamianos, “A hybrid approach to hand detection and type classification in upper-body videos,” *Proceedings of the European Workshop on Visual Information Processing (EUVIP)*, pp. 1–6, 2018.

C. Technical Reports

1. G. Potamianos, **K. Papadimitriou**, E. Efthimiou, S.-E. Fotinea, P. Maragos, and G. Sapountzaki, “D6.4: Publicity and dissemination report for Y3,” *Technical Report, SL-ReDu Project Deliverable*, Volos, Greece, 2023 [[“Public SL-ReDu Project Deliverable”](#)].

2. G. Sapountzaki, G. Potamianos, **K. Papadimitriou**, E. Efthimiou, and S.-E. Fotinea, “D5.6: Evaluation of second system version,” *Technical Report, SL-ReDu Project Deliverable*, Volos, Greece, 2023 [“Restricted SL-ReDu Project Deliverable”].
3. S.-E. Fotinea, E. Efthimiou, T. Goulas, G. Sapountzaki, **K. Papadimitriou**, and G. Potamianos, “D4.2: Second version of human-computer interface,” *Technical Report, SL-ReDu Project Deliverable*, Volos, Greece, 2023 [[“Public SL-ReDu Project Deliverable”](#)].
4. **K. Papadimitriou**, G. Potamianos, E. Efthimiou, and S.-E. Fotinea, “D2.3: Final version of GSL recognizer,” *Technical Report, SL-ReDu Project Deliverable*, Volos, Greece, 2023 [“Restricted SL-ReDu Project Deliverable”].
5. E. Efthimiou, S.-E. Fotinea, G. Sapountzaki, and **K. Papadimitriou**, “D3.5: Second version of linguistic model,” *Technical Report, SL-ReDu Project Deliverable*, Volos, Greece, 2023 [“Restricted SL-ReDu Project Deliverable”].
6. **K. Papadimitriou**, G. Potamianos, E. Efthimiou, and S.-E. Fotinea, “D1.3: Final version of visual tracking and feature extraction components,” *Technical Report, SL-ReDu Project Deliverable*, Volos, Greece, 2022 [“Restricted SL-ReDu Project Deliverable”].
7. E. Efthimiou, S.-E. Fotinea, P. Maragos, G. Sapountzaki, and **K. Papadimitriou**, “D3.4: Second version of data resources,” *Technical Report, SL-ReDu Project Deliverable*, Volos, Greece, 2022 [“Restricted SL-ReDu Project Deliverable”].
8. G. Potamianos, **K. Papadimitriou**, G. Sapountzaki, E. Efthimiou, S.-E. Fotinea, and P. Maragos, “D7.2: Management report for Y2,” *Technical Report, SL-ReDu Project Deliverable*, Volos, Greece, 2022 [[“Public SL-ReDu Project Deliverable”](#)].
9. G. Potamianos, **K. Papadimitriou**, E. Efthimiou, S.-E. Fotinea, P. Maragos, and G. Sapountzaki, “D6.2: Publicity and dissemination report for Y2,” *Technical Report, SL-ReDu Project Deliverable*, Volos, Greece, 2022 [[“Public SL-ReDu Project Deliverable”](#)].
10. G. Sapountzaki, G. Potamianos, **K. Papadimitriou**, and E. Efthimiou, “D5.4: Evaluation of first system version,” *Technical Report, SL-ReDu Project Deliverable*, Volos, Greece, 2022 [“Restricted SL-ReDu Project Deliverable”].

11. **K. Papadimitriou**, G. Potamianos, E. Efthimiou, S.-E. Fotinea, and P. Maragos, “D2.2: Intermediate version of GSL recognizer,” *Technical Report, SL-ReDu Project Deliverable*, Volos, Greece, 2021 [“Restricted SL-ReDu Project Deliverable”].
12. S.-E. Fotinea, E. Efthimiou, T. Goulas, C. Flouda, G. Ametoglou, G. Sapountzaki, **K. Papadimitriou**, and G. Potamianos, “D4.1: First version of human-computer interface,” *Technical Report, SL-ReDu Project Deliverable*, Volos, Greece, 2021 [“[Public SL-ReDu Project Deliverable](#)”].
13. **K. Papadimitriou**, G. Potamianos, E. Efthimiou, S.-E. Fotinea, and P. Maragos, “D1.2: Intermediate version of visual tracking and feature extraction components,” *Technical Report, SL-ReDu Project Deliverable*, Volos, Greece, 2021 [“Restricted SL-ReDu Project Deliverable”].
14. G. Potamianos, **K. Papadimitriou**, E. Efthimiou, S.-E. Fotinea, T. Goulas, C. Flouda, and G. Ametoglou, “D5.2: Technical specifications and system architecture definition,” *Technical Report, SL-ReDu Project Deliverable*, Volos, Greece, 2021 [“[Public SL-ReDu Project Deliverable](#)”].
15. G. Potamianos, **K. Papadimitriou**, G. Sapountzaki, E. Efthimiou, S.-E. Fotinea, and P. Maragos, “D7.1: Management report for Y1,” *Technical Report, SL-ReDu Project Deliverable*, Volos, Greece, 2021 [“[Public SL-ReDu Project Deliverable](#)”].
16. G. Potamianos, **K. Papadimitriou**, E. Efthimiou, S.-E. Fotinea, P. Maragos, and G. Sapountzaki, “D6.1: Publicity and dissemination report for Y1, including project website,” *Technical Report, SL-ReDu Project Deliverable*, Volos, Greece, 2021 [“[Public SL-ReDu Project Deliverable](#)”].
17. E. Efthimiou, S.-E. Fotinea, G. Sapountzaki, and **K. Papadimitriou**, “D3.1: First version of data resources,” *Technical Report, SL-ReDu Project Deliverable*, Volos, Greece, 2021 [“Restricted SL-ReDu Project Deliverable”].
18. **K. Papadimitriou**, G. Potamianos, E. Efthimiou, S.-E. Fotinea, and P. Maragos, “D2.1: First version of GSL recognizer,” *Technical Report, SL-ReDu Project Deliverable*, Volos, Greece, 2021 [“[Public SL-ReDu Project Deliverable](#)”].

19. **K. Papadimitriou**, G. Potamianos, E. Efthimiou, S.-E. Fotinea, and P. Maragos, “D1.1: First version of visual tracking and feature extraction components,” *Technical Report, SL-ReDu Project Deliverable*, Volos, Greece, 2020 [[“Public SL-ReDu Project Deliverable”](#)].