Recognizing Visual Communication with Limited Supervision



Supervised by Professor Andrew Zisserman

Prajwal Kondajji Renukananda

St. Anne's College

University of Oxford

A thesis submitted for the degree of

Doctor of Philosophy

Trinity Term 2024

Abstract

Understanding human communication is the first step towards enabling most human-machine interactions. Deep learning models of today can accurately transcribe and reason across spoken and written forms of human communication. In this work, we focus on the third frontier in human communication — the visual language, specifically the signals communicated using our lips and hands.

In the first part of this work, we focus on transcribing lip movements to text. We develop new attention-based visual features to track the lip region in a dynamic talking head video. Next, we leverage recent advances in automatic speech recognition (ASR) and show that we can adapt a state-of-the-art ASR model for lip-reading. Finally, we extend lip reading to languages beyond English using a new large-scale multilingual dataset with weak ASR labels.

In the second part of this thesis, we propose Transpotter, a new architecture that can localize the word being uttered in the silent video. It outperforms previous methods by a large margin – and demonstrates strong application in spotting mouthings in sign language videos.

In the third part of this work, we shift our attention to sign language and solve several problems in pursuit of our goal to recognize continuous signing sequences. We develop automatic methods to densely annotate a large corpora of signing videos with word labels. We also design a novel approach to recognize an important class of signs — Fingerspelling. Finally, we move beyond word-level recognition to tackle continuous sign language recognition in large vocabulary settings.

In the final part of the thesis, we expand our area from signing gestures to co-speech gestures. We propose JEGAL, a joint embedding space to associate hand gestures to the words being uttered. We demonstrate the efficacy of these representations in strong downstream applications.

Keywords – lip reading, sign language, gesture understanding

This thesis is submitted to the Department of Engineering Science, The University of Oxford, in fulfilment of the requirements for the degree of Doctor of Philosophy. This thesis is entirely my own work, and except where otherwise stated, describes my own research.

Prajwal Kondajji Renukananda, October 2024.

Acknowledgement

First and foremost, I offer my deepest gratitude to my supervisor Andrew Zisserman, who has been a constant guiding light throughout my PhD. I will forever be indebted for his support and trust in all my endeavours for the last four years. His wisdom, infectious enthusiasm, and attention to detail has played a pivotal role in realising this thesis.

To Triantafyllos Afouras and Liliane Momeni, whose friendship has made these four years feel effortless. They have made me feel right at home from the very start, and are the people I turned to when in need to help and motivation. To Samuel Albanie, Gül, Hannah Bull, and Charles Raude for showing me how fun research can be when you work together as a team. To Sindhu Hegde, who has always shared my highs and lows, and has inspired me through her dedication and hard work. Your energy and passion knows no bounds, thank you for making my PhD memorable.

I am privileged to be a part of the remarkable Visual Geometry Group. I am profoundly grateful to each of you for cultivating such a warm, supportive and collaborative research environment. In particular, to Jaesung Huh, Ragav Sachdeva, Tengda Han, Guanqi Zhan, Prasanna Sridhar, Yash Bhalgat, Sagar Vaze, Chuhan Zhang, Max Bain, Luke Melas-Kyriazi, Bruno Korbar, and Andrew Brown for the special times we spent together in Oxford. To Ashish Thandavan, who is the pillar of our research endeavours.

To my internship collaborators and mentors at Meta, thank you for expanding my horizons. To Bowen Shi, Wei-Ning Hsu, Apoorv Vyas, Andros Tjandra, Matthew Le, and Huiyu Wang for their generous time and invaluable insights. I owe the success of my internship to them. To my dear friends, Venkkatesh Sekar and Harish Rithish for their unwavering friendship that makes this journey all the more meaningful.

Finally and most importantly, to my family, for their unconditional love and being the backbone of all my endeavours. To my parents, Renukanand and Roopa, for their constant encouragement regardless of their hardships. Thank you for the countless sacrifices you have made, for protecting me at every turn in life. To my sister, Priyanka, who is always puts my well-being before anything else. The three of you have always been and always will be my biggest source of strength and inspiration.

Contents

1 Introduction **12**

1.1 Background and Challenges 14

1.1.1 Lip Reading 14

1.1.2 Visual Keyword Spotting 16

1.1.3 Sign Language Recognition 16

1.1.4 Learning Co-speech Gesture Representations 18

1.2 Thesis Outline and Contributions 19

1.2.1 Lip Reading 19

1.2.2 Visual Keyword Spotting 21

1.2.3 Sign Language Recognition 22

1.2.4 Learning Co-speech Gesture Representations 23

1.3 Publications 24

I Lip Reading **25**

2 Lip Reading with Visual Attention **26**

2.1 Introduction 28

2.2 Related Work 30

2.3 Method 33

2.3.1 Visual backbone 33

2.3.2 Training 36

2.4 Experiments 37

2.4.1 Data . . . 37
2.4.2 Implementation details . . . 38
2.4.3 State-of-the-art lipreading . . . 39
2.4.4 Ablations . . . 40
2.4.5 Visual attention visualization . . . 41
2.5 Visual Speech Detection Application . . . 42
2.6 Conclusion . . . 45

3 Speech Recognition Models are Strong Lip-readers 46
3.1 Introduction . . . 47
3.2 Related Work . . . 49
3.2.1 Lip reading . . . 49
3.2.2 Visual language Identification . . . 50
3.3 Method . . . 50
3.3.1 Visual backbone . . . 50
3.3.2 Cross-modal mapping network . . . 51
3.3.3 Conditioning the ASR with visual inputs . . . 51
3.3.4 Training objective . . . 53
3.4 Experiments . . . 53
3.4.1 Data . . . 53
3.4.2 Implementation details . . . 54
3.5 Results . . . 54
3.5.1 Quantitative scores . . . 54
3.5.2 Whisper decoder's cross-attention heatmaps . . . 56
3.5.3 Ablation studies . . . 56
3.6 Extending to Visual Language Identification . . . 57
3.7 Conclusion and extensions . . . 58

4 Scaling Multilingual Visual Speech Recognition 60

4.1 Introduction . . . 62
4.2 Related Work . . . 63
4.3 A New Large-scale Multilingual Dataset . . . 64
4.3.1 Data collection and pre-processing . . . 65
4.3.2 Automatic labelling . . . 66
4.3.3 Data Splits . . . 66
4.4 Multi-Task Network . . . 67
4.4.1 Implementation details . . . 67
4.5 Experiments . . . 68
4.5.1 Datasets . . . 68
4.5.2 Lip-reading Results . . . 68
4.5.3 Visual Language Identification (VLID) Results . . . 69
4.5.4 Extending to new low-resource languages . . . 70
4.5.5 Monolingual v/s Multilingual v/s Multilingual with additional fine-tuning . . . 71
4.6 Limitations and Conclusion . . . 71

II Visual Keyword Spotting 72

5 Visual Keyword Spotting with Attention 73
5.1 Introduction . . . 74
5.2 Related work . . . 76
5.3 Visual KWS with Attention . . . 78
5.3.1 The Transpotter Architecture . . . 78
5.3.2 Training . . . 80
5.3.3 Discussion . . . 81
5.3.4 Implementation details . . . 81
5.4 Experiments . . . 82
5.4.1 Datasets and Evaluation Protocol . . . 82

5.4.2 Comparison to baselines . . . 83
5.4.3 Architecture ablations . . . 84
5.4.4 Transpotter performance analysis . . . 85
5.5 Mouthing Spotting in Sign Language videos . . . 87
5.6 Conclusion . . . 88

III Sign Language Recognition 90

6 Automatic Dense Annotation of Large-vocabulary Sign Language Videos 91
6.1 Introduction . . . 93
6.2 Related Work . . . 95
6.3 Densification . . . 98
6.3.1 Mining more Spottings through In-domain Exemplars (E) . 99
6.3.2 Discovering Novel Sign Classes (N) . . . 100
6.3.3 Pseudo-labelling as a Form of Sign Spotting (P) . . . 101
6.3.4 Improving the Old (M*, D*) . . . 101
6.3.5 Evaluation Framework . . . 103
6.4 Experiments . . . 105
6.4.1 Data and Evaluation Protocol . . . 105
6.4.2 Results . . . 106
6.5 Conclusion . . . 110

7 Weakly-supervised Fingerspelling Recognition in British Sign Language Videos 112
7.1 Introduction . . . 114
7.2 Related work . . . 116
7.3 Fingerspelling detection and recognition . . . 118
7.3.1 The Transpeller architecture . . . 118

7.3.2 Multiple Hypotheses (MH) CTC loss . . . 120
7.4 Automatic annotations . . . 121
7.4.1 Exemplar and mouthing annotations . . . 122
7.4.2 Transpeller annotations . . . 123
7.4.3 Multi-stage training . . . 123
7.5 Experiments . . . 124
7.5.1 BOBSL Fingerspelling benchmark . . . 124
7.5.2 Results . . . 124
7.5.3 Architecture ablations . . . 127
7.6 Conclusion . . . 129

8 A Large-Vocabulary Continuous Sign Language Recognition Benchmark 130

8.1 Introduction . . . 132
8.2 Related Work . . . 134
8.3 A New CSLR Evaluation Benchmark . . . 136
8.4 CSLR^2: A Joint Model for CSLR and Retrieval . . . 140
8.4.1 Model overview and inference . . . 140
8.4.2 Training with sentence- and sign-level losses . . . 141
8.5 Experiments . . . 143
8.5.1 Baselines . . . 143
8.5.2 CSLR^2 ablation study . . . 144
8.5.3 Comparison to baselines . . . 145
8.5.4 Qualitative analysis . . . 145
8.6 Conclusion . . . 146
8.7 BOBSL-CSLR: Annotation . . . 147
8.8 BOBSL-CSLR-TEST Dataset Analysis and Statistics . . . 149
8.9 CSLR^2 Implementation Details . . . 151
8.9.1 Sources of supervision . . . 151

8.9.2 Architecture . . . 152
8.10 Experiments . . . 154
8.10.1 CSLR baselines . . . 154
8.10.2 Oracle CSLR performance . . . 155
8.10.3 CSLR performance breakdown and analysis . . . 155
8.11 CSLR^2 Ablations . . . 156
8.11.1 Text encoder . . . 157
8.11.2 Synonyms for CSLR . . . 157
8.11.3 Weighting of losses on CSLR performance . . . 157
8.11.4 Subtitle alignment . . . 158
8.12 Qualitative Examples . . . 160

IV Learning Co-speech Gesture Representations 162

9 Understanding Co-speech Gestures in-the-wild 163
9.1 Introduction . . . 164
9.2 Related Work . . . 167
9.3 Method . . . 170
9.3.1 Overview . . . 170
9.3.2 Gesture Encoder . . . 170
9.3.3 Text Encoder . . . 171
9.3.4 Audio Encoder . . . 171
9.3.5 Fusion Block . . . 171
9.3.6 Gesture-Word Alignment . . . 171
9.3.7 Training Objective . . . 172
9.3.8 Implementation Details . . . 173
9.3.9 Training Datasets . . . 173
9.4 Downstream Tasks and Evaluation . . . 174
9.4.1 Cross-modal Retrieval . . . 175

9.4.2 Spotting Gestured Words . . . 175
9.4.3 Active Speaker Detection . . . 176
9.4.4 AVSpeech Test Benchmarks . . . 177
9.5 Results . . . 177
9.5.1 Baselines . . . 177
9.5.2 Cross-modal Retrieval . . . 178
9.5.3 Spotting Gestured Words . . . 178
9.5.4 Active Speaker Detection . . . 180
9.6 Insights and Ablations . . . 181
9.6.1 Speech v/s Text Modalities . . . 181
9.6.2 Impact of loss functions . . . 182
9.6.3 Fusion techniques . . . 183
9.7 Conclusion . . . 183
9.8 Most Gestured Words . . . 184
9.9 Model Details . . . 184
9.10 Dataset Visualization . . . 186
9.11 Qualitative Results . . . 187
9.12 Limitations and Areas of Improvement . . . 187
9.13 Potential Negative Societal Impacts . . . 188

10 Discussion **191**
10.1 Achievements and Impact . . . 191
10.2 Ethical Considerations . . . 193
10.3 Limitations and Future Work . . . 195
10.3.1 Lip reading and Visual Keyword Spotting . . . 195
10.3.2 Sign Language . . . 196
10.3.3 Learning Co-speech Gesture Representations . . . 197
10.4 Conclusion . . . 197

References **199**

11 Statement of Authorship **237**

Chapter 1

Introduction

Humans are multi-sensory beings, continuously processing information through a combination of senses. Sight and sound are particularly dominant, as they provide rapid, dense inputs for day-to-day activities and social interactions. Our auditory processing enables us to convey complex meaning through spoken words, facilitating real-time information flow. Naturally, in our quest to build machines that function alongside us, we have constantly developed intelligent systems that are capable of deciphering human speech [Baevski et al. 2020; Radford et al. 2022], understanding the linguistic content [Touvron et al. 2023; Brown et al. 2020], and even responding with human-like voices [Le et al. 2023; Chengyi Wang et al. 2023; Casanova et al. 2022]. With speech and language being the seemingly indispensable modalities for communication, the importance of *sight* in communication has flown a bit under the radar.

Visual communication, or non-verbal cues, has always enriched spoken language by adding depth, emphasis, and clarity. Lip movements help us discern words in noisy environments, while gestures—like raising a clenched fist to rally a crowd or placing a hand on the chest to express gratitude—amplify the emotional significance of what's being conveyed. Visual communication, expressed through movements of face and body, is thus a quintessential information stream that machines need to understand.

Understanding visual communication requires fine-grained comprehension, a task that remains challenging today but was especially difficult before 2016. Video datasets were scarce compared to image [Deng et al. 2009; T.-Y. Lin et al. 2014; Everingham et al. 2010; Everingham et al. 2015] and speech datasets [Panayotov et

al. 2015] and training with videos was computationally expensive. But, by the year 2016-17, speech and image models started approaching human-level accuracies in recognition performance. Image-related tasks such as image classification [K. He et al. 2016] and image segmentation [Girshick 2015; Redmon et al. 2016] were reaching accuracies good enough to be used in commercial systems for the masses. Speech recognition models achieved single-digit word error rates [Amodei et al. 2016], with companies like Google, Amazon, and Microsoft releasing production-grade APIs [Juang and Rabiner 2005] by 2016. The attention of the research community was increasingly being given to video understanding, as evidenced with the release of large-scale video datasets starting around 2016-17: YouTube-8M [Abu-El-Haija et al. 2016], Kinetics [J. Carreira and A. Zisserman 2017], something-something [R. Goyal et al. 2017] and so on. Even in 2016, it was clear that the future of digital content and human interactions was going to be video[1][2], and with the progress being made in video understanding, it was time for models to be able to understand visual communication signals in addition to speech.

Lip movements are the most direct visual signals that closely correlate with spoken words, offering a redundant stream of information that aids in recognizing speech, especially when audio is corrupted or absent. This is, in fact, the foundation of Audio-Visual Speech Recognition (AVSR), a field that has become crucial for improving the accuracy and robustness of speech recognition systems by leveraging visual cues [Afouras et al. 2019; Makino et al. 2019; Stavros Petridis et al. 2018b; B. Shi et al. 2022c; J. S. Chung et al. 2017; J. S. Chung and Andrew Zisserman 2016a]. Beyond simple transcription, lip movements offer strong, clear signals that enhance performance in long-standing speech tasks such as Active Speaker Detection [Afouras et al. 2020c; J. Roth et al. 2020] – identifying who is speaking – and source separation, which involves isolating a speaker's voice from background noise [Afouras et al. 2018c; Sindhu B. Hegde et al. 2021; Ephrat et al. 2018]. But what if there's no audible speech at all? Can we transcribe words purely from lip movements? Lip reading, or Visual Speech Recognition, is the most challenging of these tasks, but in this thesis, we demonstrate that it can be achieved at scale.

Another facet of visual communication that is closely tied to speech is hand gestures. While speech and lip movements go hand-in-hand, co-speech hand gestures

[1] www.socialpilot.co/blog/video-marketing-statistics
[2] www.90seconds.com/blog/stats-2016-video-trends

can be sparse – only a few words in a sentence might correlate with the gestures. Unlike lip movements, gestures add a new layer of meaning to what is being said. They can emphasize particular words, illustrate what is being said, e.g. arms wide apart to signify “huge", or pointing gesture to convey what the speaker means by “over there". Learning to associate hand gestures to words/phrases has not been explored on a large-scale before. The learned representations can be useful for several downstream tasks such as gesture generation, video-subtitle alignment, and active-speaker detection using hand gestures.

Both lip movements and gestures are strongly tied to the spoken language. The visual stream is not the primary channel for information flow, which raises the question: are there any forms of visual communication that are purely ‘visual’. The most clear example of such a “visual-only" communication mode is sign language. Sign languages exist independently of spoken languages, and are communicated using hand shapes, location and motion, along with eyebrows, mouth, head and shoulders [Ronnie B Wilbur 2013]. In the realm of deep learning, the area of sign language understanding largely remains unsolved due to the scarcity of labeled training data. In this thesis, we show that we can learn to recognize signs using noisy automatically labeled data. This is a crucial step in reducing the gap between spoken and sign languages.

1.1 Background and Challenges

In this thesis, we focus on understanding three specific domains in visual communication. The first part is lip movements – the visual articulations that are directly linked to speech. The second is sign languages, a purely visual language used by over 70 million people worldwide [*World Federation of the Deaf* n.d.]. And the third, is co-speech hand gestures that complement uttered speech. In this section, we describe these three domains in detail, along with the challenges they entail, and why they are important research problems to solve.

1.1.1 Lip Reading

Recognizing what is spoken using only the lip movements is a long-standing problem [Hao et al. 2020]. While the basic theory was established in 1954 [Sumby and

Pollack 1954], it was shown only in 1984 [Petajan 1984] that lip features can make speech recognition models more robust. However, even until 2016, most of the results [Goldschen et al. 1994; Goldschen et al. 1997; Noda et al. 2014; Wand et al. 2016] were on videos recorded in laboratory settings, with limited vocabulary, and a handful of identities. The release of the LRW dataset [J. S. Chung and Andrew Zisserman 2016a] changed the trajectory of the field, allowing deep learning models that scale well with data, to be benchmarked on real-world videos. The dataset contained 500 words with 1000 instances of each word. Thus the models trained on this data were word-level – they could label each short video clip to contain any one of five hundred words in the LRW training dataset.

Despite promising results on a real-world video benchmark, it must be noted lip reading isolated words is incredibly difficult. This is because lip reading is inherently ambiguous [J. S. Chung and Andrew Zisserman 2016a; Afouras et al. 2019]. Very different-sounding words, can be uttered with near-identical lip movement patterns: "bat, mat, pat", "chop, shop, job". Recognizing such words in isolation, would be nearly impossible. However, it is quite obvious that these words usually occur in vastly different contexts; one would never say, "I bought milk at the grocery chop/job". Lip reading sentences using the surrounding context is far easier as it can help disambiguate such homophemes. Consequently, two of the most popular lip reading datasets, LRS2 [Afouras et al. 2019; J. S. Chung et al. 2017], and LRS3 [Afouras et al. 2018b] used even today, consist of sentence-level clips to evaluate sentence-level lip reading in the wild.

The LRS2 and LRS3 benchmarks paved the way for a rapid progress in lip reading, particularly using increasingly stronger sequence processing models starting with LSTMs [J. S. Chung and Andrew Zisserman 2016a], Transformers [Afouras et al. 2019] and Conformers [Ma et al. 2021c]. In this work, our first focus is to develop strong visual features for lip reading to complement the advances in the sequence processing module. We show that we can use visual attention to track the lip region in a dynamic real-world video. Another question that arises is, how do we scale models beyond the LRS2 and LRS3 datasets? Obtaining manual transcripts for videos is expensive and impractical when expanding to other low-resource languages. We show two ways to use a pre-trained speech recognition model to improve lip reading performance while being highly efficient in the use of manually labeled data.

The improvements in lip reading accuracy enables several unique applications like silent dictation, or dubbing and transcribing archival silent films. Lip reading models can help speech impaired individuals, or people with aphonia (loss of voice) to communicate just by using lip movements.

1.1.2 Visual Keyword Spotting

Lip reading transcribes every word that is uttered in a video segment. However, for many applications, it could be sufficient to only recognize specific utterances or keywords. Automatic visual keyword spotting has distinctive applications: (i) searching through silent archival videos, (ii) triggering virtual assistants with keywords (e.g. Hey Siri), (iii) locating the presence of specific words in videos, e.g. mouthings in sign language clips. In this thesis, we build upon the advancements in lip reading in terms of visual and sequence processing and design a single unified model that can (i) detect the presence (absence) of a query word in a given video clip, and (ii) localize all instances of the word in time.

Although visual keyword spotting and lip reading are closely related in terms of the features they rely on, each is unique in its own way. Visual KWS is relatively easier, because by specifying exactly "what" to search for, the search space for the model reduces, which can be crucial, especially in such an ambiguous task. A keyword spotting model does not need to care about all the surrounding words, and is thus less sensitive to recognition errors in the words we do not care about. For the same reason, keyword spotting models are more robust in localizing a word compared to lip reading models when there are silent segments and non-speech related mouth movements in the video. This is crucial because, one of the big applications for visual keyword spotting is to spot mouthings in sign language videos and associate them with spoken language word labels [Momeni et al. 2020a; Momeni et al. 2020b] – something that is invaluable to creating large-scale sign-language datasets. Thus, visual KWS serves as a natural extension to lip-reading with a strong suite of applications at which it is better at than lip reading.

1.1.3 Sign Language Recognition

Lip movements are innately tied to the spoken language, and in this work, we also explore languages that are purely visual. Sign languages are spatio-temporal lan-

guages that exist independently of spoken languages. There are over 200 distinct sign languages today, used by over 70 million individuals[3].

The visual aspect of sign language in itself makes it interesting to model and learn, but the biggest motivation is to enable seamless translation between the different sign languages and the spoken languages. The world is becoming increasingly inclusive by breaking down language barriers. Multiple works over the last few years have explored the task of "massively-multilingual translation". In the context of text-to-text translation, one of the initial large-scale works [Aharoni et al. 2019] trained a model that can translate between English and 102 other languages. But this was scaled further [Costa-jussà et al. 2022] to enable translation between 200 languages in 40K translation directions. SeamlessM4T [Barrault et al. 2023a] went even further by performing speech-to-speech translation. The dream of a universal translator is becoming more and more of a reality, but sign languages have never been discussed in this context.

In this work, we build foundations to enable sign language translation. One of the biggest obstacles for sign language translation is the lack of labeled training data. This is in contrast to translation efforts [Aharoni et al. 2019; Costa-jussà et al. 2022; Barrault et al. 2023a] in other languages, where there are several millions of paired examples. To start with, sign languages are low-resource, so there is a limited number of signing videos available online. Manual annotations for these videos are almost non-existent, because signing videos rarely co-occur with spoken language text or speech. Finally, it is incredibly expensive to manually label signing videos for training models, as the labels need to come from proficient annotators who know the language.

Solving sign language translation with a limited labeled pairs is bound to yield unsatisfactory results [N. Camgoz et al. 2021]. Instead we focus on the task that can be a pre-cursor to translation which is sign recognition. We first scale up isolated sign recognition by obtaining dense automatic annotations for the unlabeled signing videos. The mouthings from our visual keyword spotting model prove to be crucial here. The automatic labels, although noisy actually enable us to solve various tasks. We demonstrate state-of-the-art recognition results for a particular sign type called fingerspelling, using such noisy labels for training. We show that these dense labels can be used to train for continuous sign recognition, i.e.

[3]https://wfdeaf.org/our-work

predicting a word label for each sign in a clip.

Our approach to enabling sign language recognition on real-world videos takes us a step closer to realizing sign language translation. While we circumvent the need for labeled data to an extent, it does come with its own limitations. We only work with interpreted British Sign Language (BSL) videos [Albanie et al. 2021b], which allows us to obtain automatic labels with the help of cues from the spoken language subtitles. Compared to conversational signing, such interpreted data can introduce bias, however, we believe that interpreted data serves as an opportunity to learn sign representations that can later be adapted to the domain of conversational signing.

1.1.4 Learning Co-speech Gesture Representations

Besides sign language, hand gestures can also co-occur with speech, similar to lip movements. Hand gestures are an essential part of verbal communication. Gestures come in different forms [Andric and Small 2012], such as *beat* gestures to emphasize certain words of the speech, and iconic or deitic gestures that illustrate content of the speech. Understanding these co-speech gestures is essential to fully understand what the speaker is saying. For example, imagine we say this sentence to an autonomous robot, "fetch the glass of milk on the table over there". If the robot does not understand the pointing gesture where we say which table we are referring to, it will not be able to perform the task. Detecting the important gestures and associating them to spoken words and phrases is the first step towards understanding these gestures.

Learning to associate gestures with text transcriptions of the accompanying speech is a very challenging task – the associations are typically sparse and ambiguous with a large degree of variability across speakers: Most of the time, very few words in the transcript are clearly gestured; the same sentence can be gestured very differently in different contexts and by different people; and the gestures depend on the speaker's emotion, culture, and social scenario. Moreover, not all gestures can be mapped to the semantics of the words.

We propose to learn gesture video representations from large-scale weak natural language supervision. The supervision is weak because we only use transcriptions of the speech; we do not have any information on which words in the transcript

are gestured. We demonstrate that our approach can be used to solve multiple gesture-related downstream tasks.

1.2 Thesis Outline and Contributions

In this section, we present an overview of the subsequent chapters in the thesis. The thesis is structured into four main parts: (i) Lip Reading, (ii) Visual Keyword Spotting, (iii) Sign Language Recognition, and (iv) Learning Co-speech Gesture Representations. For chapter 2 to 9, we outline the main contributions below. Chapter 10 explores the implications of this research and suggests potential avenues for future exploration.

1.2.1 Lip Reading

We start with the important task of lip reading, which serves as the foundation for the subsequent tasks in this thesis, such as visual keyword spotting and sign language recognition. These downstream tasks directly benefit from mouth representations learned in lip reading. In order to learn better mouth representations, we introduce a new module that automatically learns to isolate and track the lip region in a video. This capability is quite useful for both visual keyword spotting and spotting mouthings in signing videos.

In Chapter 2, we introduce the new lip tracking module called as Visual Transformer Pooling (VTP), to attend and track the lip region in a dynamic talking head video. Notably, the tracking capability is learned solely through the task of transcribing text from lip movements, without the need for any explicit tracking annotations. Along with the improved visual input representations, we adopt a different text tokenization strategy to learn better text representations. Specifically, we introduce sub-word tokenization into the field of lip reading for the first time. This makes the model more efficient by reducing the sequence length, and increases the granularity of text supervision from characters to sub-words and words. We train our new model on standard publicly available benchmarks like LRS2 [J. S. Chung et al. 2017], LRS3 [Afouras et al. 2018b], and show clear gains in recognition performance. But how do we scale beyond these two datasets? While we can obtain more videos, the most expensive part would be obtaining accurate

ground-truth transcripts for these videos. We solve this problem by showing that we can scale "for free" by training our model using just noisy text labels from speech recognition models. Finally, aside from improvements in WER, there is still a key missing piece before we can automatically transcribe a real-world silent video. The important pre-processing step is to detect the time segment when the person is speaking using just the lip movements. We call this task as "Visual Speech Detection" (VSD), and we show that our lip reading model can be extended for this task as well.

The fact that lip reading models can be scaled very well using noisy labels from pre-trained speech recognition models is not at all surprising. The two tasks are closely related, in fact, they only differ in the input modality. ASR transcribes acoustic speech, and VSR transcribes the visual counterpart of it, with the deeper implication being that they must be learning similar representations. Is there a possibility to share the modules across these two tasks? Chapter 3 investigates this "shared knowledge" idea. We show that we can adapt a pre-trained ASR model like Whisper [Radford et al. 2022] to the task of lip reading. ASR models have learned strong speech representations, and can accurately transcribe text sequences by cross-attending to these speech features. By mapping the lip reading features to these speech features, we enable an off-the-shelf pre-trained ASR model to lip read. This approach is far more data-efficient, and achieves significant gains as opposed to training a lip reading model from scratch. We use Whisper [Radford et al. 2022] as our pre-trained ASR. In addition to recognition, Whisper can also perform other tasks like identifying the spoken language from audio. Using the same approach as we used for lip reading, we report performance of identifying language from lip movements, a.k.a, visual language identification.

In Chapter 4, we extend these advancements to languages other than English. This is challenging because other languages do not have large-scale manually curated datasets. Compared to English, it is also relatively harder to find video content for other languages, e.g. Japanese, Greek. The next challenge would be to obtain reliable transcripts for these languages. Manually transcribing is unscalable, which means the other option is to resort to our technique in Chapter 2 of using ASR labeled transcripts. But, ASR models for other languages might not be anywhere close to the accuracy of English ASR. We circumvent these problems with three contributions: (i) building the largest multilingual lip-reading dataset

to date, (ii) a single multi-task architecture that can perform two tasks simultaneously: identify the language and transcribe the utterance, and (iii) jointly training this architecture on all the languages together, resulting in large WER improvements as opposed to training monolingual models separately. We also show that our multilingual model can be readily fine-tuned for new low-resource languages to achieve significant improvements when compared to training from scratch. Our approach uses zero manual annotations and achieves state-of-the-art performance across 7 languages.

1.2.2 Visual Keyword Spotting

Lip reading might not always be the best option to recognize what is being uttered in a silent video. One might be only interested in searching for a specific word or phrase in a video library. Or the video clip might not even contain free-flowing speech, in which case, lip reading models would perform poorly. In Chapter 5, we shift from lip reading to its "easier to handle" sibling task of visual keyword spotting, where the goal is to detect and localise a given keyword in a silent (spoken) video. We start with a strong foundation from lip reading (Chapter 2) by using strong visual features from VTP, and sub-word tokenization for the text inputs. We propose a novel architecture called the "Transpotter" that can (i) jointly model video-text relationships, and (ii) localize keywords to frame-level accuracy. We outperform previous methods [Momeni et al. 2020a] on two standard benchmarks.

We also demonstrate the practical importance of our model on an out-of-domain task of spotting mouthings in sign language videos. Signers often mouth the words they sign [Rachel Sutton-Spence and Woll 1999b], to convey different meanings, or to simply provide redundancy [Woll 2001]. This means that by localizing the mouthed word, we also localize the sign. If we have a large unlabeled dataset of interpreted signing content, with weakly-aligned subtitles, we can use *Transpotter* to localize the video segments where candidate words from the subtitle are mouthed. In other words, visual KWS models can be used to automatically label sign segments with word labels and create large-scale sign-language training datasets. *Transpotter* is twice as accurate for this out-of-domain task compared previous keyword spotting models [Momeni et al. 2020a].

1.2.3 Sign Language Recognition

In Chapter 6, we shift our focus to sign language, a purely visual form of communication in its own right. We already discussed how Transpotter can be used to spot mouthings and automatically annotate signing segments with word labels. Focusing on the British Sign Language (BSL), our first goal is to increase the density of automatic sign annotations in the largest publicly available BSL video dataset: BOBSL [Albanie et al. 2021b]. By *density*, we mean that (i) there are fewer temporal gaps (without annotations) in the signing sequence timeline, and (ii) more words from the spoken language subtitle are associated with a signing segment. We improve upon Transpotter (Chapter 5) using a dual-stage fine-tuning strategy. We automatically spot mouthings on the entire BOBSL training set, and fine-tune the model on signer mouthings by treating the high-confidence spottings as positives. We repeat this process two times. We also introduce a novel approach to increase our annotations, where, we use sign-representation similarities to propagate the labels from previously labeled examples throughout the video dataset.

We now have a large number of word-level sign annotations, allowing us to effectively train word-level sign recognition models. In Chapter 7, we focus on recognizing a specific class of signs called "Fingerspelling". Fingerspelling is a means to encode words from the written language into sign language via a manual alphabet, i.e. one sign per letter. Words from a written language with no known sign may be fingerspelled, such as names of people and places. In the BOBSL dataset [Albanie et al. 2021b], fingerspelling amounts to about 10% of all the signs. Recognizing such signs is crucial, not only because they constitute a significant proportion, but also due to the fact that they typically correspond to important keywords. We improve upon the ideas from our densification efforts in Chapter 6 to curate a weakly-labeled training data for fingerspelling detection and recognition. We introduce a novel architecture called *Transpeller* that can jointly perform three tasks: (i) classify a video segment to indicate the presence (absence) of fingerspelling, (ii) localize the fingerspelled segment in time, and (iii) recognize the letters that are fingerspelled. We introduce new techniques to better deal with the noisy automatic annotations. The first one, called as the Multiple Hypotheses CTC loss, extends the standard CTC loss [Graves et al. 2006] to deal with

multiple plausible ground-truth sequence labels. The second is to use a pseudolabeling strategy that incorporates information from the subtitle. We also release a new BSL fingerspelling benchmark consisting of 5K video clips with fingerspelling annotations.

Finally, in Chapter 8, we move beyond individual sign recognition and start predicting continuous sequence of signs. The goal is to predict time aligned and dense word predictions for each sign within a signing sequence. This is a crucial step towards realizing sign language translation. We introduce a new CSLR-test benchmark, which is the largest CSLR benchmark to-date, both in terms of vocabulary (5K different signs) and duration (6 hours). We train a model that learns a joint embedding space between spoken and sign language text. We demonstrate that training a joint model for CSLR and sign retrieval tasks improves the CSLR performance.

1.2.4 Learning Co-speech Gesture Representations

In the final chapter, we re-visit visual communication cues that occur alongside speech. Earlier, we focused on lip movements; now, in Chapter 9, our aim is to explore co-speech gestures and connect them with spoken words. Co-speech gesture understanding is a relatively new and under-explored area. One possible reason for this could be the sheer difficulty of the task: gestures are only sparsely correlated with the speech. As a result, some of the works have resorted to developing models for specific cases where the gestures are clear, e.g. weather narration [Rajeev Sharma et al. 2000; Kettebekov et al. 2003]. Recent works [Molchanov et al. 2016; Morency et al. 2007; Ghaleb et al. 2024] have tried to detect and recognize gestures in laboratory settings while using multiple stereo and IR cameras. The latest work on modelling co-speech gestures is GestSync [Sindhu B Hegde and Andrew Zisserman 2023], that learns gesture representations by solving for cross-modal synchronization with speech-audio. But this focuses more on prosody, intonation in the speech – our goal is to learn to associate gestures with words. We propose to learn gesture video representations from large-scale weak natural language supervision. The supervision is weak because we only use transcriptions of the speech; we do not have any information on which words in the transcript are gestured. We demonstrate that our approach can be used to solve multiple gesture-related

downstream tasks.

1.3 Publications

In this section, we list publication contributions. Excluding Chapter 4, 8, 9 which introduces ongoing work, Chapters 2 to 7 each contain a research paper which has been peer reviewed and accepted for publication in a conference. The published papers are included here without modifications, except for formatting changes. Additional implementation details for each paper can be found in the supplementary materials of their online versions. A statement of authorship is also presented for each paper in the Appendix. The papers included in the thesis are listed below.

Chapter 2: "Sub-word Level Lip Reading With Visual Attention" K R Prajwal, Triantafyllos Afouras, Andrew Zisserman. In Computer Vision and Pattern Recognition (CVPR), 2022.

Chapter 3: "Speech Recognition Models are Strong Lip-readers" K R Prajwal, Triantafyllos Afouras, Andrew Zisserman. In INTERSPEECH, 2024.

Chapter 4: "Scaling Multilingual Visual Speech Recognition" K R Prajwal*, Sindhu Hegde*, Andrew Zisserman. In ICASSP, 2025.

Chapter 5: "Visual Keyword Spotting with Attention" K R Prajwal*, Liliane Momeni*, Triantafyllos Afouras, Andrew Zisserman. In British Machine Vision Conference (BMVC), 2021.

Chapter 6: "Automatic dense annotation of large-vocabulary sign language videos" Liliane Momeni*, Hannah Bull*, **K R Prajwal***, Samuel Albanie, Gül Varol, Andrew Zisserman. In European Conference on Computer Vision, 2022.

Chapter 7: "Weakly-supervised Fingerspelling Recognition in British Sign Language" K R Prajwal*, Hannah Bull*, Liliane Momeni*, Samuel Albanie, Gül Varol, Andrew Zisserman. In British Machine Vision Conference, 2022.

Chapter 8: "A Large-Vocabulary Continuous Sign Language Recognition Benchmark" Charles Raude*, **K R Prajwal***, Liliane Momeni*, Hannah Bull, Samuel Albanie, Andrew Zisserman, Gül Varol. Work in progress.

Chapter 9: "Understanding Co-speech Gestures in-the-wild" Sindhu Hegde, **K R Prajwal**, Taein Kwon, Andrew Zisserman. In International Conference on Computer Vision (ICCV), 2025.

Part I

Lip Reading

Chapter 2

Lip Reading with Visual Attention

The paper has been accepted for publication at the Computer Vision and Pattern Recognition (CVPR), 2022.

Sub-word Level Lip Reading With Visual Attention

K R Prajwal Triantafyllos Afouras Andrew Zisserman

Visual Geometry Group, University of Oxford

{prajwal,afourast,az}@robots.ox.ac.uk

https://www.robots.ox.ac.uk/~vgg/research/vtp-for-lip-reading

July 8, 2025

Abstract

The goal of this paper is to learn strong lip reading models that can recognise speech in silent videos. Most prior works deal with the open-set visual speech recognition problem by adapting existing automatic speech recognition techniques on top of trivially pooled visual features. Instead, in this paper, we focus on the unique challenges encountered in lip reading and propose tailored solutions. To this end, we make the following contributions: (1) we propose an attention-based pooling mechanism to aggregate visual speech representations; (2) we use sub-word units for lip reading for the first time and show that this allows us to better model the ambiguities of the task; (3) we propose a model for Visual Speech Detection (VSD), trained on top of the lip reading network. Following the above, we obtain state-of-the-art results on the challenging LRS2 and LRS3 benchmarks when training on public datasets, and even surpass models trained on large-scale industrial datasets by using an order of magnitude less data. Our best model achieves 22.6% word error rate on the LRS2 dataset, a performance unprecedented for lip reading models, significantly reducing the performance gap between lip reading and automatic speech recognition. Moreover, on the AVA-ActiveSpeaker benchmark, our VSD model surpasses all visual-only baselines and even outperforms several recent audio-visual methods.

2.1 Introduction

Lip reading, or visual speech recognition, is the task of recognising speech from silent video. It has many practical applications which include improving speech recognition in noisy environments, enabling silent dictation, or dubbing and transcribing archival silent films [Jha et al. 2019]. It also has important medical applications, such as helping speech impaired individuals, e.g. people suffering from Lou Gehrig's disease speak [Shillingford et al. 2019], or enabling people with aphonia (loss of voice) to communicate just by using lip movements.

Lip reading and audio-based automatic speech recognition (ASR) both have the common goal of transcribing speech, however, they differ regarding the input: while in ASR the input signal is an audio waveform, in essence, a one-dimensional time series, lip reading has to deal with high-dimensional video inputs that have both temporal and spatial complexity. This makes training large end-to-end models harder due to GPU memory and computation constraints. Furthermore, understanding speech from visual information alone is challenging due to the inherent ambiguities present in the visual stream, i.e. the existence of homophemes where different characters that are visually indistinguishable (e.g. 'pa', 'ba', and 'ma'). That lip reading is a much harder task is also supported by the fact that although humans can understand speech reasonably well even in the presence of noise and across a variety of accents, they perform relatively poorly on lip reading [J. S. Chung and Andrew Zisserman 2016a; Assael et al. 2016].

Designing a lip reading model requires both a visual component – mouth movements need to be identified – as well as a temporal sequence modelling component, which typically involves learning a language model that can resolve ambiguities in individual lip shapes. Recent developments in deep learning models and the availability of large-scale annotated datasets has led to breakthroughs surpassing human performance [J. S. Chung and Andrew Zisserman 2016a]. However, most of these works have taken the approach of adapting techniques used for ASR and machine translation, without catering to the particularities of the vision problem.

The conjecture in this paper is that the performance of lip reading, in terms of both accuracy and data efficiency, can be improved if the model is designed from the start taking into account the peculiarities of the visual, rather than the audio domain. To this end, we consider both the visual encoding and the text

tokenisation.

Visual encoding. Our first contribution is the design of a novel visual backbone for lip reading. The spatiotemporal complexity in lip reading requires dealing with problems such as tracking the mouth in moving talking heads. This is usually achieved with complicated pre-processing pipelines based on facial landmarks. However, those are sub-optimal in many cases. For example, landmarks don't work well in profile views [B. Johnston and Chazal 2018]. Moreover, it is unclear what is the optimal region-of-interest for lip reading: it has been shown that besides the lips, other parts of the face, e.g. the cheeks, may also contain useful discriminative information [Yuanhang Zhang et al. 2020]. Also, this region-of-interest can vary drastically in terms of scale, aspect ratio across identities and utterances. Thus, in this work, we propose an end-to-end trainable attention-based pooling mechanism that learns to track and aggregate the lip movement representations, resulting in a significant performance boost.

Text tokenisation. Lip reading methods most commonly output character-level tokens. This output representation however is sub-optimal as characters are sometimes more fine-grained than the input, with multiple characters corresponding to a single video frame. Furthermore, characters do not encode any prior knowledge about the language, which leads to higher dependency on the decoder's language modeling capacity that must also 'learn to read'. In this work, we instead use subword tokens (word-pieces) which not only match with multiple adjacent frames but are also semantically meaningful for learning a language easily. Word-pieces result in much shorter (than character) output sequences which greatly reduces the run-time and memory requirements. They also provide a language prior, reducing the language modelling burden of the model. We experimentally compare character and word-piece tokenization to justify this choice.

Visual Speech Detection. One issue with performing lip reading inference on real-world silent videos is that, since there is no audio track, there is no automated procedure for cropping out the clips where the person is speaking. ASR models use Voice Activity Detection (VAD) as a key pre-processing step, but this is clearly not applicable for silent videos. Here, the parts of a video containing speech have to be determined using the video input alone; in other words, by performing Visual Speech Detection (VSD). This can be very useful *e.g.* for running inference on silent movies. Among other findings in this work, we show that it is possible to

train a strong VSD model on top of our pre-trained lip reading encoder.

Other downstream tasks. Besides improving performance on the sentence-level lip reading task itself, obtaining improved lip movement representations can have a broader impact, as those are often used for other related downstream tasks – e.g. sound source separation [Ephrat et al. 2018], visual keyword spotting [Momeni et al. 2020a], and visual language identification [Afouras et al. 2020b].

In summary, we make the following three contributions: (i) a visual backbone architecture using attention-based pooling on the spatial feature map; (ii) the use of sub-word units, rather than characters for the language tokens; and (iii) a strong Visual Speech Detection model, directly trained on top of the lip reading encoder.

In the experiments we show the benefits of (i) and (ii) on improving lip reading performance, and we also introduce a two-stage training protocol that simplifies the curriculum used in prior works. As will be seen, with these design choices and training methodology, the performance of our best models exceeds prior work on standard evaluation benchmarks, and even outperforms proprietary models that use an order of magnitude more data for training. Similarly, we show the benefit of (i) and the lip reading encoder to our visual speech detection model that is far superior to previous methods on a standard evaluation benchmark.

Please check our project page for video examples, code, and pre-trained models.

2.2 Related Work

We present an overview of prior work on lip reading, including a discussion of how these methods select and track the visual regions of interest, as well as the output tokenizations they use, followed by a brief overview of the use of attention for visual feature aggregation in other domains.

Lip reading. Early works on lip reading relied on hand-crafted pipelines and statistical models for visual feature extraction and temporal modelling [G. Potamianos et al. 2003; Gowdy et al. 2004; Papandreou et al. 2009; Livescu et al. 2007; E.-J. Ong and Bowden 2011]; an extensive review of those methods is presented in [Ziheng Zhou et al. 2014]. The advent of deep learning and the availability of large-scale lip reading datasets such as LRS2 [J. S. Chung et al. 2017] and LRS3 [Afouras et al. 2018b], rejuvenated this area. Progress was initially on word-level recognition [J. S. Chung and Andrew Zisserman 2016a; Stafylakis and Tz-

imiropoulos 2017], and then moved onto sentence-level recognition by adapting models developed for ASR using LSTM sequence-to-sequence [J. S. Chung et al. 2017] or CTC [Assael et al. 2016; Shillingford et al. 2019] approaches. [Stavros Petridis et al. 2018b] take a hybrid approach, training an LSTM-based sequence-to-sequence model with an auxiliary CTC loss. One trend in recent work is moving to Transformer-based architectures [Afouras et al. 2019], or variants using convolution blocks [X. Zhang et al. 2019], and hybrid architectures like a Conformer [Gulati et al. 2020]. Another trend is to investigate the benefits of training with larger datasets, either directly by training on proprietary data that is orders of magnitude larger than any public dataset [Makino et al. 2019], or indirectly by distilling ASR models into lip reading ones [Afouras et al. 2020a; Yu et al. 2020; W. Li et al. 2019]. For visual feature extraction and short-term dynamic modelling, most modern pipelines rely on spatiotemporal CNNs consisting of multiple 3D convolutional layers [Assael et al. 2016; Shillingford et al. 2019], or more lightweight alternatives that comprise a single 3D convolutional layer followed by 2D ones [J. S. Chung and Andrew Zisserman 2016a; Stafylakis and Tzimiropoulos 2017; Afouras et al. 2019] applied frame-wise.

Mouth ROI selection, registration, and tracking. A thorough investigation on facial region of interest (ROI) selection for lip reading is provided by [Yuanhang Zhang et al. 2020]. The videos included in datasets like LRS2 and LRS3 are commonly pre-processed with a face detection and tracking pipeline which outputs clips roughly centered around the speaker's face. Many previous works use a central crop on the provided videos as input to the feature extractors [Stafylakis and Tzimiropoulos 2017; Afouras et al. 2019; Ma et al. 2021c]. More elaborate pipelines use facial landmarks to register the face to a canonical view and/or only extract the crops of the mouth area [Assael et al. 2016; Koumparoulis et al. 2017; Shillingford et al. 2019; Stavros Petridis et al. 2018a; Makino et al. 2019; X. Zhang et al. 2019]. [Yuanhang Zhang et al. 2020] propose to input a large part of the face, combined with Cutout[DeVries and Taylor 2017] to encourage the model to also use the extra-oral face regions. After selecting which input region to extract the low-level CNN features from, all above works apply Global Average Pooling (GAP) on the extracted visual features map; this obtains a compact representation, but discards spatial information. Recent works [X. Zhang et al. 2019] have shown that replacing GAP with a spatio-temporal fusion module improves performance.

Text tokenization. Most prior works on lip reading output character-level predictions [J. S. Chung et al. 2017; J. S. Chung et al. 2018; Afouras et al. 2019; X. Zhang et al. 2019; Stavros Petridis et al. 2018b; Ma et al. 2021c; Makino et al. 2019]. Those approaches usually use an external language model during inference to boost performance [Kannan et al. 2018; Maas et al. 2015]. Instead [Shillingford et al. 2019] chose to output phoneme sequences, using phonetic dictionaries. This approach has the advantage of a more accurate mapping of lip-movements to sounds but requires a complicated decoding pipeline involving a proprietary finite-state-transducer. [Koller et al. 2015b; Fenghour et al. 2020] use a hard-crafted heuristic to map words onto viseme sequences and vice versa, and use viseme tokens for representing the output and target text. In this work, we instead propose using sub-word level tokenisation, which greatly reduces the output sequence length, thus accelerating both training and inference, and neatly encodes prior language information that improves the overall performance.

Visual feature aggregation with attention. Our work is also related to methods that use attention for improving visual representations of images or videos. [X. Wang et al. 2018; Jetley et al. 2018] use attention-weighted-averages of visual features as building blocks for various classification and detection tasks, while OCNet [Yuhui Yuan and J. Wang 2018] uses self-attention to model context between pixels for semantic segmentation. Several recent papers have replaced convolutions with Transformer [Vaswani et al. 2017] blocks in visual representation pipelines. DETR [Carion et al. 2020] and Efficient DETR [Z. Yao et al. 2021] learn object detectors by applying spatial transformers on top of CNN feature extractors. Similarly, the Visual Transformer [B. Wu et al. 2020] tokenises low-level CNN features and then processes them using a Transformer to model relationships between tokens. ViT [Dosovitskiy et al. 2021] completely replaces CNNs in the visual pipeline with Transformer layers applied on image patch sequences, while the Timesformer [Bertasius et al. 2021] has been suggested as a purely Transformer-based solution for video representation learning.

Speech Detection. An important pre-processing stage in ASR pipelines is Voice Activity Detection (VAD), which involves the detection of the presence of speech in audio [Ramirez et al. 2007]. The reliability of audio-based VAD systems deteriorates in the presence of noise or cocktail party scenarios [Q. Liu et al. 2014]. In audio-visual pipelines, such as ones used for creating large-scale audio-visual

speech datasets [J. S. Chung and Andrew Zisserman 2016a; Afouras et al. 2018b], this step is commonly replaced by an Active Speaker Detection (ASD) stage which determines face tracks that match the speech. Audio-visual ASD models have been effectively trained either using direct supervision [J. Roth et al. 2020; Alcázar et al. 2020; J. S. Chung 2019; León-Alcázar et al. 2021; Tao et al. 2021] or in a self-supervised [J. S. Chung and Andrew Zisserman 2016a; Afouras et al. 2020c] fashion by employing contrastive objectives. The visual counterpart to VAD is Visual Speech Detection (VAD), which operates only on the video input. Early work on VSD (also termed Visual VAD or V-VAD) was based on handcrafted visual features and statistical modelling using methods such as HMMs, GMMs and PCA [Gerasimos Potamianos et al. 2004; P. Liu and Z. Wang 2004; Siatras et al. 2009; D. Sodoyer et al. 2006; David Sodoyer et al. 2009; A. Aubrey et al. 2007; Q. Liu et al. 2011; Q. Liu et al. 2014; Patrona et al. 2016]. More recent works proposed methods based on optical flow [A. J. Aubrey et al. 2010] or a combination of CNN and LSTMs [Rahul Sharma et al. 2019; Guy et al. 2021]. These methods are limited in having been trained or evaluated on constrained or non-public datasets. The train set of WildVVAD [Guy et al. 2021], a new annotated VSD dataset has been made public, however, at the time of publication of this paper, its test set was not available, we were, therefore, unable to use this dataset for benchmarking.

2.3 Method

In this section, we describe our proposed method. The architecture of the model is outlined in Figure 2.1. Next, we explain each stage of the pipeline.

2.3.1 Visual backbone

CNN. The input to the pipeline is a silent video clip of T frames, $\boldsymbol{x} \in \mathbb{R}^{T\times H\times W\times 3}$. A spatio-temporal residual CNN is applied on sub-clips of 5 frames (i.e. 0.2s) with a unit frame stride, to extract visual spatial feature maps $\boldsymbol{f} \in \mathbb{R}^{T\times h\times w\times c}$. For our best model, $H = W = 96$, $(h, w) = (H/4, W/4) = (24, 24)$, and $c = 128$.

Visual Transformer Pooling (VTP). The CNN feature map $\boldsymbol{f}_t \in \mathbb{R}^{hw\times c}$ corresponding to every input frame $t \in \{1, \ldots, T\}$ is processed individually by a shared Visual Transformer Pooling (VTP) block. The feature map is first flattened to

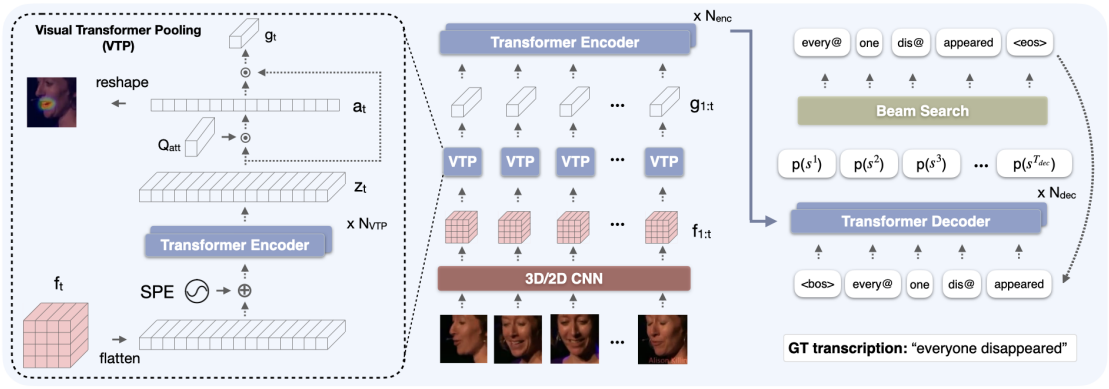


Figure 2.1: **Proposed lip reading architecture.** *Left:* The input video frames are passed through a spatio-temporal CNN to extract low-level visual features $\boldsymbol{f}$. The feature map corresponding to every input frame is then separately processed by a Visual Transformer Pooling module (VTP). The VTP block adds spatial positional encodings (SPE) to the input features and passes the result through a Transformer encoder to produce a self-attended feature map $\boldsymbol{z}_t$. A query vector $\boldsymbol{Q_{att}}$ is used to compute an attention mask which is in turn used to obtain a spatially weighted average of $\boldsymbol{z}_t$. This produces a compact visual representation of the lip appearance and movement around each input video frame. Concatenating the frame-wise features forms a temporal feature sequence $\boldsymbol{g}$. This is passed as input to an encoder-decoder Transformer (*right*) that auto-regressively predicts sub-word probabilities for one token at a time. An output sentence is eventually inferred from these distributions using a beam search.

$\boldsymbol{f}_t \in \mathbb{R}^{h \times w \times c}$ and projected to a desired Transformer feature dimension d to get $\boldsymbol{f}_t \in \mathbb{R}^{hw \times d}$. Then, spatial positional encodings (SPE) are added to it; the result is passed through an encoder consisting of N_{VTP} Transformer layers, to get an enhanced self-attended feature map

$$\boldsymbol{z}_t = encoder_v(\boldsymbol{f}_t + SPE_{1:hw}) \in \mathbb{R}^{hw \times d}.$$

A learnable query vector $\boldsymbol{Q_{att}} \in \mathbb{R}^{d \times 1}$ is then used to extract a visual attention mask

$$\boldsymbol{a}_t = softmax(\boldsymbol{Q}_{\boldsymbol{att}}^{\top} \boldsymbol{z}_t) \in \mathbb{R}^{hw \times 1}.$$

The attention mask is used to compute a weighted average over the self-attended feature map

$$\boldsymbol{g}_t = \frac{1}{hw} \sum_{u=1}^{hw} a_t^u z_t^u \in \mathbb{R}^d$$

where a_t^u and z_t^u denote the feature and attention weight respectively, associated with frame t and location $u \in \{1, \ldots, hw\}$. By stacking the resulting vectors $\boldsymbol{g}_t$ in time, we obtain an embedding sequence $\boldsymbol{g} = (\boldsymbol{g}_1, \boldsymbol{g}_2 \cdots, \boldsymbol{g}_T) \in \mathbb{R}^{T \times d}$ which contains a compact spatio-temporal representation for every input frame.

Transformer encoder-decoder. An encoder-decoder Transformer model is used to predict a text token sequence $s = (s_1, s_2 \cdots, s_{T_{dec}})$ from the source video embedding sequence $\boldsymbol{g}$, one token at a time: temporal positional encodings (PE) are added to $\boldsymbol{g}$, and the result is input to an encoder, which consists of N_{enc} multi-head Transformer layers, to produce a self-attended embedding sequence

$$\boldsymbol{g_{enc}} = \text{ENCODER}(\boldsymbol{g} + PE_{1:T}) \in \mathbb{R}^{T \times d}.$$

The decoder, which consists of N_{dec} Transformer layers, then attends on this sequence and predicts the output text token sequence s in an auto-regressive manner, by factorising its joint probability:

$$\log p(s|\boldsymbol{x}) = \sum_{t=1}^{T_{dec}} \log p(s_t|\ \boldsymbol{g_{enc}}(\boldsymbol{x}), s_{1:t-1}) \tag{2.1}$$

where positional encodings have also been added to the auto-regressive decoder inputs as in [Vaswani et al. 2017].

The text sentences are encoded into token sequences (and vice versa tokens are decoded into text) using a sub-word level tokeniser, in particular WordPiece [Yonghui Wu et al. 2016]. We tried other sub-word tokenizations such as Byte-Pair-Encoding (BPE) that is used in GPT2 [Radford et al. 2019], but it performed worse compared to using WordPiece.

Beam search decoding and rescoring. Decoding is performed with a left-to-right beam search of width B. We also decode a second time after flipping all the input video frames horizontally. Additional language knowledge is incorporated by using an external language model (LM) to re-score [Chan et al. 2016] the $2 \times B$-best hypotheses $S = \{s_1 \cdots s_B; s_1^h \cdots s_B^h\}$ from the beam searches, and obtain the highest scoring one as the final sentence prediction:

$$s_{best} = \arg\max_{s \in S} [\ \alpha \log p(s|\boldsymbol{x}) + (1 - \alpha) \log p_{LM}(s)\ ]$$

Here, $s_1^h \cdots s_B^h$ denotes the beam sequences after horizontally flipping the input. We found that additional test-time augmentations such as small degrees of rotation and/or color jitters did not improve the results.

2.3.2 Training

Optimisation objective. Given a training dataset $\mathcal{D}$ consisting of pairs (x, s^*) of video clips and their ground truth transcriptions, the model is trained to maximise the log likelihoods of the transcriptions by optimising the following objective

$$\mathcal{L} = -\mathbb{E}_{(x,s^*)\in\mathcal{D}} \log p(s^*|\boldsymbol{x}) \tag{2.2}$$

Teacher forcing. To accelerate training, we follow common practice for sequence-to-sequence training with Transformers, and feed in the previous ground-truth token as the decoder input at every step, instead of using auto-regression. The tokens are fed into the decoder via a learnable embedding layer.

Training protocol. Training is performed in two stages. First, the whole network is trained end-to-end on short phrases of 2 words. Following [J. S. Chung and Andrew Zisserman 2016a; Afouras et al. 2019], we use frame word-boundaries to crop out training samples from all the possible combinations of 2 consecutive words in the dataset, which provides natural augmentation. Upon convergence, we freeze the visual backbone, then pre-extract and dump the visual features for all the samples. In the second training stage that follows, we train the encoder-decoder sub-network on all possible sub-sequences (n-grams) of length 2 or larger that can be generated by combining consecutive word utterances in the dataset.

Discussion. We note that our training protocol is much simpler than the ones commonly used in prior works [J. S. Chung and Andrew Zisserman 2016a; Afouras et al. 2019; Stavros Petridis et al. 2018b]. By using the same network and loss during the backbone pre-training stage, we obtain a good initialization of the entire network and enable a smooth transfer. This is in contrast to other works that pre-train with a different proxy loss and require a separate word classification head that is subsequently discarded. The second stage is significantly simpler to implement and requires a single run, unlike curriculums that gradually increase the length of the training sentences and usually require a complicated tuning process with multiple manual restarts to achieve the best results. We observed that our proposed second stage training setup matches the performance of the complicated curriculum strategy used in previous works, while being more efficient in terms of training time and manual efforts.

Method	Training: Datasets used	Training: # hours	Evaluation: LRS2	Evaluation: LRS3
LIBS [Y. Zhao et al. 2020]	LRS2, LRS3	698	65.3	-
Hyb. CTC [Stavros Petridis et al. 2018b]	LRS2, LRW	389	63.5	-
TDNN [Yu et al. 2020]	LRS2	224	48.9	-
Conv [X. Zhang et al. 2019]	LRS2, LRS3	698	51.7	60.1
Hyb. [Ma et al. 2021c]	LRS2, LRW	389	37.9	-
Hyb. [Ma et al. 2021c]	LRS3, LRW	639	-	43.3
Ours	LRS2, LRS3	698	**28.9**	**40.6**
TM [Afouras et al. 2019]	LRS2, LRS3, LRW, MV-LRS*	1,637	48.3	58.9
V2P [Shillingford et al. 2019]	LSVSR*	3,886	-	55.1
RNN-T [Makino et al. 2019]	YT31k*	31,000	-	33.6
Ours	LRS2, LRS3, MV-LRS*, TEDx_{ext}	2,676	**22.6**	**30.7**

Table 2.1: Comparison of different lip reading models on the test sets of the LRS2 and LRS3 datasets in terms of Word Error Rate % (WER, lower is better), along with the datasets and the aggregate number of hours used for training each model. Our model achieves state-of-the-art results, outperforming all previous baselines when trained on publicly available data (i.e. LRS2 and LRS3). If we additionally use MV-LRS and TEDx_{ext} for training, then our best model obtains results comparable with that of [Makino et al. 2019], even though we are only using an order of magnitude less data. This is indicative of the data efficiency of our proposed pipeline. *Large non-public labelled datasets: MV-LRS [Afouras et al. 2019] contains 730 hours, LSVSR [Shillingford et al. 2019] 3.9k hours, and YT31k [Makino et al. 2019] 31k hours of transcribed video. Results shown in blue have been obtained by training (partly or entirely) on non-public data.

2.4 Experiments

2.4.1 Data

LRS2 & LRS3. For training and evaluation, we use two publicly available sentence-level lip reading datasets: LRS2 [J. S. Chung and Andrew Zisserman 2016a] and LRS3 [Afouras et al. 2018b]. LRS2 contains video clips from a variety of shows from British television, such as Countryfile and Top Gear; the transcribed content sums up to approximately 224 hours in total. LRS3 has been collected from over $5,000$ TED and TEDx talks in English, available on YouTube, totalling 475 hours. Both datasets have been created using a detection and tracking pipeline that produces face-cropped clips roughly centered around the speaker's talking head. All videos are available at a 224×224 pixel resolution and 25 fps. The datasets contain a "pretrain" partition that includes extensive head tracks including word boundaries that have been produced by force-aligning subtitles to the audio. Those word alignments enable training at any granularity. The test sets contain only full sentences.

Additional dataset: TEDx_{ext}. To obtain more training data, we create a new

dataset from TEDx talks downloaded from YouTube, by using a pipeline similar to [Afouras et al. 2018b]. We collect 13,211 TEDx talks in English that are not included in LRS3. Unlike the videos used for the creation of LRS3 which contain manually annotated transcripts, the new videos only have the closed captions automatically produced by the YouTube ASR system. As these captions are only approximately aligned to the audio, we use the Montreal Force Aligner [McAuliffe et al. 2017] to obtain accurate alignments for the word boundaries needed by our training pipeline. For the rest of the processing (face detection, tracking, and cropping) we used the same pipeline as in [Afouras et al. 2018b]. The resulting training dataset contains 1,204 hours in total over 318,459 visual speech tracks, including text transcriptions with word boundary alignment. We call this new training set TEDx_{ext}. We note that since this pipeline does not require any manual transcriptions, the supervision comes for free, therefore it is easily scalable. However, the supervision is not as strong due to the noise in the training data introduced by the imperfect ASR transcriptions. But, as we will see, our model achieves a huge performance boost after training on this noisy data.

2.4.2 Implementation details

During the first training stage, we apply random visual augmentations on the input frames to reduce overfitting: the input videos are first resized to a square 160 pixels resolution, from which a central square 96-pixel crop is extracted. Random horizontal flipping and rotation (up to 10°) are also applied before inputting to the lip reading pipeline. During inference we use the central 96-pixel crop, and only apply the horizontal flipping augmentation.

For our best model, *i.e.*, VTP on $(H/4, W/4)$, we set $N_{VTP} = 6$ layers with 8 heads each for the encoder of the VTP module. For computational efficiency, VTP uses the recently proposed Linear Transformer [Katharopoulos et al. 2020] instead of the original Transformer [Vaswani et al. 2017]. We found that this change did not lead to a drop in the recognition performance, while being much more computationally efficient. Another design choice we had to make is deciding after which CNN layer the VTP should be applied. Transformer layers are computationally expensive at higher resolution feature maps (*i.e.* earlier layer activations), but can capture more detailed information. Given this

trade-off, we experiment with three different feature map resolutions, at $(h, w) = (H/4, W/4), (H/8, W/8), (H/16, W/16)$. For the latter two variants, we set the feature dimension $d = 512$. When pooling on $(h, w) = (H/4, W/4)$, we keep the compute and memory needs in check by performing two small changes: using $d = 256$ for the first 3 VTP layers, and then setting $d = 512$ but down-sampling the feature map to $(H/8, W/8)$ for the remaining 3 layers.

The encoder-decoder Transformer contains $N_{enc} = 6$ and $N_{dec} = 6$ layers, each with 8 attention heads. We use sinusoidal positional encodings [Vaswani et al. 2017] for PE and learnable positional encodings for SPE. We use the WordPiece tokenizer of the BERT model in HuggingFace[1], with a vocabulary of 30522 tokens. We also use an off-the-shelf pre-trained GPT2 language model for beam rescoring. For the beam rescoring, we set hyperparameter $\alpha = 0.7$ for LRS2 and $\alpha = 0.6$ for LRS3 respectively. We train all models with the Adam optimiser[Diederik P. Kingma and Welling 2014] with $\beta_1 = 0.9$, $\beta_2 = 0.98$ and $\epsilon = 10^{-9}$. In the first stage of the training, we follow a Noam learning rate schedule [Vaswani et al. 2017] for the first 50 epochs and then reduce the learning rate by a factor of 5 every time the validation loss plateaus, until reaching 10^{-6}. For the second stage, the learning rate is initially set to $5e^{-5}$ and reduced by a factor of 5 on plateau down to 10^{-6}. For our best reported models on public data, the first stage of training takes approximately 14 days on 4 Tesla v100s GPUs. The second stage takes 1.5 days on 1 Tesla v100 GPU.

2.4.3 State-of-the-art lipreading

We compare the results of our method to existing works in Table 2.1. It is clear that our best model outperforms all prior work trained on public data, on both the LRS2 and LRS3 benchmarks. In particular, compared to the strongest baseline of [Ma et al. 2021c], our best model performs 9% better on LRS2 and 2.7% better on LRS3. When also using MV-LRS and TEDx_{ext} for training, we obtain a significant boost, achieving 22.6% and 30.7% WER for LRS2 and LRS3 respectively. We even outperform [Makino et al. 2019] by a considerable margin while using 10× lesser training data. This clearly suggests that our pipeline is highly data-efficient.

[1]https://huggingface.co/transformers/pretrained_models.html



Figure 2.2: Visualization of the visual attention masks $\boldsymbol{a}$ from the VTP module superimposed on the input frames that produce them. The video clips used here are random samples from the LRS3 dataset. It is evident that the model follows the more discriminative mouth region.

2.4.4 Ablations

We perform various ablations to better understand the different aspects of our pipeline. For all the ablation experiments the models are trained and evaluated on public data only, *i.e.* the LRS2 and LRS3 datasets.

Importance of each module. We show the impact of each proposed module in the final scores, starting from a variation of the TM-seq2seq model[Afouras et al. 2019][2], and building up to our full model. We summarize the results of this study in Table 2.2. It is clear that all the proposed improvements give significant performance boosts and are largely orthogonal. In particular, the use of WordPiece tokens contributes a 3.8% absolute improvement on LRS2, while introducing the VTP module decreases the WER by 6.3%. Using LM to rescore the beams and applying test-time horizontal flipping leads to another 1.1% and 0.9% improvement respectively.

VTP resolution. The VTP module is capable of aggregating the spatial features at arbitrary feature map resolutions. But, we show that it is more effective when operating on finer high-resolution feature maps rather than coarser low-resolution feature maps. This is evident in Table 2.3, where pooling after $conv_{2,3}$ at a spatial resolution of 24×24 is much more effective than pooling on lower-resolution feature maps of 12×12 or 6×6.

Training protocol. Previous works [Afouras et al. 2019] follow a curriculum

[2]Using the same CNN extractor as our model for fair comparison

Method	WER	Δ
TM-seq2seq† baseline	41.0	-
+ WordPiece	37.2	−3.8
+ VTP	30.9	**−6.3**
+ Beam LM rescoring	29.8	−1.1
+ Test-time augmentation	28.9	−0.9

Table 2.2: Ablation on the design improvements proposed in this work. The results reported are for the test set of the LRS2 dataset. It is clear that all the proposed components contribute independently to the performance boost. †The baseline is an improved version of TM-seq2seq [Afouras et al. 2019].

Method	# transformer layers	WER
Without VTP	0	37.2
VTP @ $(H/16, W/16)$	2	35.7
VTP @ $(H/8, W/8)$	3	33.8
VTP @ $(H/4, W/4)$	6	30.9

Table 2.3: Ablation on the input spatial resolution for the VTP module. The number of Transformer layers for each stage is chosen such that the total number of parameters in the visual front-end is approximately the same. We see that pooling from higher resolution feature maps clearly leads to better results.

strategy during training: the sequence length is gradually increased over the course of the training. While this protocol, indeed works better, we argue that the boost in performance does not come from the curriculum learning but from something else: data augmentation. Over the course of the training process, the model gets to train on all sub-sequences (n-grams) of various lengths, and this is an effective data augmentation that reduces over-fitting. Indeed, we observe that if we simply train on all n-gram sub-sequences at once (as opposed to a slow length increase), we achieve a WER of 30.92, which is comparable to the WER of 30.91 following a curriculum protocol. Not only does this experiment shed new light on the current understanding of the training pipeline of lip reading, it also achieves similar results while following a much simpler training process that requires significantly less manual tuning.

2.4.5 Visual attention visualization

In Figure 2.2, we visualize the visual attention maps that the VTP module produces. Note that the lips region is tracked very accurately while the speakers turn their heads around, even for extreme profile views.

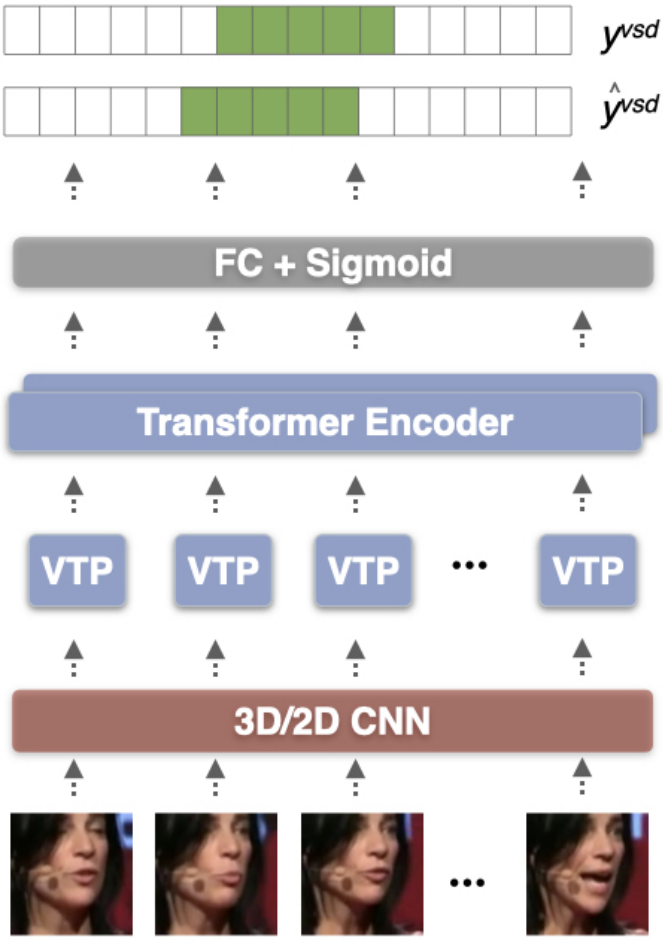


Figure 2.3: Visual Speech Detection pipeline. To predict the presence of speech at each video frame, we add a sigmoid prediction head on top of the lip reading model's visual encoder.

2.5 Visual Speech Detection Application

We build a VSD model on top of our lip reading transformer encoder by simply adding a fully connected (FC) layer and a sigmoid activation on top of the frame-level encoder outputs to classify whether the person is speaking in that frame or not:

$$\boldsymbol{y} = \sigma(\ \mathrm{FC}\ (\boldsymbol{g_{enc}})\ ) \in \mathbb{R}^T.$$

The architecture is illustrated in Figure 2.3. We train the VSD head on top of the pre-trained lip reading encoder using a binary cross-entropy loss between the predicted and ground truth labels, y_t and y_t^*:

$$\mathcal{L}^v = \frac{1}{T}\sum_{t=1}^{T} y_t^* \log y_t + (1 - y_t^*) \log(1 - y_t) \tag{2.3}$$

Dataset and evaluation. We train our VSD model on the train split of the popular AVA ActiveSpeaker dataset [J. Roth et al. 2020]. This dataset is created from movies and contains 120 videos ($2.6M$ frames) for training, 33 videos ($768K$ frames) for validation and 109 videos ($2M$ frames) in the test set. Each frame contains bounding box annotations for the face, along with a label indicating if the person is (i) speaking and audible, (ii) speaking but not audible, (iii) inaudible. The second class covers cases where the person is visually speaking in the background, but his voice/speech is not audible. Since we operate only on the visual frames, we combine the samples of the first two classes and train the model to perform binary classification. We initialize the weights from our pre-trained lip-reading models and fine-tune all the layers with the Adam optimizer by using a small learning rate of 10^{-6}. To evaluate the performance of our model and baselines we use the mean Average Precision (mAP) metric as defined by the dataset authors [J. Roth et al. 2020]; we compute the metric using the evaluation script[3] that the authors supply. We also report our scores on the held-out non-public test set with the assistance of the Ava-ActiveSpeaker challenge organizers.

Results. We show quantitative results in Table 2.4, where we report the VSD performance of our best VTP-based model (corresponding to the last row of Table 2.1), alongside results from previous works. It is evident that our model greatly outperforms the video-only baseline of [J. Roth et al. 2020], and even outperforms several of the recently proposed audio-visual methods ([Alcázar et al. 2020; J. S. Chung 2019]). Moreover, to once again showcase the benefits of the VTP module, we also compare with an ablation of our model that uses Global Average Pooling (GAP) on top of the CNN features instead of VTP, with identical train-val-test settings. VTP outperforms this model on this task as well, by 8 mAP points. We refer to the project page for qualitative video examples and examples of inference on silent videos using the whole pipeline, including VSD and lip reading.

Limitations and Ethical considerations. Narrowing the gap between lip reading and ASR performance opens up opportunities for useful applications, as stated in the introduction of the main paper: (i) improving speech recognition when the audio is corrupted in some manner; (ii) enabling silent dictation; (iii) transcribing archival silent films; (iv) helping speech-impaired individuals, e.g. people suffering from Lou Gehrig's disease speak [Shillingford et al. 2019]; and (v) enabling people

[3] https://tinyurl.com/ava-script

Method	A	V	mAP (val)	mAP (test)
Roth *et al.* [J. Roth et al. 2020]	✓	✓	79.2	82.1
Alcazar *et al.* [Alcázar et al. 2020]	✓	✓	87.1	86.7
Chung *et al.* [J. S. Chung 2019]	✓	✓	87.8	87.8
MAAS-TAN [León-Alcázar et al. 2021]	✓	✓	88.8	88.3
TalkNet [Tao et al. 2021]	✓	✓	**92.3**	**90.8**
Roth *et al.* [J. Roth et al. 2020]	✗	✓	73.5	71.1
Ours (CNN + GAP)	✗	✓	81.4	80.2
Ours (VTP)	✗	✓	**89.2**	**88.2**

Table 2.4: Visual Speech Detection performance on the validation (*val*) and test sets of the AVA ActiveSpeaker benchmark dataset. The *A* and *V* columns denote which modalities the corresponding method uses as input. Our VTP model outperforms the baseline video-only model of [J. Roth et al. 2020] by a large margin (over 17 mAP improvement). In fact, we even outperform several of the recently proposed audio-visual methods ([Alcázar et al. 2020; J. S. Chung 2019]), obtaining results close to the current state-of-the-art without using any audio.

with aphonia (loss of voice) to communicate just by using lip movements.

But, it also raises privacy issues and the risk of potential malign uses. One issue that is often raised is the potential for malign surveillance, e.g. using CCTV footage from public spaces to eavesdrop on private civilian conversations. However, this is in fact very low risk due to a number of factors: we achieve a low WER on benchmarks containing video material that is professionally produced and at high resolutions and frame-rates, and under good lighting conditions. Moreover the speakers are aware of being filmed and collaborate, most of the time speaking while frontally facing the camera. In contrast, CCTV usually operate at much lower resolution and frame rates and from unusual angles. As shown in prior work [J. S. Chung and Andrew Zisserman 2017; Shillingford et al. 2019; Makino et al. 2019], lip reading performance greatly deteriorates with lower frame rate or input resolutions, or when non-frontal (e.g. profile or overhead viewpoints) rather than frontal speaker views are considered.

We will be making the code and pre-trained models of this work public. This technology is already available to a small handful of corporations that have access to enough data and compute resources for training. We believe that open access is important in order to accelerate progress in the field, as well as to enable research on defences against potential adversarial attacks [Ma et al. 2021b; Gupta et al. 2021; Jethanandani and D. Tang 2020]. Finally, releasing our code and models

leads to democratisation of research – making strong models publicly available rather than to only a handful of companies can accelerate research while also making the entire process more transparent.

Overall, we believe that the benefits of the positive applications of lip reading that we have discussed (e.g. medical) greatly outweigh the risk of malevolent uses, the latter ones being inflated, therefore transparent research into this field should be continued and encouraged by the community.

2.6 Conclusion

We have presented an improved architecture for lip reading based on attention-based aggregation of visual representations as well as several enhancements to the training protocol, including the use of sub-word tokenisation. Our best models achieve state-of-the-art results, outperforming prior work trained on public data by a significant margin, and even industrial models trained on orders of magnitude more data. We have also designed a Visual Speech Detection model on top of our lip reading system that obtains state-of-the-art results on this task and even outperforms several audio-visual baselines.

Acknowledgements. Funding for this research is provided by the EPSRC Programme Grant VisualAI EP/T028572/1, and a DeepMind Graduate Scholarship.

Chapter 3

Speech Recognition Models are Strong Lip-readers

The paper has been accepted for publication at INTERSPEECH, 2024.

Speech Recognition Models are Strong Lip-readers

K R Prajwal Triantafyllos Afouras Andrew Zisserman

Visual Geometry Group, University of Oxford

{prajwal,afourast,az}@robots.ox.ac.uk

July 8, 2025

Abstract

In this work, we show that a large pre-trained ASR model can be adapted to perform lip-reading. Our method enables an ASR model like Whisper to interpret lip movements in a video and output text transcriptions. We achieve this by learning a cross-modal mapping from a lip sequence to a speech sequence, allowing a pre-trained ASR model to directly perform lip-reading. The mapping can be learnt simply by backpropagating the cross-entropy loss on the text labels through the pre-trained, frozen ASR model. We achieve an impressive gain of 5.7 WER in the low data regime on the LRS3 benchmark over previous lip-reading methods. Finally, we demonstrate that the same strategy can be extended to other visual speech tasks, such as identifying the spoken language in silent videos.

3.1 Introduction

The advent of foundational models has brought to light previously unexplored dimensions of deep learning's ability to generalize. Large models pre-trained on massive datasets achieve state-of-the-art performance on a wide range of downstream tasks. Even more remarkably, these models also demonstrate strong cross-modal learning capabilities [Tsimpoukelli et al. 2021; Alayrac et al. 2022; Junnan Li et al. 2023].

In this paper, we study the cross-modal generalization capabilities of a pre-trained automatic speech recognition (ASR) model like Whisper [Radford et al.

2022] for the highly challenging task of visual speech recognition (lip-reading). This allows us to leverage the speech modeling capabilities of state-of-the-art ASR models. ASR models can be trained on massive audio datasets and learn to form compact latent speech representations that are then decoded into text. Therefore, in order to learn a strong lip-reading model, we can rely on a pre-trained ASR model for the speech and language modeling and need only focus on learning two things specific to the visual modality: a good visual representation, and a module for mapping the visual features into a speech-like representation. This approach paves the road to much more sample-efficient learning for lip-reading, which is important because publicly available transcribed video datasets are orders of magnitude smaller than their audio counterparts. For instance, among the public datasets, VoxPopuli [Changhan Wang et al. 2021] consists of 400K hours of multilingual speech data, whereas LRS3 [Afouras et al. 2018b], the most commonly used public lip-reading benchmark contains only about 400 hours. In addition to sample-efficiency, using a pre-trained ASR model provides an additional benefit of saving computational resources for training, as a large part of the model is reused.

The essence of our approach to lip-reading is to replace the audio input sequence of a pre-trained ASR model with the visual input sequence of a talking face, while maintaining the speech transcription abilities of the pre-trained ASR model. We study the recently released Whisper ASR model [Radford et al. 2022], which is based on a vanilla Transformer encoder-decoder architecture using audio spectrograms as input, and has been trained on 680k hours of weakly-labelled audio speech data.

Our contributions are: (1) re-purposing a large-scale, frozen, pre-trained ASR model to perform lip-reading; (2) achieving state-of-the-art results on the standard lip-reading benchmarks, and (3) demonstrating that our approach can be successfully extended to the visual language identification task, obtaining a significant improvement over previous methods. Please follow the website for demo samples and code release: https://www.robots.ox.ac.uk/~vgg/research/whisperer

3.2 Related Work

3.2.1 Lip reading

The advent of deep learning, along with the creation of larger datasets such as LRW [J. S. Chung and Andrew Zisserman 2016a], LRS2 [J. S. Chung et al. 2017] and LRS3 [Afouras et al. 2018b] has led to rapid development of strong machine lip reading models in the past few years. Sentence-level lip reading methods followed two main paradigms, borrowed from the machine translation and ASR literature, namely sequence-to-sequence [Afouras et al. 2019] and CTC [Assael et al. 2016; Shillingford et al. 2019], as well as hybrid approaches [S. Petridis and Pantic 2016]. The sequence learning architecture has also evolved from LSTMs to Transformers [Vaswani et al. 2017], to recent works using Conformers [Ma et al. 2021c; Chang et al. 2024; Ma et al. 2022] that can model both the local information and long-term temporal information.

Self-supervised learning for lip reading. Following the trend in speech recognition [Schneider et al. 2019; Baevski et al. 2020; Hsu et al. 2021; Baevski et al. 2022], there have been many works [Hsu and B. Shi 2022; Lian et al. 2023; B. Shi et al. 2022c] that learn audio-visual speech representations from unlabeled videos. These models are data-efficient – they can be fine-tuned with far fewer video-text pairs to achieve a much better performance. The most prominent one among them is AV-HuBERT [B. Shi et al. 2022c] which learns representations by predicting masked cluster assignments. In this work, we adapt these AV-HuBERT features to condition a pre-trained ASR model like Whisper.

Utilizing ASR for lip-reading. Over the years, efforts have been made to leverage ASR models to enable lip-reading training. For example [Afouras et al. 2020a] obtain noisy text labels for training by running an ASR on the accompanying audio. A similar data-efficient and scalable approach was shown by Auto-AVSR [Ma et al. 2023], which obtained impressive performance gains by pre-training on large amounts of videos labeled with noisy transcripts from a pre-trained ASR. Although our method also relies on a pre-trained ASR model, we follow a different approach: we do not merely use ASR as a labeling tool – the ASR is the key component of the lip-reading model itself. This idea is by no means a new one, previous methods [Djilali et al. 2023; Laux et al. 2023] have shown that ASR models can be

adapted for lip-reading. However, they achieve a far worse performance and employ additional pre-training strategies and loss functions to match the audio and video features. Our approach is straightforward and much faster to train – we directly train for lip-reading using limited video-text pairs and achieve a superior performance as shown in Table 3.1. Our approach is easily extensible to solve additional tasks (Section 3.6).

3.2.2 Visual language Identification

Language identification is a well-studied problem [Richardson et al. 2015; Gonzalez-Dominguez et al. 2014] as it is the first step for multi-lingual ASR. With the recent efforts being made towards multi-lingual lip-reading [Ma et al. 2022; Anwar et al. 2023], identifying the spoken language from lips becomes an important task. Few modern works exist in this space with more recent ones [Afouras et al. 2020b] aggregating lip-reading representations with shallow temporal processing networks, followed by simple classifier heads.

3.3 Method

In this section, we describe our approach. Our goal is to show that we can adapt a pre-trained ASR model to lip-read, by mapping visual features representing lip movements into the same space as the ASR model's audio features. In this paper, we realize this goal using the Whisper ASR model [Radford et al. 2022]. An overview of the approach is given in Fig. 3.1. It consists of three major parts: (i) a visual feature encoder, (ii) a cross-modal mapping network, and (iii) a pre-trained ASR network, which is the Whisper model in our case. We explain each of the three parts below.

3.3.1 Visual backbone

Our lip reading network's input is a silent video clip of T frames, $\boldsymbol{x} \in \mathbb{R}^{T \times H \times W \times 3}$. We use a pre-trained AV-HuBERT-Large model [B. Shi et al. 2022c] to extract the visual features for this input. It consists of two modules: (i) A ResNet18-like CNN, and a (ii) Transformer encoder, both of which are trained in a self-supervised manner through masked cluster prediction. The video input sequence is divided

into sub-clips of 5 frames (i.e. 0.2s) with a unit frame stride. The sub-clips are then processed by a ResNet18-like CNN architecture, where only the first CNN layer is spatio-temporal. The CNN outputs T feature vectors, one per input frame: $\boldsymbol{f} \in \mathbb{R}^{T \times C}$, where $C = 1024$ in our implementation. The feature sequence $\boldsymbol{f}$ is processed by a series of Transformer encoder layers after a linear projection to give the visual features $\boldsymbol{g_{enc}} \in \mathbb{R}^{T \times C}$. Note that we obtain our visual features without any text supervision.

3.3.2 Cross-modal mapping network

Given the visual features $\boldsymbol{g_{enc}}$, we would like to now map them to the space of the ASR model's audio features. We start by re-sampling the video features to match the number of time-steps in the audio stream. In the case of Whisper, the audio input is at twice the frame rate (50 FPS) of the video input stream, so visual inputs are simply repeated along the time dimension in order to match the audio time-steps. We denote the time-steps as T instead of $2T$ for simplicity. The re-sampled visual features ($\boldsymbol{g_{enc}}$) are passed through a cross-modal mapping network $\mathbb{M}$, which consists of an an MLP and a stack of N Transformer encoder layers to map these features to Whisper's audio space. The output of the mapping network $\boldsymbol{h_{enc}}$ can be directly fed into the pre-trained ASR model.

$$\boldsymbol{h_{enc}} = \mathbb{M}(\boldsymbol{g_{enc}} + PE_{1:T}) \in \mathbb{R}^{T \times C}.$$

where PE is the temporal position embedding.

3.3.3 Conditioning the ASR with visual inputs

The next step is to condition the ASR with the mapped features $\boldsymbol{h_{enc}}$. We briefly describe the ASR we use below.

Whisper architecture. Whisper inputs audio mel-spectrogram features and embeds these using a stack of convolutional layers to get the initial audio feature vectors $\boldsymbol{A} \in \mathbb{R}^{T \times D}$. These are then processed by a Transformer encoder network to get $\boldsymbol{A_{enc}} = \mathbb{W}_{enc}(\boldsymbol{A}) \in \mathbb{R}^{T \times D}$. A Transformer decoder is used to auto-regressively decode the text tokens s while cross-attending to $\boldsymbol{A_{enc}}$. We use Whisper-medium in all our experiments and keep the Whisper model weights frozen.

Feeding the mapped visual features to Whisper. There are several possi-

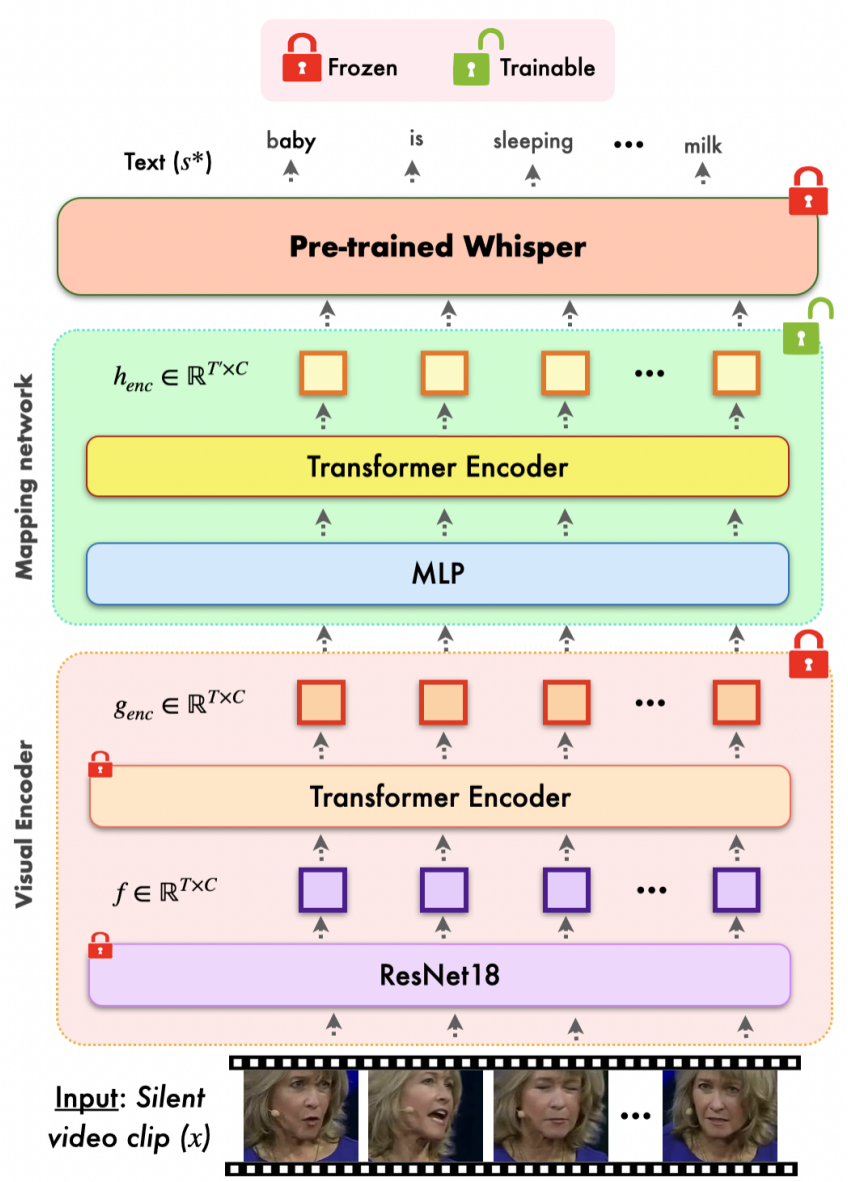


Figure 3.1: **Proposed lip reading network**. Given a silent video clip, we first encode it using a pre-trained AV-HuBERT encoder consisting of a ResNet18 followed by a Transformer encoder. The obtained visual features are temporally up-sampled (not shown in the figure for simplicity) and passed through a cross-modal mapping network to obtain inputs that can be fed into a pre-trained Whisper model. The frozen Whisper model outputs text transcriptions. Only the mapping network (specified by open lock) is trained.

bilities for feeding in the mapped visual features $\boldsymbol{h_{enc}}$ to Whisper. We found that feeding $\boldsymbol{h_{enc}}$ in place of $\boldsymbol{A}$, i.e., directly to the start of the Transformer encoder of Whisper ($\mathbb{W}_{enc}$) gives the best results. Thus, the conditioning of Whisper on the visual features $\boldsymbol{h_{enc}}$ to transcribe the text sequence s can be expressed as follows:

$$\log p(s|\boldsymbol{h_{enc}}) = \sum_{t=1}^{T_{dec}} \log p(s_t|\ \mathbb{W}_{enc}(\boldsymbol{h_{enc}}), s_{1:t-1}) \tag{3.1}$$

The output sequence s is represented in Byte-Pair-Encoding (BPE) tokens, exactly the same as the ones used by the Whisper model [Radford et al. 2022]. At inference, we perform a left-to-right beam search of width $B = 40$. We do not perform any test-time augmentations and do not use any external language models.

3.3.4 Training objective

We maximise the log likelihoods of the text transcriptions by optimising the following objective:

$$L = -\mathbb{E}_{(x,s^*)\in\mathcal{D}} \log p(s^*|\boldsymbol{x}) \tag{3.2}$$

3.4 Experiments

3.4.1 Data

The visual feature extractor from AV-HuBERT is pre-trained on the English split of VoxCeleb2 [J. S. Chung et al. 2018] and LRS3 [Afouras et al. 2018b], a total of 1759 hours of unlabeled video data. Our model is trained and evaluated on the largest publicly available sentence-level lip-reading dataset, LRS3 [Afouras et al. 2018b]. LRS3 contains clips from over $5,000$ TED and TEDx talks in English, available on YouTube, totalling 475 hours. The manual text transcriptions were automatically aligned to the audio using force-alignment to yield word boundaries; these word alignments enable training at any granularity. The dataset is split into pretrain (403h), trainval (30h) and test sets. For the low-resource setting, we use only the LRS3-trainval set of 30h, as also done in [B. Shi et al. 2022c]. The models are evaluated on the LRS3 test set containing 1,321 samples. Both LRS3 and

VoxCeleb have been created using a detection and tracking pipeline that produces clips loosely cropped around the speaker's talking head. All videos are available at 25 fps, accompanied with audio at 16 kHz. We follow AV-HuBERT's official code to extract tight mouth ROIs from the face tracks. The transcripts are tokenized with Byte-Pair Encoding (BPE) in the exact same manner as done in Whisper's official code.

3.4.2 Implementation details

We now provide the implementation details of our best performing model. We use AV-HuBERT Large as our visual feature extractor. The pre-trained model weights are obtained by optimizing for the self-supervised clustering objective on 1759 hrs of unlabeled video data from LRS3 [Afouras et al. 2018b] and the English split of VoxCeleb2 [J. S. Chung et al. 2018]. The feature encoder consists of a ResNet18 followed by 24 transformer encoder layers with a hidden dimension of 1024 and 16 heads. Our mapping network $\mathbb{M}$ consists of (i) an MLP with two FC layers with ReLU activation and Layer Normalization, and (ii) a $N = 3$ layer Transformer encoder. The feature dimensions and the number of heads match that of the Whisper's encoder, which is $C = 1024, heads = 16$. For the Whisper model, we use Whisper-medium-English, whose weights are also publicly available. We use the Adam optimizer [Diederik P Kingma and Ba 2015] with an initial learning rate of $5e^{-5}$. The learning rate is reduced by 5 every time the WER on the LRS3 validation set does not improve for 2 consecutive epochs, with a minimum learning rate of $1e^{-6}$. We finetune the AV-HuBERT encoder at this minimum learning rate for two epochs and choose the checkpoint with the best validation word error rate. We clip all gradients to have a maximum L2 norm of 0.1. The training takes only a few hours on 4 Tesla V100 GPUs.

3.5 Results

3.5.1 Quantitative scores

In Table 3.1, we compare our method with the previous works on the LRS3 test set. We categorize the previous works in Table 3.1 into three groups. The first two groups use only publicly available text transcripts for training and evaluation

Method	Unlabeled (hrs)	Labeled (hrs)	WER (%)
Fully Supervised models with publicly available data			
ASR distillation [Afouras et al. 2020a]	-	590	68.8
Conv-Seq2Seq [X. Zhang et al. 2019]	-	855	60.1
Discriminative AVSR [B. Xu et al. 2020]	-	590	57.8
Hyb. + Conformer [Ma et al. 2021c]	-	590	43.3
VTP [Prajwal et al. 2022a]	-	698	40.6
Self-supervised pre-training + Supervised finetuning			
ASR distillation [Afouras et al. 2020a]	334	590	59.8
LiRA [Ma et al. 2021a]	433	30	71.9
	1,759	433	49.6
LiteVSR [Laux et al. 2023]	639	59	45.7
AV-HuBERT Large [B. Shi et al. 2022c]	1,759	30	32.5
		433	28.6
SynthVSR [X. Liu et al. 2023]	3,652	438	27.9
Auto-AVSR [Ma et al. 2023]	1,759	433	25.0
Lip2Vec [Djilali et al. 2023]	1,759	30	31.2
		433	26.0
Ours	1,759	30	**25.5**
		433	**24.3**
Trained on large-scale non-publicly available datasets			
Deep-AVSR [Afouras et al. 2019]	-	1,519	58.9
Large-scale AVSR [Shillingford et al. 2019]	-	3,886	55.1
RNN-T [Makino et al. 2019]	-	31,000	33.6
VTP [Prajwal et al. 2022a]	-	2,676	30.7
ViT-3D [Serdyuk et al. 2022]	-	90,000	17.0
Synth-VSR [X. Liu et al. 2023]	3,652	3,068	16.9
LP [Chang et al. 2024]	-	1,00,000	12.5

Table 3.1: WER (%) of our models and the comparison with prior works on the LRS3 test set. Among the different approaches trained on similar number of hours of labeled data, our method gives clear improvements.

such as LRS2 [J. S. Chung et al. 2017], LRS3 [Afouras et al. 2018b]. The first group consists of models that are trained only with text supervision. The second group involves a mix of self-supervised representation learning and/or ASR-aided training, followed by finetuning with manually annotated clean transcripts. The third and final group are methods that use large, proprietary datasets for training. Our method falls into the second category. We consider two settings, depending on the amount of labeled data used for training. In the 30hrs setting, we achieve a WER of **25.5**, which is **5.7 WER** points better than the next best model (31.2) at this data scale. Our best model that uses 433hrs of labeled data achieves a WER of **24.3**, surpassing all other approaches that train using the same datasets. Thus, our approach is particularly useful when there is limited video-text paired training data.

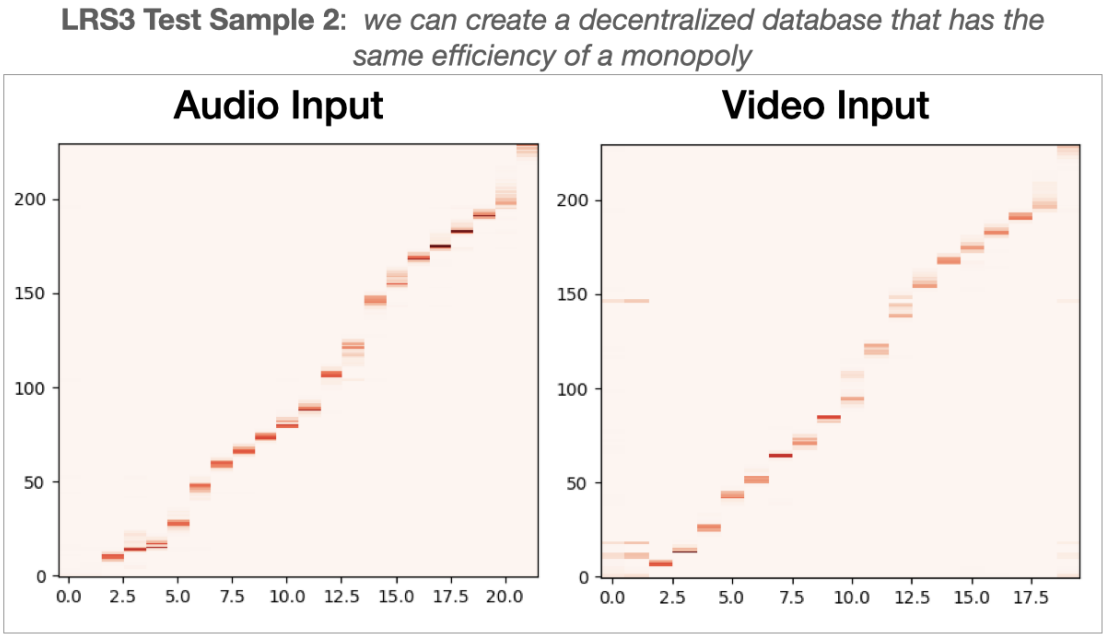


Figure 3.2: **Whisper's cross attention map** for a specific decoder layer's attention head. X-axis denotes the text tokens. Y-axis denotes the input source time-steps which can be either audio (ASR) or video (lip reading). The cross-attention plot are very similar across modalities, indicating that the decoder is not distinguishing between the modalities.

3.5.2 Whisper decoder's cross-attention heatmaps

We now investigate how the pre-trained Whisper model behaves when we feed in (i) audio input to perform ASR, (ii) the mapped visual features (h_{enc}) to perform lip-reading. We specifically visualize the cross-attention heatmaps between the text tokens and the input source (audio or video). Figure 3.2, plots the cross-attention heatmaps for a sample from the LRS3 test set for audio and video inputs. It can be seen that, except for a few minor differences, the attention plots are strikingly similar, indicating that the Whisper's text decoder is able to interpret the input from different modalities in a similar manner.

3.5.3 Ablation studies

In the arXiv version of the paper, we provide ablations of our approach, analysing the various design choices such as: (i) the right place to feed the visual features to Whisper, (ii) which mapping network is the best, (iii) how the size of the Whisper model affects the lip-reading performance, (iv) does fine-tuning the ASR model help? We also analyze the performance of our model against various sequence lengths.

3.6 Extending to Visual Language Identification

We demonstrate that our approach can be easily extended to another task that Whisper is proficient at: language identification. Given a speech input, Whisper has been trained to identify the language of the speech in addition to transcribing what is being spoken. Similar to our re-purposing of Whisper for lip-reading, we show that we can identify the language spoken purely from the lip movements, i.e. visual language identification [Afouras et al. 2020b].

Dataset. Due to the lack of public benchmarks for this task, we show results by creating train-val-test splits on the LRS3-lang+ dataset [Afouras et al. 2020b]. This dataset contains 14 languages sourced from 19k TEDx talks on YouTube. We split the dataset into 85-5-10 for train-val-test splits. We ensure that the same TEDx talk is not repeated across the splits to ensure there is no overlap of content or speakers. Details on the distribution of the languages in the dataset is provided in the original paper [Afouras et al. 2020b] and also in the arXiv version of the paper.

Approach. We follow the same approach as we did for lip-reading, with only a few minor changes. We now switch from a monolingual (English) Whisper model to the multi-lingual variant. For example, when given a French speech, the multilingual Whisper model would output a transcripts like <|fr|><|transcribe|> *Chacun voit midi à sa porte.* <|endoftranscript|>. For this task, we would like to output only the first token, which gives us the language code. Thus, the task of language identification is cast as a sequence-to-sequence task, where the input can be speech (visual or audio) and output is the language code.

Implementation details and Evaluation. We follow the same hyper-parameter settings that we use for the lip-reading model. We train on 1–5 second clips. At test-time, we evaluate on clips of lengths up to 5 seconds. We use the average class accuracy (chance: 7%) to measure the model's performance.

The current best model [Afouras et al. 2020b] evaluates on 5s clips and reports a mean class accuracy of **67.2%**. Unfortunately, the authors did not release the train-test splits to directly compare with this score. On our test split, which can contain clips of less than 5s (which makes it harder due to shorter segments), we achieve a mean class accuracy of **78.7%**, which is already far better than the previously reported performance. We also show the confusion matrix across the

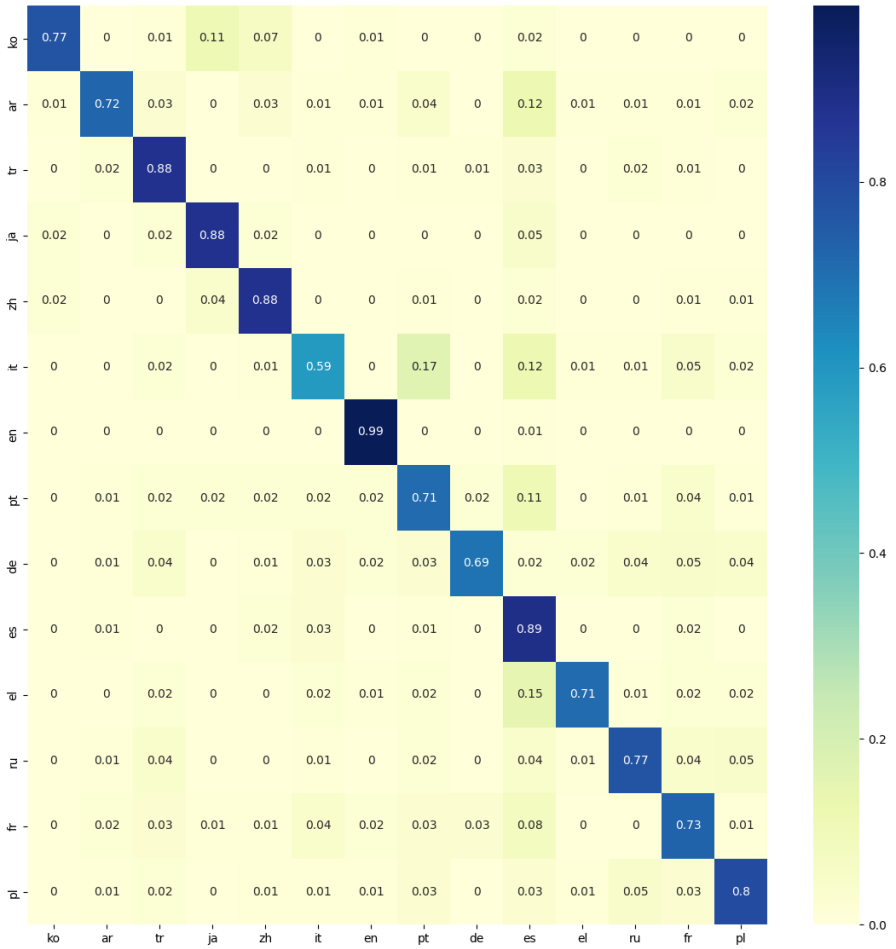


Figure 3.3: **Row-normalized Confusion matrix** of the language identification predictions on the test set. The languages are as follows: ('ko' : Korean), ('ar' : Arabic, ('tr' : Turkish), ('ja' : Japanese), ('zh' : Mandarin), ('it' : Italian), ('en' : English), ('pt' : Portugese), ('de' : German), ('es' : Spanish), ('el' : Greek), ('ru' : Russian), ('fr' : French), and ('pl' : Polish).

different languages in Fig 3.3. English and Spanish are by far the most accurate of the languages, presumably because there are more training data for these two languages. Italian seems to be a very difficult language, often being confused with Portuguese, Spanish or French. The visual language identification results serve as a strong evidence that our approach can be extended to solve several other lip-related tasks that Whisper performs with audio inputs.

3.7 Conclusion and extensions

We present an approach that employs an ASR model to achieve state-of-the-art performance in lip-reading. Our method is also well-positioned to make use of the rapid ongoing progress in large-scale speech recognition models [B. Shi et al. 2022c; Haliassos et al. 2022; Lian et al. 2023]. We believe that this work is only

the tip of an iceberg – it is yet to be explored if we can adapt the visual features to the full range of Whisper's capabilities other than English speech recognition and language identification: (i) multi-lingual lip-reading, (ii) visual speech detection (detecting the speech time-steps), (iii) visual speech-to-text translation (e.g. English lip movements to German text), and (iv) conditioning on a larger context. We believe that our approach can have an impact across all these tasks, especially when there is a scarcity of labeled data.

Acknowledgements. Funding for this research is provided by the EPSRC Programme Grant VisualAI EP/T028572/1, and a DeepMind Graduate Scholarship.

Chapter 4

Scaling Multilingual Visual Speech Recognition

The paper has been accepted for publication at ICASSP, 2025.

Scaling Multilingual Visual Speech Recognition

K R Prajwal* Sindhu Hegde* Andrew Zisserman

Visual Geometry Group, University of Oxford

{prajwal,sindhu,az}@robots.ox.ac.uk

July 8, 2025

Abstract

Visual Speech Recognition (lip-reading) has witnessed tremendous improvements, reaching word error rates as low as 12.8 WER in English. However, the performance in other languages is lagging far behind, due to the lack of labeled multilingual video data. In this work, we reduce the performance gap with the help of three key advances: (i) the largest multilingual lip-reading dataset to date, (ii) a single multi-task architecture that can perform two tasks simultaneously: identify the language and transcribe the

*Equal contribution.

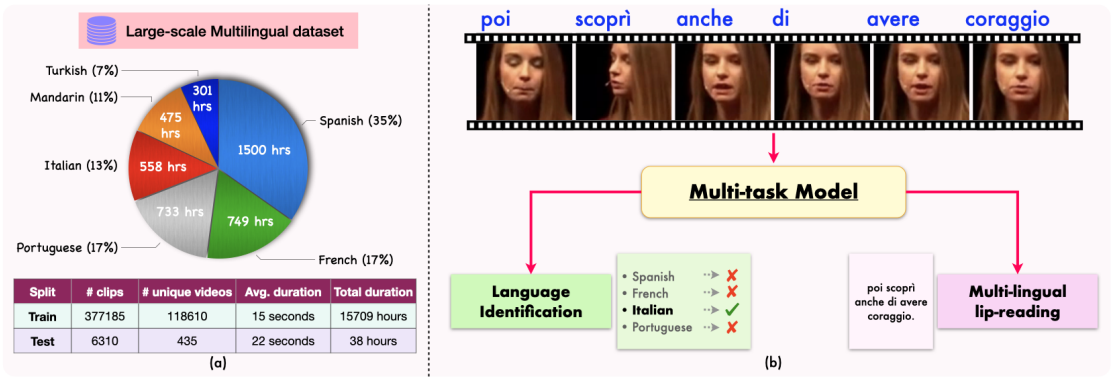


Figure 4.1: (a) Our new *MultiVSR* dataset: We curate the largest multilingual lipreading corpus to date, consisting of $\approx 15,700$ hours of video paired with word-aligned transcripts (11430 hours English + 4316 hours multilingual). (b) We develop a unified multi-task framework capable of performing both lip-reading and visual language identification tasks. The model, trained on our newly collected dataset, achieves state-of-the-art results in both tasks.

utterance, and (iii) jointly training this architecture on all the languages together, resulting in large WER improvements as opposed to training monolingual models separately. We achieve state-of-the-art performance across both visual language identification and multilingual lip-reading tasks. Moreover, our pipeline uses zero manual annotations, as all the training transcriptions are obtained using a pre-trained ASR model. We also show that our multilingual model can be readily fine-tuned for new low-resource languages to achieve significant improvements when compared to training from scratch. Our data, code, and models will be released for future research.

4.1 Introduction

Automatic Speech Recognition (ASR) models have taken giant strides in recent years, achieving human-like accuracy [Y. He et al. 2019; Jinyu Li et al. 2022; Prabhavalkar et al. 2023; Manh Tien Anh and Ho Sy 2024; Hono et al. 2024] across a large number of languages [Tjandra et al. 2023] while also being fast and real-time. Visual speech recognition (VSR) a.k.a lip-reading is a far more challenging task compared to ASR, because lip movements are inherently ambiguous to transcribe from. In spite of this, the word error rates have plummeted from 50-60 [J. S. Chung et al. 2017; Afouras et al. 2019] to just 12.8 [Chang et al. 2024] on the LRS3 [Afouras et al. 2018b] test set in a matter of five years. However, even today, in stark contrast to ASR models, lip-reading models have been fairly accurate only for English. One possible reason for this could be the lack of resources for non-English data. The state-of-the-art multilingual lip-reading model today [Yeo et al. 2024] still only uses 331 hours of Spanish data, which is 25% smaller than that used for English even in 2018 [Afouras et al. 2018a], and 95% smaller than what is used for English lip-reading today [Ma et al. 2023].

In this work, we scale multilingual lip-reading with the help of three key advances. We start by collecting *MultiVSR*, a large-scale multilingual dataset from TEDx YouTube videos. With over 15,700 hours of video data across 7 languages, this is the largest multilingual lip-reading dataset, as far as we know. Our data collection process requires little to no manual effort, because the transcriptions are automatically obtained from a pre-trained ASR model like Whisper [Radford et al. 2022; Yeo et al. 2024]. Using this data, we train a single model that can

perform two tasks in a single framework: identifying language from lip movements and transcribing what was uttered. Finally, we train this multi-task model jointly across all languages. This is in contrast to previous works [Ma et al. 2022; Yeo et al. 2024] where each language has a separate model. We clearly show that the joint multilingual training leads to large gains in each individual language. We also demonstrate that our pre-trained multilingual model can be fine-tuned for a new language with limited data, achieving word error rates that are 15-20 points lower than training on these languages from scratch.

In summary, our contributions are: (i) A new large-scale multilingual dataset, *MultiVSR*, comprising $\approx 15,700$ hours of video data, (ii) A multi-task model capable of both language identification and transcription from silent lip movements, and (iii) A unified multilingual lip-reading model that is far superior to previous monolingual models, both on our *MultiVSR* dataset and multilingual TEDx (mTEDx) [Salesky et al. 2021] benchmark.

4.2 Related Work

English Lip-reading: Lip-reading has gained significant attention in recent years, with advancements in both the scale of training data and model architectures leading to substantial performance improvements. Early works primarily focused on English-speaking datasets, such as LRW [J. S. Chung and Andrew Zisserman 2016a], LRS2 [J. S. Chung et al. 2017], and LRS3 [Afouras et al. 2018b], which laid the foundation for visual speech recognition. With the availability of vast amounts of video data, numerous works [Afouras et al. 2018a; Stavros Petridis et al. 2018a; Afouras et al. 2019; X. Zhang et al. 2019; Y. Zhao et al. 2020; Afouras et al. 2020a; Ma et al. 2021c; Prajwal et al. 2022a; Yeo et al. 2023; Ma et al. 2023; Prajwal et al. 2024; B. Shi et al. 2022c; Djilali et al. 2023; X. Liu et al. 2023; Laux et al. 2023; Haliassos et al. 2022] have emerged, resulting in a drop of over 40 WER points in English lip-reading over the last few years.

Non-English Lip-reading: Non-English Lip-reading remains under-explored primarily due to the lack of large-scale multilingual datasets. Existing multilingual video datasets such as VoxCeleb2 [J. S. Chung et al. 2018] and AVSpeech [Ephrat et al. 2018] are not accompanied by transcripts, and the individual language subsets are relatively small in scale. The multilingual TEDx (mTEDx) [Salesky et al.

2021] dataset provides speech transcriptions in several languages, but it lacks video data and even if we take the corresponding (unofficial) video clips from YouTube, it is too small for training accurate lip-reading models. MuAViC [Anwar et al. 2023] extends mTEDx by including videos, but has been mainly used for audio-visual ASR and translation. Recently, a few works [Ma et al. 2022; M. Kim et al. 2023b; Yeo et al. 2024] have focused on multilingual lip-reading, proposing various methods to either combine or extend existing datasets to cover multiple languages. The largest multilingual lip-reading dataset to date was curated in [Yeo et al. 2024], where the authors merged data from mTEDx, VoxCeleb2, and AVSpeech to create a dataset spanning four languages: Spanish (384 hours), French (331 hours), Portuguese (420 hours), and Italian (152 hours). The spike in data size led to a drop in word error rates. But, it is worth noting that the LRS3 dataset [Afouras et al. 2018b] used for training English lip-reading back in 2018 [Afouras et al. 2018a] was bigger than each of the above language splits. Moreover, most of the recent works [Ma et al. 2022; M. Kim et al. 2023b; Yeo et al. 2024] have only trained monolingual lip-reading models, which we show is sub-optimal.

Visual Language Identification: A critical pre-processing step for multilingual lip-reading is to identify what language is being uttered. Only a handful of works [Newman and Cox 2012; Prajwal et al. 2024; Afouras et al. 2020b] have tackled this problem, and none of them perform multilingual transcription. We outperform the previous best model [Prajwal et al. 2024], while also doing this jointly with multilingual lip-reading.

4.3 A New Large-scale Multilingual Dataset

We present a new large-scale multilingual lip-reading dataset that includes video clips paired with aligned text transcripts. The dataset covers English along with six other languages: Spanish, French, Italian, Portuguese, Mandarin, and Turkish. Each clip features a visible speaker who is the sole person speaking in the video. In total, the dataset encompasses approximately $15,700$ hours of video segments, representing a diverse range of individuals, languages, and facial poses. Figure 4.1 (left) showcases a summary of the data statistics and Table 4.1 gives the detailed dataset information.

Table 4.1: **Data Statistics:** The *MultiVSR* dataset comprises video data from seven languages, totaling over 15,700 hours. This makes it the largest multilingual lip-reading data available to date.

Language	**split**	**# clips**	**# unique videos**	**avg. duration per clip (secs)**	**duration (hours)**
English	Train	2,681,218	86,942	15.35	11,429.73
Spanish	Train	399,026	11,540	13.44	1,490.08
	Test	1598	141	22.04	9.78
French	Train	223,796	5,307	11.92	741.28
	Test	1536	82	18.29	7.80
Portuguese	Train	171,827	5,407	15.20	725.70
	Test	1152	79	21.75	6.96
Italian	Train	129,522	4,225	15.36	552.72
	Test	963	62	19.81	5.30
Mandarin	Train	101,221	3,170	16.75	470.82
	Test	666	43	24.80	4.59
Turkish	Train	64,915	2,019	16.54	298.18
	Test	395	28	28.89	3.17
Combined	**Train**	**3,771,825**	**118,610**	**14.99**	**15,708.52**
	Test	**6310**	**435**	**21.45**	**37.60**

4.3.1 Data collection and pre-processing

The dataset is automatically gathered from publicly available TEDx videos on YouTube. We curated approximately 17,000 playlists and downloaded around 155,000 videos. TEDx is an ideal data source for our purposes because most of the videos feature a single visible speaker who is actively speaking, and the videos are of high quality.

Pre-processing: Our pre-processing pipeline (Fig. 4.2) involves two stages: (i) person tracking and cropping, and (ii) synchronization detection. We begin by resampling all videos to 25 fps and then detecting faces in each video frame using the S3FD face detector [S. Zhang et al. 2017]. We then track segments where the face is clearly visible and speaking (see Figure 4.2 (b)). Clips shorter than 1 second are discarded, and we also filter out faces with a resolution below 96×96 pixels. Next, in the synchronization stage, we verify whether the crop is synchronized with the corresponding audio segment. Specifically, we use SyncNet [J. S. Chung and Andrew Zisserman 2016b] to calculate synchronization offsets and retain only the crops where the offset is within ± 10 frames by sync correction (see Figure 4.2



Figure 4.2: ***MultiVSR* Dataset:** (a) We begin by collecting a large set of 155,000 TEDx videos across multiple languages. (b) These videos are pre-processed by detecting and tracking person crops. (c) In the final step, we utilize SyncNet [J. S. Chung and Andrew Zisserman 2016b] to discard clips that do not contain the face of the speaker and the clips with synchronization offsets of $\geq \pm 10$ frames.

(c)). The clips containing a different person other than the speaker, and clips where the synchronization offset is above ± 10 frames are discarded. This step is critical since we rely on a pre-trained automatic speech recognition (ASR) model to generate transcripts from the audio segments (described below), rather than manual labeling. This process results in approximately 16,000,000 video clips, comprising a total of 15,709 hours of footage for our dataset.

4.3.2 Automatic labelling

As demonstrated in [Yeo et al. 2024], automatic annotations are scaleable and produce results comparable to manual labeling for lip-reading; we thus opt for automatic label generation for our dataset. We identify the language and generate the word-level time-aligned transcripts for the video segments using the pre-trained Whisper Large-v3 ASR model in WhisperX [Bain et al. 2023].

4.3.3 Data Splits

We split the data into train and test, with no speaker common across the splits. This protocol is in line with other standard lip-reading datasets like LRS2 [J. S. Chung et al. 2017] and LRS3 [Afouras et al. 2018b]. We also exclude the speakers present in the test sets of LRS3 [Afouras et al. 2018b] and mTEDx [Salesky et al. 2021], ensuring that models trained on our data can be benchmarked on all these datasets. We also manually verify our test set and correct transcription errors.

4.4 Multi-Task Network

We design a multi-task encoder-decoder model as illustrated in Figure 4.3. Given a silent video sequence with N_f frames, $V = \{V_1, V_2, \ldots, V_{N_f}\}$, the objective is to predict both the language token L and the corresponding text sequence, $T = \{T_1, T_2, \ldots, T_W\}$, where W is the number of words. The model is jointly trained in a sequence-to-sequence framework [Sutskever et al. 2014] to perform the tasks of language identification and lip-reading simultaneously.

The visual features are extracted using the pre-trained Visual Transformer Pooling (VTP) model [Prajwal et al. 2022a], which has a temporal footprint of 5 frames and serves as a robust visual backbone. It outputs per-frame lip-movement features, $f_{vtp} \in \mathbb{R}^{N_f \times d}$. We follow a simple Transformer encoder-decoder architecture from [Prajwal et al. 2022a]. The N-layer Transformer encoder temporally encodes the VTP features to produce $E_{enc} \in \mathbb{R}^{N_f \times d}$. This is followed by a N-layer Transformer Decoder, which cross-attends to the E_{enc} and auto-regressively predicts the language token and the text transcription.

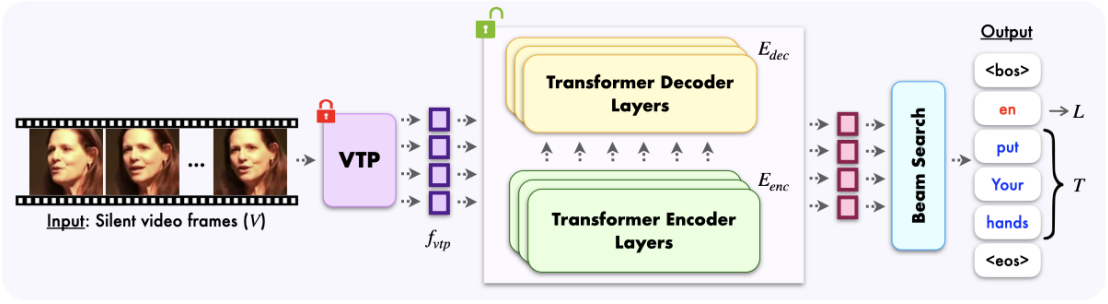


Figure 4.3: **Proposed multi-task architecture:** The input video frames are first passed through the Visual Transformer Pooling (VTP) model, which extracts the visual features, f_{vtp}. These features are then processed by a Transformer encoder-decoder model, generating the embeddings E_{dec}. Finally, the network outputs both the language token L for visual language identification and the text sequence T for lip-reading, using beam search.

4.4.1 Implementation details

Input/output data: We use Whisper's tokenizer [Radford et al. 2022] with a shared vocabulary of 51865 tokens for all the languages. This also includes the special tokens such as the start and end tokens, and language tokens. Using the start-end time of the word boundaries from WhisperX [Bain et al. 2023], we randomly sample a video clip between 2-10 seconds in length. For the mTEDx dataset, we use the full clips specified by [Yeo et al. 2024].

Hyper-parameters and Training: We set the number of layers N to 12, with a hidden dimension of 768 and number of attention heads as 12. We train the lip-reading model with the Adam optimizer with $\beta_1 = 0.9$, $\beta_2 = 0.98$ and an initial learning rate of $1e^{-4}$. We reduce the learning rate to $1e^{-5}$ when validation loss plateaus and train until convergence. Our model is trained on 4 Tesla $v100s$ GPUs for 4 days.

Inference: The first predicted token is always the language token L. Thus, the model directly performs language identification as part of the transcription. While decoding the text T, we use beam search with a beam width of 20. We do not use any external language model to improve the performance.

4.5 Experiments

4.5.1 Datasets

We leverage our newly collected large-scale dataset of approximately 15,700 hours, as outlined in Table 4.1, for both training and testing. A single model is jointly trained on all seven languages, for both lip-reading and VLID tasks. To compare with the existing multilingual works, we also report our results on mTEDx dataset [Salesky et al. 2021] as explained below.

Additional dataset – mTEDx: Multilingual TEDx (mTEDx) [Salesky et al. 2021] is a publicly available multilingual speech corpus, but lacks video data. Thus, we prepare the video data following [Yeo et al. 2024; M. Kim et al. 2023b] and use it to evaluate against the current lip-reading models. As done in prior state-of-the-art lip-reading works [Yeo et al. 2024; M. Kim et al. 2023b; Ma et al. 2022], we consider four languages for evaluation: Spanish, French, Portuguese, and Italian.

4.5.2 Lip-reading Results

Results on the *MultiVSR* dataset: We evaluate our model on the test set of our newly collected dataset across six languages and report the word error rate (WER). We do not evaluate on English, as the focus is on non-English languages. For comparison, we use the pre-trained model released by Yeo et al.[Yeo et al. 2024] and evaluate it on our test set. [Yeo et al. 2024] trains separate models for each language, whereas ours is a single model trained on all languages collec-

tively. As shown in Table 4.2, our model significantly outperforms the current state-of-the-art across all languages resulting in an impressive ≈ 23 WER average improvement over [Yeo et al. 2024] across four languages. The gains can largely be attributed to the (i) larger number of hours in each language and the quality of the transcriptions, and (ii) joint multilingual training to further increase the data scale.

Table 4.2: Performance comparison of lip-reading and visual language identification (VLID) across six languages on *MultiVSR* dataset.

Language	**Lip-reading WER ↓**		**VLID accuracy ↑**	
	[Yeo et al. 2024]	**Ours**	**[Prajwal et al. 2024]**	**Ours**
Spanish	60.5	**38.6**	88.4	**93.7**
French	65.4	**37.5**	69.8	**97.6**
Portuguese	59.5	**39.4**	62.1	**94.8**
Italian	59.8	**35.7**	44.6	**94.6**
Mandarin	n/a	**42.2**	90.6	**99.8**
Turkish	n/a	**49.5**	48.9	**93.5**

Results on mTEDx dataset: To validate the effectiveness of our proposed model and dataset, and to maintain consistency with previous lip-reading studies, we compare with the prior state-of-the-art [Yeo et al. 2024; Ma et al. 2022; M. Kim et al. 2023b] on the standard mTEDx dataset. We fine-tune on the train split specified in [Yeo et al. 2024] and report results on the unseen speakers of test clips [Yeo et al. 2024] in Table 4.3. In-line with our earlier findings, our model outperforms existing methods across all languages.

Table 4.3: Lip-reading WERs on mTEDx dataset [Salesky et al. 2021]. Lower is better.

Language	**[Ma et al. 2022]**	**[M. Kim et al. 2023b]**	**[Yeo et al. 2024]**	**Ours**
Spanish	56.30	56.96	45.71	**38.8**
French	66.20	64.92	58.30	**51.8**
Portuguese	61.50	58.57	47.89	**42.8**
Italian	57.40	59.74	51.79	**47.5**

4.5.3 Visual Language Identification (VLID) Results

Our network is also capable of identifying the language based solely on lip movements. Notably, we do not train a separate model for this task; instead, it is learned jointly during the lip-reading training process. Table 4.2 shows the results

of VLID across all the 6 languages. We compare with the pre-trained VLID model from Prajwal et al. [Prajwal et al. 2024]. Note that [Prajwal et al. 2024] trains a VLID network independently from the lip-reading task. From Table 4.2, it is evident that our model achieves a high accuracy of over 93% across all languages, which is an unprecedented improvement over the previous best model [Prajwal et al. 2024].

4.5.4 Extending to new low-resource languages

To address the challenge of training models for languages with limited data, we investigate the feasibility of using our pre-trained multilingual model for such low-resource languages. Specifically, we consider four languages for this study: Greek (134 hours), Japanese (152 hours), Polish (165 hours) and Russian (173 hours). We curate the training and testing data for these languages in the exact same manner as described in Section 4.3. We deem these languages low-resource as they contain only about 150 hours on average as opposed to at least 300+ hours from our set of 7 languages.

First, we train individual models for each language; the blue bars in Figure 4.4 show their performance. Next, we fine-tune our pre-trained multilingual model (trained on 15,700 hours) for each of these languages separately. As shown in Figure 4.4 (the green bars), fine-tuning leads to a significant performance boost, with over 15 WER improvement. This demonstrates the power of leveraging large-scale multilingual pre-training to enhance lip-reading performance, particularly for low-resource languages, setting the stage for broader and more inclusive applications of visual speech recognition.

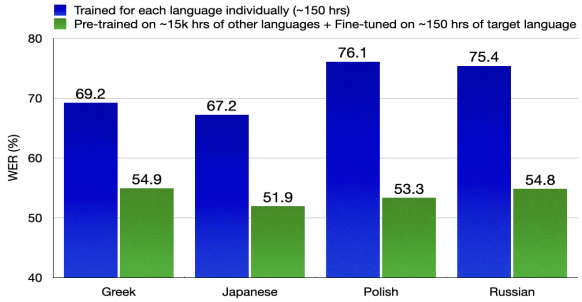


Figure 4.4: Lip-reading performance comparison on low-resource languages between models trained individually on each language (blue bar) and models fine-tuned on the target language using our pre-trained model (green bar).

4.5.5 Monolingual v/s Multilingual v/s Multilingual with additional fine-tuning

In Table 4.4, we assess three possible training strategies for lip-reading and discuss the trade-offs. The first case is training a separate model for each language. This significantly reduces the total training data, thus drastically hurting the performance. The second case is to train a single model on all languages jointly. In the third case, we fine-tune the model from the second case on the target language by removing the data of the other languages. No new data is added for the fine-tuning in this experiment, only the languages except the target language are removed. We can see in Table 4.4 that fine-tuning on the same target language data yields marginal improvements – barely 0.6 WER on average while also losing the language identification capabilities. On the other hand, the joint model is a single unified model that can identify and transcribe all languages with a comparable accuracy.

Table 4.4: Monolingual v/s multilingual v/s the effect of additional target language fine-tuning on *MultiVSR* dataset.

Language	**Monolingual**	**Multilingual (w/o FT)**	**Multilingual (with FT)**
Spanish	47.2	38.6	38.1
French	49.6	37.5	36.9
Portuguese	51.2	39.4	39.1
Italian	48.1	35.7	35.1
Mandarin	53.1	42.2	41.8
Turkish	65.5	52.1	51.7

4.6 Limitations and Conclusion

In this work, we built the largest multilingual lip-reading dataset to date. We train the first lip-reading model that can jointly identify the language and transcribe the utterance while surpassing performance of all prior works on both these tasks. A clear limitation of our dataset is that it has a large composition of English clips. While, our multilingual training strategy makes good use of this, a promising future direction would be to increase the number of languages and generalize to new languages in a few-shot manner.

Part II

Visual Keyword Spotting

Chapter 5

Visual Keyword Spotting with Attention

The paper has been accepted for publication at BMVC, 2021.

Visual Keyword Spotting with Attention

K R Prajwal* Liliane Momeni* Triantafyllos Afouras
Andrew Zisserman
Visual Geometry Group, University of Oxford, UK

Abstract

In this paper, we consider the task of spotting spoken keywords in silent video sequences – also known as visual keyword spotting. To this end, we investigate Transformer-based models that ingest two streams, a visual encoding of the video and a phonetic encoding of the keyword, and output the temporal location of the keyword if present. Our contributions are as follows: (1) We propose a novel architecture, the *Transpotter*, that uses full cross-modal attention between the visual and phonetic streams; (2) We show through extensive evaluations that our model outperforms the prior state-of-the-art visual keyword spotting and lip reading methods on the challenging LRW, LRS2, LRS3 datasets by a large margin; (3) We demonstrate the ability of our model to spot words under the extreme conditions of isolated mouthings in sign language videos.

5.1 Introduction

In recent years, there has been significant progress in automatic visual speech recognition (VSR) due to the availability of large-scale annotated datasets and the development of powerful neural network-based learners [J. S. Chung et al. 2017; Assael et al. 2016; Afouras et al. 2019]. These methods are continually improving and becoming more sophisticated, by incorporating better visual models, stronger

*Equal contribution.

language modelling and training on larger datasets. Indeed the best industrial grade lip reading models today are far superior to humans, and achieve error rates approaching Automatic Speech Recognition (ASR) performance [Makino et al. 2019; Prajwal et al. 2022a].

However, for many applications it is not necessary to transcribe every word that is spoken in a silent video (the task of VSR), rather only specific utterances or keywords need to be recognised. This is for example the case in "wake word" recognition, where only particular keywords need to be spotted over long input sequences. A further drawback of VSR methods is that they are heavily reliant on language modelling; in general, their performance decreases significantly when context is limited (e.g. short utterances) or parts of the input are occluded, e.g. from the speaker's hands or a microphone. In this work, we focus instead on the task of *Visual Keyword Spotting* (KWS), where the the goal is to detect and localise a *given* keyword in (silent) spoken videos.

Automatic visual KWS enables a diverse range of practical applications: indexing archival silent videos by keyword to enable content-based search; helping virtual assistants (e.g. Alexa and Siri) and smart home technologies respond to wake words and phrases; assisting people with speech impairment (e.g. amyotrophic lateral sclerosis patients) or aphonia in communication [Shillingford et al. 2019]; and detecting mouthings in sign language videos [Albanie et al. 2020].

KWS differs in complexity from VSR primarily because in KWS we are armed with the keyword we need to recognise, whereas VSR has the harder task of recognising every word from scratch. The core hypothesis motivating this work is that *this additional knowledge renders visual KWS an easier task* than VSR; and it is therefore expected that KWS should achieve a higher performance than VSR, and generally be more robust to challenging and adversarial situations. Nevertheless, visual KWS remains a *very difficult* task and shares similar challenges to VSR methods: first, some words sound different but involve identical lip movements ('man', 'pan', 'ban'), these *homopheme* words cannot be distinguished using only visual information. Second, speech variations such as accents, speed, and mumbling can alter lip movements significantly for the same word. Third, co-articulation of the lips between preceding and subsequent words in continuous speech also affects lip appearance and motion.

In this paper, we make the following three contributions: (i) We propose a novel

Transformer-based architecture, the *Transpotter* (a portmanteau of *Trans*former and *Spotter*), that is tailored to the visual KWS task. The model takes as input two streams, one encoding visual information from a video and the other providing a phonetic encoding of the keyword; the heterogeneous inputs are then fused using full cross-modal attention. (ii) Through extensive evaluations, we show that our Transpotter model outperforms the prior state-of-the-art visual KWS and VSR methods on the challenging LRW, LRS2 and LRS3 lip reading datasets by a large margin. (iii) We test our best model under extreme conditions: finding words in mouthings of people communicating using sign language. Signers sometimes mouth words as they sign as an additional non-manual signal to disambiguate and help understanding [Rachel Sutton-Spence 2007]. This new task is extremely challenging as there is a significant domain shift between full spoken sentences (in our training and test sets) and mouthings, where the context is sporadic and phonemes of the keyword may be missing – as sometimes only parts of words are mouthed [Boyes Braem and RL Sutton-Spence 2001]. Our approach outperforms previous KWS models in this challenging, practical use-case. Video examples are available at the project's webpage: www.robots.ox.ac.uk/~vgg/research/transpotter.

5.2 Related work

Our work relates to prior work on KWS, lip reading, visual grounding, and applications of Transformers for text and video. We present a brief discussion of these topics below.

KWS. KWS in audio (speech) is a well studied problem with a long history, spanning several decades. Prior to the establishment of deep learning models, KWS methods were based on Hidden Markov Models [Rose and Paul 1990; Wilpon et al. 1989], dynamic time warping [Itakura 1990; Sakoe and Chiba 1978; Yaodong Zhang and Glass 2010] or indexing of ASR lattices [Can and Saraçlar 2011]. A number of works have since used deep architectures suitable for sequence modelling (e.g. RNNs, CNNs, or graph convolutional networks) [Tara N. Sainath and Parada 2015; Yundong Zhang et al. 2017; Yuxuan Wang et al. 2017; Palaz et al. 2016; X. Chen et al. 2019; S. Fernandez et al. 2007; Hwang et al. 2015; M. Sun et al. 2016; Arik et al. 2017; Lengerich and Hannun 2016; Taejun Kim and Nam 2019], in-

cluding encoder-decoder approaches [Haitong Zhang et al. 2018; Audhkhasi et al. 2017; Zhuang et al. 2016; Rosenberg et al. 2017]. Berg *et al.* [Berg et al. 2021] recently proposed using a Transformer model for the same task. Different from ours, this work uses a single input stream (audio) and only learns to spot a fixed vocabulary of keywords. In contrast, we use Transformers to temporally process, then fuse the multi-modal inputs, building a model that can eventually perform open-set KWS. Visual KWS has also received attention recently. The proposed methods include query-by-example [Jha et al. 2018] approaches, sliding window classification [Y. Yao et al. 2019], or looking up phonetic queries in lip reading feature sequences [Stafylakis and Tzimiropoulos 2018; Momeni et al. 2020a], while audio-visual methods [P. Wu et al. 2016; Ding et al. 2018; Momeni et al. 2020a] that fuse the two modalities to improve robustness to noise have also been proposed. Our method builds upon these approaches: we address various weaknesses and propose superior video-text modelling as well as explicit keyword localization, resulting in significantly improved performance.

Lip reading. Early works in lip reading usually relied on hand-crafted pipelines and features [G. Potamianos et al. 2003; Gowdy et al. 2004; Papandreou et al. 2009; Ziheng Zhou et al. 2014]. The availability of large scale lip reading datasets [J. S. Chung et al. 2017; Afouras et al. 2018b] and the development of deep neural network models resulted in major performance improvements, initially in word-level lip reading [J. S. Chung and Andrew Zisserman 2016a; Stafylakis and Tzimiropoulos 2017] and constrained sentences [Assael et al. 2016]. Sentence level models were subsequently developed, using sequence-to-sequence architectures based on RNNs [J. S. Chung et al. 2017], CTC-based [Shillingford et al. 2019] approaches, or a hybrid of the two [Stavros Petridis et al. 2018b]. Replacing RNNs with Transformers resulted in better performing architectures [Afouras et al. 2019; X. Zhang et al. 2019; Gulati et al. 2020]. Joint audio-visual training and cross-modal distillation [Afouras et al. 2020a; Yu et al. 2020; W. Li et al. 2019] have also been investigated. The current state-of-the-art model uses Transformers in the visual front-end and achieves remarkable results with word error rates reaching as low as 30.7% [Prajwal et al. 2022a].

Visual grounding. Our work is also related to tasks such as natural language grounding in videos [Hendricks et al. 2017; Gao et al. 2017; M. Liu et al. 2018; H. Xu et al. 2019; Yitian Yuan et al. 2019; Ghosh et al. 2019; J. Chen et al. 2018;

Zeng et al. 2020] and subtitle alignment in sign language clips [Bull et al. 2021].

Transformers. Since their introduction for machine translation, Transformers [Vaswani et al. 2017] have become ubiquitous and are used today in a wide range of applications from natural language processing [Devlin et al. 2018; Radford et al. 2019] and speech recognition [L. Dong et al. 2018; Karita et al. 2019; Mohamed et al. 2019] to visual representation learning [Dosovitskiy et al. 2021; Bertasius et al. 2021; B. Wu et al. 2020; Prajwal et al. 2022a]. In this work, we rely on Transformers as our building blocks for their strong sequence modelling capability and inherent potential for localisation through attention.

5.3 Visual KWS with Attention

In this section, we describe our proposed method shown in Figure 5.1. We outline the architecture of our model (Section 5.3.1), our training procedure (Section 5.3.2) and differences to prior work (Section 5.3.3). We refer the reader to the arXiv version of the paper for further details.

5.3.1 The Transpotter Architecture

Our model ingests two input streams: (i) a textual keyword $q = (q_1, q_2 \cdots, q_{n_p})$, and (ii) a silent video clip $v \in \mathbb{R}^{T \times H \times W \times 3}$ in which we need to spot the keyword. For each of the inputs, we have separate encoders that learn initial modality-specific representations. This is followed by a joint multi-modal Transformer that learns cross-modal relationships between the video and text features. The joint transformer predicts two outputs: (i) a sequence-level probability of the keyword occurring in the video and (ii) frame-level probabilities indicating the location of the keyword in the video if present. We describe each of the modules next.

Text Representations. Our textual input is a phonetic representation of the keyword, obtained using a pronunciation dictionary. The input phoneme sequence of length n_p is mapped to a sequence of learnable embedding vectors $Q \in \mathbb{R}^{n_p \times d}$. Sinusoidal positional encodings are added to the input phoneme feature vectors, and the result is passed through a Transformer Encoder [Vaswani et al. 2017] with N_t layers to capture temporal information across the phoneme sequence:

$$Q_{enc} = encoder_q(Q + PE_{1:n_p}) \in \mathbb{R}^{n_p \times d}.$$

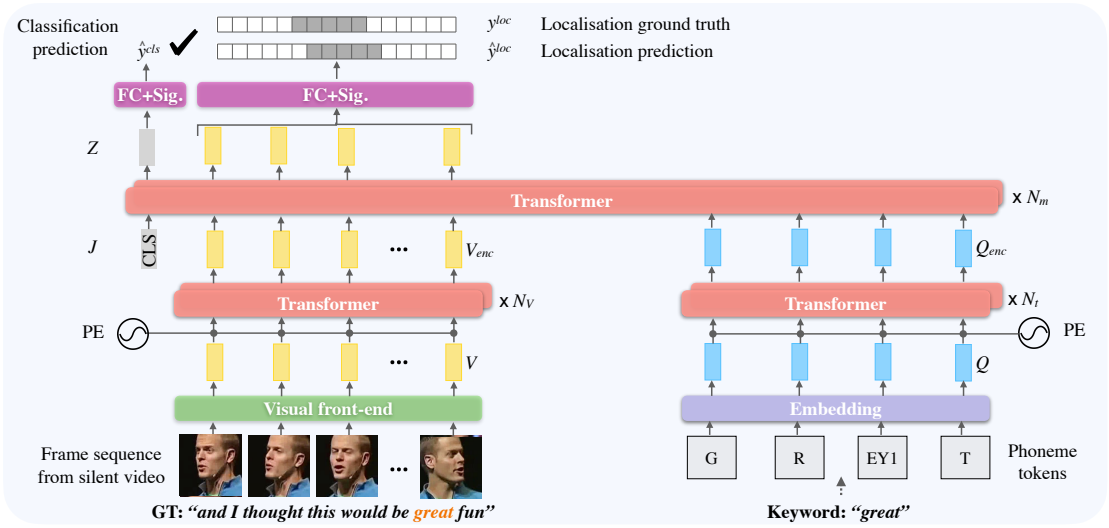


Figure 5.1: **The Transpotter architecture:** Video frames are inputted to a visual front-end (CNN [Afouras et al. 2019] or VTP [Prajwal et al. 2022a]) to extract low-level visual features, which are then passed to N_v Transformer layers to encode temporal information. The keyword in the form of a phoneme sequence is encoded using N_t Transformer layers. The text and visual features are finally concatenated in time and processed using a joint multi-modal Transformer which predicts: (i) the probability the keyword occurs in the video, (ii) frame-level probabilities indicating the location of the word. PE corresponds to positional encoding.

Video Representations. We use a pre-trained visual front-end (either a CNN [Afouras et al. 2019] or VTP [Prajwal et al. 2022a]) to extract a feature vector for each input video frame, $V \in \mathbb{R}^{T \times d}$. Similar to the text encoder Q_{enc}, we pass the visual features through a Transformer Encoder with N_v layers to capture temporal information, after adding positional encodings:

$$V_{enc} = encoder_v(V + PE_{1:T}) \in \mathbb{R}^{T \times d}.$$

Joint Video-Text Representations. The uni-modal representations V and Q are concatenated along the time dimension to produce a single sequence of feature vectors. A learnable $[CLS]$ token embedding (such as in BERT [Devlin et al. 2018] and ViT [Dosovitskiy et al. 2021]) is then prepended to the result:

$$J = ([CLS]; V_{enc}; Q_{enc}) \in \mathbb{R}^{(1+T+n_p) \times d}.$$

We use a Transformer encoder with N_m layers to jointly learn the relationships

across video and phoneme vectors:

$$Z = encoder_{vq}(J + PE_{1:(1+T+n_p)}) \in \mathbb{R}^{(1+T)\times d}.^{1}$$

Prediction heads. The $[CLS]$ output feature vector Z_1 serves as a joint aggregate representation for the video-text pair. An MLP head for binary classification, f_c is attached to Z_1 to predict the probability of the keyword being present in the video:

$$\hat{y}^{cls} = \sigma(f_c(Z_1)) \in \mathbb{R}^1,$$

where σ denotes a sigmoid activation. To localise the keyword, we attach a second MLP head f_l that is shared across all the video output states from the multi-modal joint Transformer:

$$\hat{y}^{loc} = \sigma(f_l(Z_{2:(T+1)})) \in \mathbb{R}^T.$$

The output y_t^{loc} at each video frame time-step $t \in T$ indicates the probability of the frame t being a part of the keyword utterance.

5.3.2 Training

Optimisation objectives. Given a training dataset $\mathcal{D}$ consisting of tuples (v, q, y^{cls}, y^{loc}) of silent video clips, text queries, class labels and location labels (indicating the position of the keyword within the clip), we define the following objectives:

$$\mathcal{L}^{cls} = -\mathbb{E}_{(v,q,y^{cls})\in\mathcal{D}}\ BCE(y^{cls}, \hat{y}^{cls}) \tag{5.1}$$

$$\mathcal{L}^{loc} = -\mathbb{E}_{(v,q,y^{cls},y^{loc})\in\mathcal{D}}\ y^{cls}\left[\frac{1}{T}\sum_{t=1}^{T} BCE(y_t^{loc}, \hat{y}_t^{loc})\right] \tag{5.2}$$

$$BCE(y, \hat{y}) = y\log\hat{y} + (1-y)\log(1-\hat{y}), \tag{5.3}$$

where BCE stands for the binary cross-entropy loss. The labels y^{cls} are set to 1 when the given keyword occurs in the video and 0 otherwise; the frame labels y^{loc} are set to 1 for the frames where the keyword is uttered and 0 otherwise. To train the model the optimise the total loss $\mathcal{L} = \lambda\mathcal{L}^{cls} + (1-\lambda)\mathcal{L}^{loc}$, where λ is a balancing hyper-parameter.

[1] the n_p outputs corresponding to the phonetic embeddings are dropped.

5.3.3 Discussion

Compared to prior approaches, the design of our model offers several important advantages.

Stronger Visual Representations. Previous works [Stafylakis and Tzimiropoulos 2018; Momeni et al. 2020a] model temporal relationships between video frames using RNNs. In contrast, we employ Transformers [Vaswani et al. 2017], which are far more effective in modeling temporal relationships [Al-Rfou et al. 2019; Hochreiter et al. 2001].

Joint Video-text Modeling. Prior works such as KWS-Net [Momeni et al. 2020a] follow a late-fusion strategy. In our model each frame-wise video feature can attend to any keyword token (phoneme) and vice-versa. The information exchange across the modalities occurs at every layer, without restrictions on the receptive field for either modality.

Stronger keyword localisation. Fine-grained localisation of the keyword in the video can be important for applications such as sign spotting [Albanie et al. 2020]. Existing methods [Momeni et al. 2020a; Stafylakis and Tzimiropoulos 2018] "weakly" localise the keyword by taking the sequence-level prediction to be the maximum probability over all the video time-steps. We instead provide stronger frame-level supervision, by enforcing the model to predict the exact temporal extent of the keyword in the video.

5.3.4 Implementation details

Pre-training the visual front-end. We explore two different visual front-end architectures for the Transpotter: (1) a CNN, highly similar in architecture to TM-seq2seq [Afouras et al. 2019] and (2) VTP [Prajwal et al. 2022a], the current state-of-the-art for lip reading (trained only on public data). Both models are trained end-to-end on two-word video clips of LRS2 [J. S. Chung et al. 2017] and LRS3 [Afouras et al. 2018b] for lip reading. We refer the reader to the arXiv version of the paper for the exact CNN architecture and training hyper-parameters. We refer the reader to [Prajwal et al. 2022a] for architectural hyper-parameters and training protocols for VTP. We pre-compute the visual features for each backbone for both datasets and then train directly on them for faster training. All our models and ablations use the pre-trained CNN features, unless otherwise stated.

Sampling. We form the training dataset $\mathcal{D}$ by randomly sampling with 50% probability a positive or negative video clip v for each query q. Each video v contains word boundary annotations, which allows (i) performing data augmentation by randomly cropping video segments during training, and (ii) creating frame labels y^{loc}, as described in 5.3.2.

Misc. The keyword q is mapped to a phoneme sequence using the CMU dictionary [Speech Group at Carnegie Mellon University 2014]; words not present in the dictionary are discarded from training $\mathcal{D}$. We set $\lambda = 0.5$.

5.4 Experiments

This section is structured as follows: We first present the datasets used as well the evaluation protocols that we follow in our experiments (Section 5.4.1). Next, we compare the performance of our proposed Transpotter model against strong baselines (Section 5.4.2) and then present a comprehensive study ablating our design choices (Section 5.4.3). Finally, we perform further performance analysis and provide qualitative results (Section 5.4.4).

5.4.1 Datasets and Evaluation Protocol

Datasets. All models and baselines are trained and evaluated on LRS2 [J. S. Chung et al. 2017] and LRS3 [Afouras et al. 2018b] lip reading datasets. LRS2 contains BBC broadcast footage from British television and LRS3 is based on TED/TEDx videos downloaded from YouTube (refer to the arXiv version of the paper for detailed statistics). The video clips for both datasets are tightly cropped face-tracks of active speakers only. For each clip, a full transcription of the utterance as well as word boundary alignments are provided. The number of videos, number of keyword instances and keyword vocabulary for each of the test sets is shown in Table 5.1.

Evaluation Protocol. Evaluation is performed for every test dataset as follows: First, the vocabulary of test keywords is determined, by considering all the words occurring in the test set transcriptions with above a certain phoneme length n_p. If not specified, we use $n_p \geq 3$. Every word in the query vocabulary is then searched for in all the test set videos.

Metrics. Given ground truth video-keyword samples, we assess the performance of our model in two ways. First, we assess classification performance, *i.e.* whether the model can accurately predict whether the keyword occurs in the video or not. We compute accuracy ($\text{Acc}^{Cls}_{@k}$) and mean average precision (mAP^{Cls}) metrics, where $\text{Acc}^{Cls}_{@k}$ measures how often a given keyword occurs in any of the top-k retrievals, and mAP^{Cls} is obtained with the above criterion (where every word in the test keyword vocabulary is considered as a separate class).

Second, we assess the model's localisation capability, *i.e.* whether the model can accurately localise the keyword in the video clip. We follow common practice from the detection literature: we consider a keyword accurately detected when the intersection-over-union (IOU) between the prediction $\hat{y}^{loc}$ and ground truth label y^{loc} is above a certain threshold, and calculate the mean average precision mAP^{Loc}. To calculate the IOU, we binarise the model's predictions using a threshold $\tau = 0.5$.

5.4.2 Comparison to baselines

We compare our model's performance against a state-of-the-art VSR model and KWS-Net [Momeni et al. 2020a], the previous state-of-the-art visual KWS model.

VSR baseline. We use an improved version of the TM-seq2seq [Afouras et al. 2019] VSR model, with the same pre-trained CNN backbone (Section 5.3.4) that we use for the KWS models. The model is trained with the curriculum training strategy of [Afouras et al. 2019] (details in the arXiv version of the paper). The VSR model achieves state-of-the-art Word Error Rate (WER) performance of 36.9% and 48.0% on the LRS2, LRS3 test sets respectively. Since the VSR model only produces text transcriptions of a given video, but no localisation prediction, we can only evaluate its classification performance ($\text{Acc}^{Cls}_{@k}$,mAP^{Cls}). We follow the method detailed in [Y. He et al. 2017] to estimate the posterior probability that the keyword occurs in a video clip.

KWS-Net. As a KWS baseline we use the state-of-the-art model of Momeni *et al.* [Momeni et al. 2020a]. For fair comparison, here too we use the same CNN backbone that is also used for our model.

State-of-the-art KWS. We report our model's performance and compare it with strong baselines in Table 5.1. It is clear that our model outperforms both baselines. On the last row, we show the boost in performance by replacing the CNN with the

	LRS2				LRS3			
	1.2K vids. / 4.3K inst. / 1.6K vocab.				1.3K vids. / 6.1K inst. / 1.9K vocab.			
Model	$\text{Acc}^{Cls}_{@1}$	$\text{Acc}^{Cls}_{@5}$	mAP^{Cls}	mAP^{Loc}	$\text{Acc}^{Cls}_{@1}$	$\text{Acc}^{Cls}_{@5}$	mAP^{Cls}	mAP^{Loc}
KWS-Net [Momeni et al. 2020a]	36.1	61.2	41.0	36.2	29.8	54.6	34.3	29.2
VSR	63.7	76.3	64.3	-	52.3	66.0	50.3	-
Transpotter	65.0	87.1	69.2	68.3	52.0	77.1	55.4	53.6
Transpotter (VTP)	**68.7**	**90.7**	**72.5**	**71.6**	**55.7**	**78.5**	**58.2**	**56.1**

Table 5.1: **Comparison to baselines:** We outperform the current state-of-the-art KWS and VSR methods by a large margin. Our Transpotter model is particularly effective in localising the keyword in the video. Moreover, by using the recently proposed VTP [Prajwal et al. 2022a] architecture as the Transpotter's visual backbone instead of a CNN, we achieve even better performance.

recently proposed VTP backbone [Prajwal et al. 2022a], resulting in state-of-the art performance on both the LRS2 and LRS3 datasets.

Evaluation on LRW. We also compare the performance of KWS-Net [Momeni et al. 2020a] with our proposed Transpotter model on the LRW [J. S. Chung and Andrew Zisserman 2016a] test set following the same evaluation protocol. The test set contains $25K$ single-word video clips spanning a vocabulary of 500 words (50 instances per word). Note that KWS-Net has been pretrained on the LRW training split, but the Transpotter has only been trained on LRS2 and LRS3. As we can see in Table 5.2, the Transpotter outperforms the previous state-of-the-art baseline KWS-Net by a large margin. We refer the reader to the arXiv version of the paper for a qualitative error analysis in this setting.

Model	$\text{Acc}^{Cls}_{@1}$	$\text{Acc}^{Cls}_{@5}$	mAP^{Cls}
KWS-Net [Momeni et al. 2020a]	66.6	89.0	33.0
Transpotter	**85.8**	**99.6**	**64.1**

Table 5.2: **Comparison on LRW [J. S. Chung and Andrew Zisserman 2016a]:** The Transpotter outperforms the previous state-of-the-art KWS model on the LRW test set, despite not having been trained on LRW data. The localization metric mAP^{Loc} is not reported as the input videos are single-word clips.

5.4.3 Architecture ablations

To assess our design choices for the Transformer skeleton, we perform a number of ablations considering variations of the model architecture. We briefly explain the alternative approaches below; more details can be found in the arXiv version of the paper.

In particular we consider two alternative encoder-decoder architectures, with

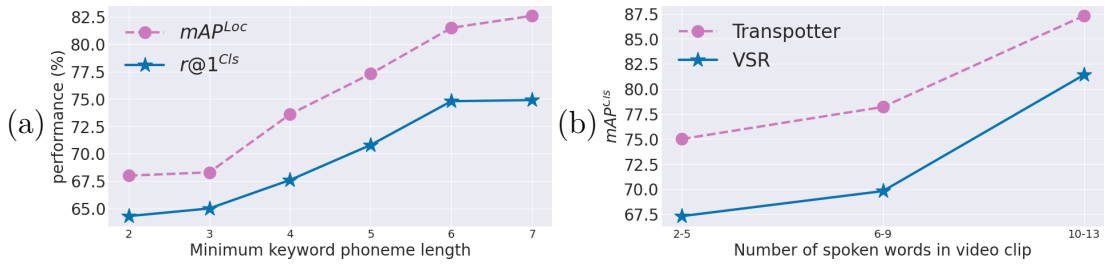


Figure 5.2: (a) Transpotter's performance increases with the keyword length; (b) Transpotter performs far better than VSR with limited context. Both methods improve with more context.

the video input at the encoder side and the text query at the decoder (Enc_{vid}-Dec_{text}) and vice versa (Enc_{text}-Dec_{vid}). Since the latter model outputs at the temporal resolution of the video input, it can explicitly localise the keyword (in the same way as the Transpotter), while the former can only perform classification. We also consider a variant of the Transpotter, where the model does not output localisation predictions (hence no $\mathcal{L}^{loc}$ is used for its training). We show the results in Table 5.3. The selected Transpotter architecture outperforms all variants. In particular, by comparing rows 2 and 4, we observe that training with a localisation head and loss $\mathcal{L}^{loc}$ also improves classification (e.g. 64.0 vs 69.2 mAP^{Cls}).

	LRS2				LRS3			
Model	$\text{Acc}^{Cls}_{@1}$	$\text{Acc}^{Cls}_{@5}$	mAP^{Cls}	mAP^{Loc}	$\text{Acc}^{Cls}_{@1}$	$\text{Acc}^{Cls}_{@5}$	mAP^{Cls}	mAP^{Loc}
Enc_{vid}-Dec_{text}	52.5	80.0	57.9	-	40.3	66.9	43.2	-
Transpotter w/o loc.	59.4	84.1	64.0	-	46.5	72.1	49.8	-
Enc_{text}-Dec_{vid}	63.8	86.8	68.4	67.8	**52.1**	76.6	54.9	53.1
Transpotter	**65.0**	**87.1**	**69.2**	**68.3**	52.0	**77.1**	**55.4**	**53.6**

Table 5.3: **Model ablations:** Our approach of jointly modeling text and video sequences with a localisation head for stronger supervision outperforms other architectural designs.

5.4.4 Transpotter performance analysis

In this section, we analyse the performance of our proposed method when varying the keyword length and the size of the surrounding visual context.

Keyword length. In Figure 5.2a, we plot the model's performance on the LRS2 test set against the minimum keyword length in phonemes n_p. As expected, longer keywords are easier to spot and therefore result in better retrieval performance. Indeed for long 7-phoneme keywords, mAP^{Loc} reaches as high as 82.5. We note however that even for very challenging short keywords with only 2 phonemes (such as "my", "to", "at"), mAP^{Loc} stays high at 67.5.

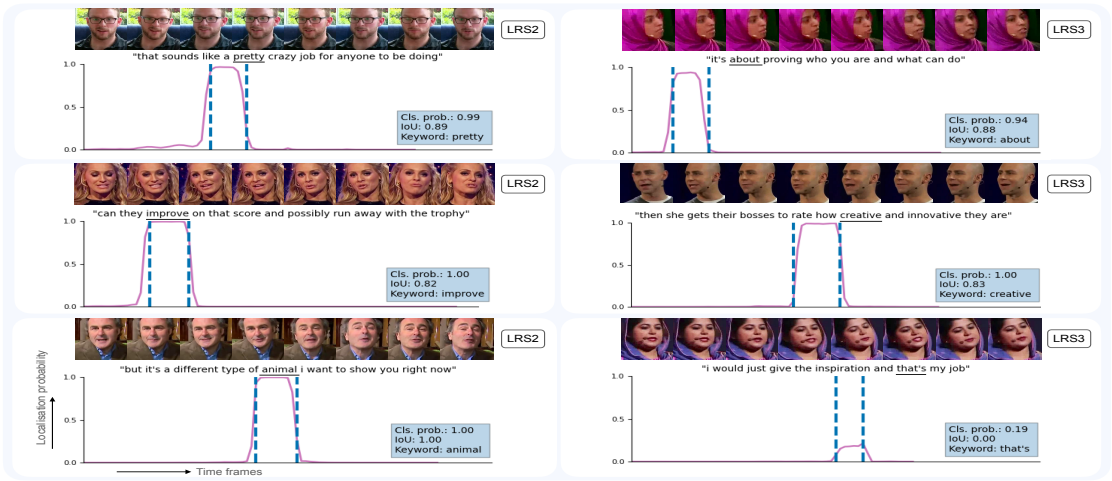


Figure 5.3: **Qualitative results on LRS2 and LRS3:** The Transpotter accurately localises the keyword in most examples. In the bottom right example, the model's confidence is low, most likely because it is a short word. The IOU is zero since we threshold at $\tau = 0.5$.

Context. The visual appearance of spoken words can be highly ambiguous [Afouras et al. 2019], therefore recognising isolated words from visual input alone may be very challenging. Current lip reading models utilise the surrounding visual context to resolve this ambiguity. In Figure 5.2b, we illustrate how the performances of our Transpotter KWS model and our VSR baseline vary based on the amount of contextual information available. We plot the mAP^{Cls} against the number of words in the video clip. We observe that both models benefit from larger surrounding context, with the Transpotter outperforming the VSR baseline consistently.

Qualitative analysis. In Figure 5.3, we show qualitative examples from the LRS2 and LRS3 test sets. It is clear that the model produces smooth predictions that precisely indicate the full location of the word. In the bottom right corner we observe a failure case where the model's confidence is low – the keyword "that's" in this case is short.

Model response to homophemes. We further probe our Transpotter model for failure cases. In visual-only keyword spotting, a common failure case is due to homophemes, *i.e.* words with identical lip movements. To investigate the response of our model to such cases, we construct a list of keywords from the LRS2 test set sentences that are known to have homopheme counterparts (*e.g.mark*, which has two matching homophemes, *bark* and *park*) and then for each test set clip that contains one of the keywords, we query that keyword along with its corresponding homophemes and plot the model's outputs. We illustrate several examples in

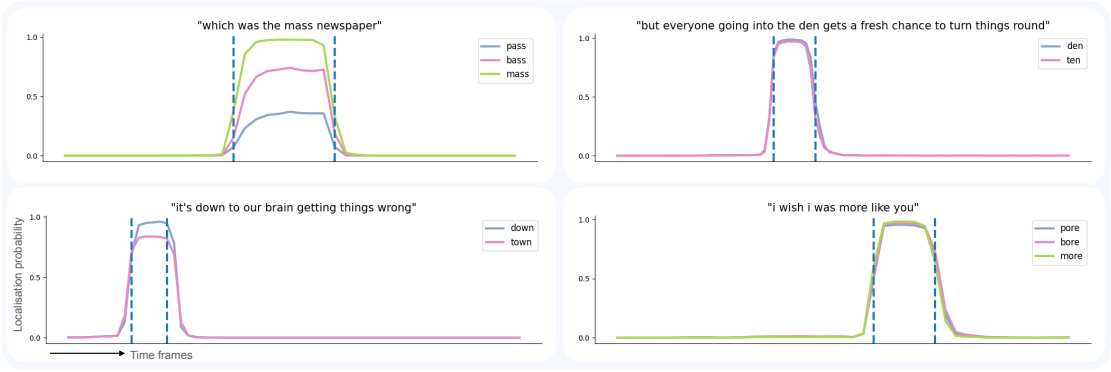


Figure 5.4: **Model's response to homophemes:** We query words and their corresponding homophemes for LRS2 test set clips. We observe that the model spots the words and their homophemes at the same (ground truth) location.

Figure 5.4. We observe that in such cases, the model spots the keyword as well as its homophemes at the same (ground truth) location.

5.5 Mouthing Spotting in Sign Language videos

In this section, we investigate the application of our method for spotting mouthed words in sign language videos. This is an important application of visual KWS, as it has enabled an entire line of work on sign language recognition [Albanie et al. 2020; Momeni et al. 2020b; Varol et al. 2021].

Data description & evaluation protocol. Here, we use a subset of BSL Corpus [Schembri et al. 2017] as a test set. BSL Corpus is a large public dataset containing videos of conversations conducted in sign language by deaf signers, from various regions across the UK. We extend the dataset's annotations by adding a *Mouthing* tier and asking a deaf annotator to identify and localise mouthing occurrences that correspond to visible signs. We obtain 383 mouthing instances, from 29 different signers, over a keyword vocabulary size of 187. We use a pre-processing pipeline similar to [J. S. Chung et al. 2017] to obtain face-cropped tracks around the faces of the signers. To evaluate KWS performance, we take 8-second video clips centered around the annotated mouthings and follow the same evaluation protocol described in Section 5.4.

Results. We summarise the evaluation results in Table 5.4. The Transpotter model is far superior to the prior state-of-the-art KWS baseline, achieving a great improvement in performance (e.g. 29.6 vs 15.6 mAP^{Cls} score). To complete this analysis, we also show qualitative examples of the spotted mouthings in Figure 5.5.

Discussion. We note that sign language mouthings are often very different from equivalent spoken words. Words may be partially mouthed and can be occluded by the signing hands. There is therefore a significant domain gap between the BSL-Corpus signing videos and our lip reading training videos. However, we note that our proposed model greatly outperforms the KWS-Net baseline – a variant of which has been successfully deployed for detecting mouthings in order to bootstrap learning of sign spotting methods [Albanie et al. 2020; Momeni et al. 2020b; Varol et al. 2021]. This indicates the potential of our proposed method to greatly improve these pipelines.

Model	$\text{Acc}^{Cls}_{@1}$	$\text{Acc}^{Cls}_{@5}$	mAP^{Cls}
KWS-Net [Momeni et al. 2020a]	12.4	29.6	15.6
Transpotter	**22.5**	**47.1**	**29.6**

Table 5.4: **Spotting mouthings in BSL-Corpus:** The Transpotter is far more accurate than the current state-of-the-art in spotting keywords in videos.

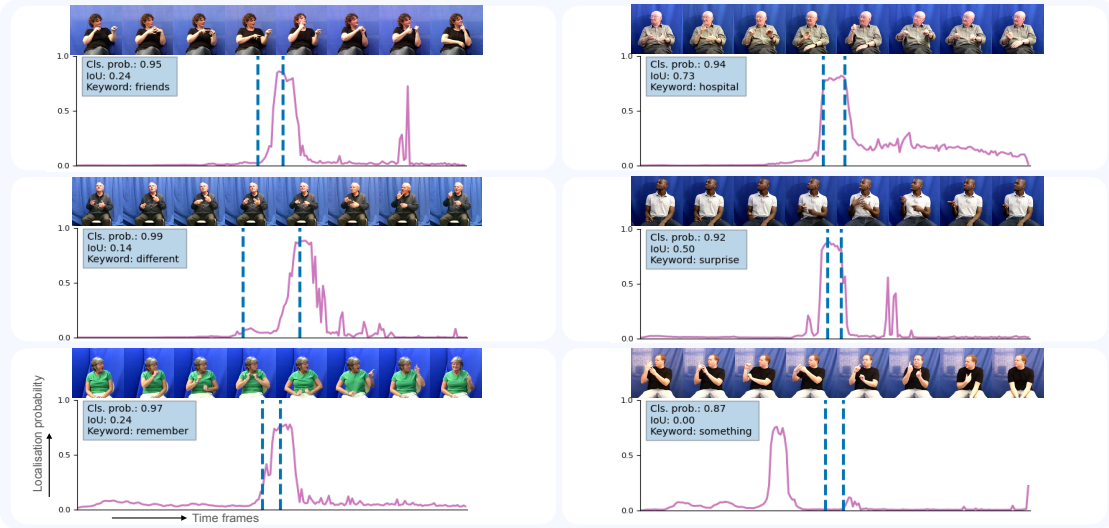


Figure 5.5: **Qualitative results on BSL-Corpus:** Despite the large domain shift from our training examples and additional challenges such as partially mouthed words and hand occlusions, the Transpotter succeeds in correctly spotting mouthings in these challenging conditions. We observe a failure case (bottom right) where the localisation is incorrect. We note that contrary to LRS2 and LRS3, where word boundaries are obtained through robust audio-based forced alignment, the annotations for BSL-Corpus are noisier as they are performed manually.

5.6 Conclusion

We have presented the *Transpotter*, a cross-modal attention based architecture for visual keyword spotting. Our method surpasses the performance of the previous

best visual keyword spotting approach by a large margin, as well as that of a state-of-the-art lip reading baseline. We demonstrate the ability of our model to generalise to sign language videos where it can be used to spot mouthings, enabling automatic annotation of sign instances. In future work, we plan to further improve our method's performance by incorporating keyword semantics and context of the surrounding words.

Acknowledgements. Funding for this research is provided by the UK EPSRC CDT in Autonomous Intelligent Machines and Systems, the Oxford-Google DeepMind Graduate Scholarship, the EPSRC Programme Grant VisualAI (EP/T028572/1) and the Royal Society Research Professorships 2019 RP/R1/191132. We thank Samuel Albanie for his invaluable help in applying our method to signer mouthings.

Part III

Sign Language Recognition

Chapter 6

Automatic Dense Annotation of Large-vocabulary Sign Language Videos

The paper has been accepted for publication at the European Conference on Computer Vision (ECCV), 2022.

Automatic dense annotation of large-vocabulary sign language videos

Liliane Momeni[1*] Hannah Bull[2*] K R Prajwal[1*]
Samuel Albanie[3] Gül Varol[2] Andrew Zisserman[1]
[1] Visual Geometry Group, University of Oxford, UK
[2] LIGM, École des Ponts, Univ Gustave Eiffel, CNRS, France
[3] Department of Engineering, University of Cambridge, UK

Abstract

Recently, sign language researchers have turned to sign language interpreted TV broadcasts, comprising (i) a video of continuous signing and (ii) subtitles corresponding to the audio content, as a readily available and large-scale source of training data. One key challenge in the usability of such data is the lack of sign annotations. Previous work exploiting such weakly-aligned data only found *sparse* correspondences between keywords in the subtitle and individual signs. In this work, we propose a simple, scalable framework to *vastly* increase the *density* of automatic annotations. Our contributions are the following: (1) we significantly improve previous annotation methods by making use of synonyms and subtitle-signing alignment; (2) we show the value of pseudo-labelling from a sign recognition model as a way of sign spotting; (3) we propose a novel approach for increasing our annotations of *known* and *unknown* classes based on *in-domain exemplars*; (4) on the BOBSL BSL sign language corpus, we increase the number of confident automatic annotations from 670K to 5M. We make these annotations publicly available to support the sign language research community.

*Equal contribution.

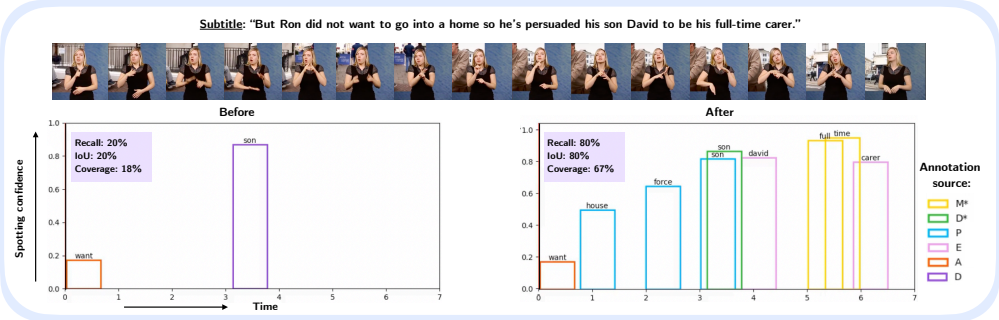


Figure 6.1: **Densification**: For continuous sign language, we show automatic sign annotation timelines, along with their confidence and annotation source, *before* and *after* our framework is applied. M, D, A refer to automatic annotations from previous methods from mouthings [Albanie et al. 2020], dictionaries [Momeni et al. 2020b] and the Transformer attention [Varol et al. 2021]. M*, D*, P, E, N refer to new and improved automatic annotations collected in this work. Annotation methods are compared in the appendix.

6.1 Introduction

Sign languages are visual-spatial languages that have evolved among deaf communities. They possess rich grammar structures and lexicons that differ considerably from those found among spoken languages [Rachel Sutton-Spence and Woll 1999a]. An important factor impeding progress in automatic sign language recognition – in contrast to automatic speech recognition – has been the lack of large-scale training data. To address this issue, researchers have recently made use of sign language interpreted TV broadcasts, comprising (i) a video of continuous signing, and (ii) subtitles corresponding to the audio content, to build datasets such as Content4All [N. Camgoz et al. 2021] (190 hours) and BOBSL [Albanie et al. 2021b] (1460 hours).

Although such datasets are orders of magnitude larger than the long-standing RWTH-PHOENIX [N. C. Camgoz et al. 2018] benchmark (9 hours) and cover a much wider domain of discourse (not restricted to only weather news), the supervision they provide on the signed content is limited in that it is *weak* and *noisy*. It is weak because the subtitles are temporally aligned with the audio content and not necessarily with the signing. The supervision is also noisy because the presence of a word in the subtitle does not necessarily imply that the word is signed; and subtitles can be signed in different ways. Recent works have shown that training automatic sign language translation models on such *weak* and *noisy* supervision leads to low performance [N. Camgoz et al. 2021; Varol et al. 2021; Albanie et al.

2021b].

In an attempt to increase the value of such interpreted datasets, multiple works [Albanie et al. 2020; Momeni et al. 2020b; Varol et al. 2021] have leveraged the subtitles to perform lexical *sign spotting* in an approximately aligned continuous signing segment – where the aim is to determine *whether* and *when* a subtitle word is signed. Methods include using visual keyword spotting to identify signer mouthings [Albanie et al. 2020], learning a joint embedding with sign language dictionary video clips [Momeni et al. 2020b], and exploiting the attention mechanism of a transformer translation model trained on weak, noisy subtitle-signing pairs [Varol et al. 2021]. These works leverage the approximate subtitle timings and subtitle content to significantly reduce the correspondence search space between temporal windows of signs and spoken language words. Although such methods are effective at automatically annotating signs, they only find *sparse* correspondences between keywords in the subtitle and individual signs.

Our goal in this work is to produce *dense* sign annotations, as shown in Fig. 6.1. We define densification in two ways: (i) reducing gaps in the timeline so that we have a densely spotted signing sequence; and also (ii) increasing the number of words we recall in the corresponding subtitle. This process can be seen as automatic annotation of lexical signs. Automatic dense annotation of large-vocabulary sign language videos has a large range of applications including: (i) *substantially* improving recall for retrieval or intelligent fast forwards of online sign language videos; (ii) enabling *large-scale* linguistic analysis between spoken and signed languages; (iii) providing *supervision* and *improved alignment* for continuous sign language recognition and translation systems.

In this paper, we ask the following questions: (1) Can we improve current methods to improve the yield of automatic sign annotations whilst maintaining precision? (2) Can we increase the vocabulary of annotated signs over previous methods? (3) Can we 'fill in the gaps' that current spotting methods miss? The answer is yes, to all three questions, and we demonstrate this on the recently released BOBSL dataset of British Sign Language (BSL) signer interpreted video.

We make the following four contributions: (1) we significantly improve previous methods by making use of synonyms and subtitle-signing alignment; (2) we show the value of pseudo-labelling from a sign recognition model as a way of sign spotting; (3) we propose a novel approach for increasing our annotations of *known*

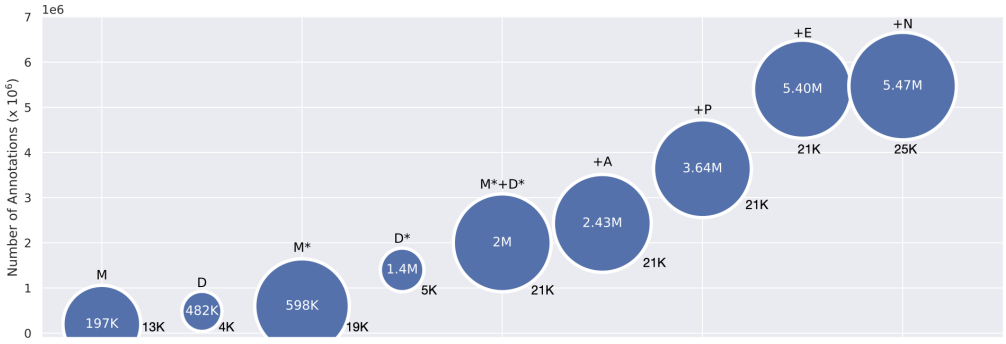


Figure 6.2: **Yield of automatic annotations and vocabulary size:** We highlight the increase in the number of automatic annotations and vocabulary size at each stage in our proposed framework. M, D, A refer to annotations from previous methods. M*, D*, P, E, N refer to new and improved annotations collected in this work. The number of annotations is shown within each circle. The vocabulary size is reported below each circle and also represented by the circle diameter.

and *unknown* sign classes based on in-domain exemplars; (4) we will make all 5 million automatic annotations publicly available to support the sign language research community. Our increased yield and vocabulary size is shown in Fig. 6.2. Our final vocabulary of 24.8K represents the vocabulary of English words (including named entities) from the subtitles which have been automatically associated to a sign instance; different words may have the same sign.

We note that this work is focused on *interpreted* data, which can differ from *conversational* signing in terms of style, vocabulary and speed [Bragg et al. 2019]. Although our long-term aim is to move to conversational signing, learning good representations of signs from interpreted data can be a 'stepping stone' in this direction. Moreover, non-lexical signs, such as a pointing sign and spatially located signs, are essential elements of sign language, but our method is limited to the annotation of lexical signs associated to words in the text.

6.2 Related Work

Our work relates to several themes which we give a brief overview of below.

Sign Spotting. One line of research has focused on the task of *sign spotting*, which seeks to detect signs from a given vocabulary in a target video. Early efforts for sign spotting employed lower-level features (colour histograms and geometric cues) in combination with Conditional Random Fields [H.-D. Yang et al. 2008], Hidden Markov Models (HMMs) [Viitaniemi et al. 2014] and Sequential Interval

Patterns [E.-J. Ong et al. 2014] for temporal modelling. A related body of work has sought to localise signs while leveraging weak supervision from audio-aligned subtitles. These include the use of external dictionaries [D. Li et al. 2020b; Momeni et al. 2020b; T. Jiang et al. 2021] and other localisation cues such as mouthing [Albanie et al. 2020] and Transformer attention [Varol et al. 2021]. The performance of these approaches depends on the quality of the visual features, keywords, and the search window. In this work, we show improved yield of existing sign spotting techniques by employing automatic subtitle alignment techniques to adjust the time window and incorporating synonyms when forming the keywords. Going further beyond the spotting task explored in prior work, we use the automatic spottings to initiate additional algorithms for sign discovery based on *in-domain* exemplar matching (6.3.1). This is similar to dictionary-based sign spotting techniques [Momeni et al. 2020b; T. Jiang et al. 2021] except we do not source the exemplars from external dictionaries, avoiding the domain gap issue. Besides in-domain *sign* exemplars as in [T. Jiang et al. 2021], we explore the weak *subtitle* exemplars with unknown sign locations.

A recent progress in mouthing-based keyword spotting was presented by *Transpotter* [Prajwal et al. 2021]. This architecture comprises a transformer joint encoder of visual features and phoneme features that is trained to regress both the presence and location of the target keyword in a sequence from mouthing patterns. Preliminary small-scale experimental results reported by Prajwal et al. [Prajwal et al. 2021] demonstrated that Transpotter can perform visual keyword spotting in signing footage. Here, we showcase its suitability for the large-scale annotation regime, and further train it on sign language data to obtain a greater density of sign annotations.

In this work, we demonstrate the additional value of *pseudo-labelling* [Yarowsky 1995; D.-H. Lee et al. 2013] with a sign classifier as an effective mechanism for sign spotting. While pseudo-labelling has been explored previously for category-agnostic sign segmentation [Renz et al. 2021b] and temporal alignment of glosses [Koller et al. 2017; K. L. Cheng et al. 2020] to the best of our knowledge, this is the first use of pseudo-labelling for sign spotting by directly leveraging the predictions of a sign classifier in combination with a pseudo-label filter constructed from the subtitles themselves.

Sign Language Recognition. Efforts to develop visual systems for sign recog-

nition stretch back to work in 1988 from Tamaura and Kawasaki [Tamura and Kawasaki 1988], who sought to classify signs from hand location and motion features. There were later efforts to design hand-crafted features for sign recognition [Charayaphan and Marble 1992; Starner 1995; Vogler and Metaxas 1997; Vogler and Metaxas 1998; E. Ong et al. 2012]. Deep convolutional neural networks then came to dominate sign representation [Koller et al. 2016], particularly via 3D convolutional architectures [Joze and Koller 2019; D. Li et al. 2019; Albanie et al. 2020; D. Li et al. 2020b] with extensions to focus model capacity around human skeletons [Huang et al. 2018a] and non-manual features [Hu et al. 2021b].

In the domain of continuous sign language recognition, in which the objective is to infer a sequence of sign glosses, prior work has explored HMMs [Bauer and Hienz 2000; Koller et al. 2015a] in combination with Dynamic Time Warping (DTW) [Jihai Zhang et al. 2014], RNNs [R. Cui et al. 2017] and architectures capable of learning effectively from CTC losses [H. Zhou et al. 2020b; K. L. Cheng et al. 2020]. Recently, sign representation learning methods inspired by BERT [Devlin et al. 2018] have shown the potential to learn effective representations for both isolated [Hu et al. 2021a] and continuous [Zhenxing Zhou et al. 2021] recognition. Koller [Koller 2020] provides an extensive survey of the sign recognition literature, highlighting the extremely limited supply of datasets with large-scale vocabularies suitable for continuous sign language recognition. In our work, we aim to take a step towards addressing this gap by developing "densification" techniques for constructing such datasets automatically.

Sign Language Translation. The task of translating sign language video to spoken language sentences was first tackled with neural machine translation by Camgöz et al. [N. C. Camgoz et al. 2018], who also introduced the PHOENIX-Weather-2014T dataset to facilitate research on this topic. Several frameworks have been proposed to employ transformers for this task [N. C. Camgoz et al. 2020b; K. Yin and Read 2020], with extensions to improve temporal modelling [D. Li et al. 2020a], multi-channel cues [N. C. Camgoz et al. 2020a] and signer independence [Jin and Z. Zhao 2021]. Related work has also sought to contribute to progress on this task by exploiting monolingual data [H. Zhou et al. 2020a] and gloss sequence synthesis [Moryossef et al. 2021; D. Li et al. 2021]. To date, various works have shown promise on the PHOENIX-Weather 2014T [N. C. Camgoz et al. 2018] and CSL Daily [H. Zhou et al. 2020a] benchmarks. However, sign language

translation has not yet been demonstrated for a large vocabulary across multiple domains of discourse. Differently from the works above, this paper focuses on developing methods that are applicable to large/open vocabulary regimes.

Weakly-supervised Object Discovery and Localisation. Our approach is also related to the rich body of literature on object cosegmentation [Rother et al. 2006; Joulin et al. 2010; G. Kim et al. 2011; Rubinstein et al. 2013], weakly supervised object localisation [Nguyen et al. 2009; Deselaers et al. 2010; Z. Shi et al. 2013; Chong Wang et al. 2014; Gokberk Cinbis et al. 2014], object colocalisation [K. Tang et al. 2014; Joulin et al. 2014] and unsupervised object discovery and localisation [M. Cho et al. 2015; V. H. Vo et al. 2021]. Here, we propose an algorithm for discovering and localising novel signs (i.e. for which we have no labelled examples), but instead have weak supervision in the form of subtitles containing keywords of interest. Moving beyond initial work that sought to learn from subtitles in an aligned setting [Farhadi and Forsyth 2006], classical approaches for sign discovery using subtitles have included Multiple Instance Learning where the subtitles are considered as positive and negative bags for a particular keyword [Buehler et al. 2009; Kelly et al. 2010; Pfister et al. 2013] and a priori mining [Cooper and Bowden 2009]. Differently from these works, we first bootstrap our sign discovery process with sign spotting to both obtain initial candidates and learn robust sign representations, then propagate these examples across video data by leveraging the similarities between the resulting representations together with noisy constraints imposed by the subtitle content.

6.3 Densification

Our goal is to leverage several ways of sign spotting to achieve dense annotation on continuous signing data. To this end, we introduce both new sources of automatic annotations, and also improve the existing sign spotting techniques. We start by presenting two new spotting methods using in-domain exemplars: to mine more sign instances with individual *exemplar signs* (Sec. 6.3.1) and to discover novel signs with weak *exemplar subtitles* (Sec. 6.3.2). We also show the value of pseudo-labelling from a sign recognition model for sign spotting (Sec. 6.3.3). We then describe key improvements to previous work which substantially increase the yield of automatic annotations (Sec. 6.3.4). Finally, we present our evaluation

framework to measure the quality of our sign spottings in a large-vocabulary setting (Sec. 6.3.5). The contributions of each source of annotation are assessed in the experimental results.

6.3.1 Mining more Spottings through In-domain Exemplars (E)

The key idea is: given a continuous signing video clip and a set of exemplar clips of a particular sign, we can use the exemplars to search for that sign within the video clip. In our case, the exemplars are obtained from other *automatic* spotting methods (M*, D*, A, P), described in Sec. 6.3.3 and Sec. 6.3.4, and come from the same *domain* of sign language interpreted data, i.e. the same training set. We hypothesise that signs from the same domain are more likely to be signed in a similar way and in turn help recognition; in contrast, for example, to signs from a different domain such as dictionaries.

Formally, suppose we have a reference video V_0 in which we wish to localise a particular sign w, whose corresponding word occurs in the subtitle. We also have N video exemplars $V_1, \ldots, V_N$ of the sign w. For each video, V_i, let $\mathcal{C}_i$ denote the set of possible temporal locations of the sign w and let $c = (f, p) \in \mathcal{C}_i$ denote a candidate with features f at temporal location p. We compute a score map between our reference video V_0 and each exemplar $V_1, \ldots, V_N$ by computing the cosine similarity between each feature at each position in $c_0 \in \mathcal{C}_0$ and $(c_1, c_2, \ldots c_n) \in \mathcal{C}_1 \times \cdots \times \mathcal{C}_N$. This results in N score maps of dimension $|\mathcal{C}_0| \times |\mathcal{C}_i|$ for $i = 1 \ldots N$. We then apply a max operation over the temporal dimension of the exemplars, giving us N vectors of length $|\mathcal{C}_0|$, which we call $M_1, \ldots, M_N$.

We subsequently apply a voting scheme to find the location of the common sign w in V_0. Specifically, we let $L = \frac{1}{N} \sum_{i=1}^{N} \mathbb{1}_{(M_i > h)}$ for a threshold h, where the vector $\mathbb{1}_{(M_i > h)}$ takes the value 1 for entries of M_i which are greater than h and 0 otherwise. The candidate location of w in V_0 is then $c = (f, p) \in \mathcal{C}_0$ where p corresponds to the position of the maximum non-zero entry in the vector L (see Fig. 6.3 for a visual illustration). If there are multiple maxima, we assign p to be the midpoint of the largest connected component. If all entries in L are zero, we conclude w is not present. We perform two variants of this approach using mean and max pooling of the score maps (instead of voting); these are described in the

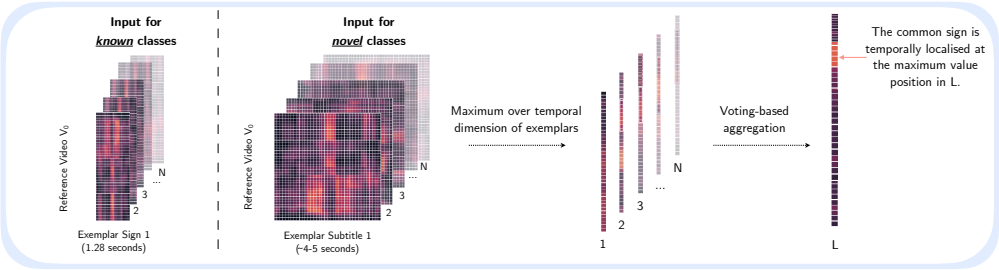


Figure 6.3: **Sign spotting through exemplars to find instances of known classes (E) and novel classes (N)**: By comparing a reference video V_0 to a set of exemplars (either sign exemplars for known sign class instances or weak subtitle exemplars for novel sign class instances), we can find the common lexical sign in the collection. We (1) form a set of score maps by calculating the cosine similarities between reference and exemplar representations; (2) we perform a maximum operation over the temporal dimension of exemplars; (3) we apply a voting-based aggregation to find the temporal location of the commmon sign in V_0. The duration of exemplar signs is fixed.

appendix. We note that for a given signing sequence, we only focus on finding signs for words in the subtitle that have *not* been annotated by other methods.

6.3.2 Discovering Novel Sign Classes (N)

One limitation of our proposed method in Sec. 6.3.1 is that we are only able to collect more sign instances from a *closed* vocabulary, determined by sign exemplars obtained from other methods (described in Sec. 6.3.3 and Sec. 6.3.4). Here, we extend our approach to localise *novel* signs, for which we have no exemplar signs but whose corresponding word appears in the subtitle text. We follow our approach described in Sec. 6.3.1, computing score maps between our reference video and exemplar subtitles (instead of exemplar signs, see Fig. 6.3). We note that by 'exemplar subtitle', we are referring to the video frames corresponding to the subtitle timestamps. Non-lexical signs, such as pointing signs or pause gestures, are very common in sign language. To avoid annotating such non-lexical signs as the common sign across V_0 and $V_1, \ldots, V_N$, we also choose N^- negative subtitle exemplars $U_1 \ldots U_{N^-}$ presumed to not contain w (due to the absence of w in the subtitle). We compute L^+ and L^- using the score maps from positive exemplars $V_1, \ldots, V_N$ and negative exemplars $U_1, \ldots, U_{N^-}$ respectively. We then let $L = L^+ - L^-$. Implementation details on the number of positive and negative exemplars used can be found in the appendix.

6.3.3 Pseudo-labelling as a Form of Sign Spotting (P)

We propose to re-purpose a pretrained large-vocabulary sign classification model (see vocabulary expansion in Sec. 6.3.5) for the task of sign spotting. Specifically, we predict a sign class from a fixed vocabulary for each time step in a continuous signing video clip. We subsequently filter the predicted signs to words which occur in the corresponding English subtitle. Similarly to [Varol et al. 2021], here the task is to recognise the sign from scratch, without a query keyword. The subtitle is only used as a post-processing step to filter out signs which are less likely performed (due to absence in the subtitle).

6.3.4 Improving the Old (M*, D*)

Here, we briefly describe our improvements over the existing sign spotting techniques, additional details are provided in the appendix.

Better Mouthings with an Upgraded KWS from Transpotter [Prajwal et al. 2021]. In previous work [Albanie et al. 2021b], an improved BiLSTM-based visual-only keyword spotting model of Stafylakis et al. [Stafylakis and Tzimiropoulos 2018] from [Momeni et al. 2020a] (named "P2G [Stafylakis and Tzimiropoulos 2018] baseline") is used to automatically annotate signs via mouthings. In this work, we make use of the recently proposed transformer-based *Transpotter* architecture [Prajwal et al. 2021], provided by the authors, that achieves state-of-the-art results in visual keyword spotting on lipreading datasets. We follow the procedure described in [Albanie et al. 2020; Albanie et al. 2021b] to query words in the subtitle in continuous signing video clips.

Finetuning KWS on Sign Language Data through Bootstrapping. The visual keyword spotting Transpotter architecture in [Prajwal et al. 2021] is trained on silent speech segments, which differ considerably from signer mouthings. In fact, signers do not mouth continuously and sometimes only partly mouth words [Boyes Braem and RL Sutton-Spence 2001]. In order to reduce this severe domain gap, we propose a dual-stage finetuning strategy. First, we extract high-confidence mouthing annotations using the pre-trained Transpotter from [Prajwal et al. 2021] on the BOBSL training data. We query for the words in the subtitle and obtain the temporal localization of the word in the video. We finetune on this pseudo-labeled data using the same training pipeline of [Prajwal et al. 2021], where the spotted

mouthings (word-video pairs) act as positive samples. For the negative samples, we pair a given word with a randomly sampled video segment from the dataset. As we observe the Transpotter to predict a large number of false positives, we remedy this by sampling a larger number of negative pairs in each batch. We also do a second round of fine-tuning by training on the pseudo-labels from the finetuned model of the first stage. We did not achieve significant improvements with further iterations.

Better Search Window with Subtitle Alignment with SAT [Bull et al. 2021]. One challenge in using sign language interpreted TV broadcasts is that the original subtitles are not aligned to the signing, but to the audio track. In [Albanie et al. 2021b], a signing query window is defined as the audio-aligned subtitle timings together with padding on both sides to account for the misalignment. We automatically align spoken language text subtitles to the signing video by using the SAT model introduced in [Bull et al. 2021], trained on manually aligned and pseudo-labelled subtitles as described in [Albanie et al. 2021b]. By using subtitles which are better aligned to the signing, we reduce the probability of missing spottings.

Better Keywords with Synonyms and Similar Words. To determine whether a keyword belongs to a subtitle, previous works [Albanie et al. 2021b] check whether the raw form, the lemmatised form, or the text normalised form (e.g. *two* instead of *2*) appears in the subtitle text. We notice that this is sub-optimal as multiple words may correspond to the same sign, often due to (i) English synonyms, (ii) identical signs for similar words, or (iii) ambiguities in spoken language. For example, *dad* and *father* or *today* and *now* can be the same signs in BSL. In this work, we investigate whether the automatic annotation yield could be improved by querying words beyond the subtitle, by querying synonyms and similar words to the words in the subtitle. We collect the additional words to query through (i) English synonyms from WordNet [Feinerer and Hornik 2020], (ii) the metadata present in online sign language dictionaries such as SignBSL[1] [Momeni et al. 2020b] and BSL Sign-Bank[2] which provide a set of 'related words' for each sign video entry; (iii) words with GloVe [Pennington et al. 2014] cosine similarity above 0.9 to account for ambiguities in spoken language.

[1] www.signbsl.com
[2] bslsignbank.ucl.ac.uk

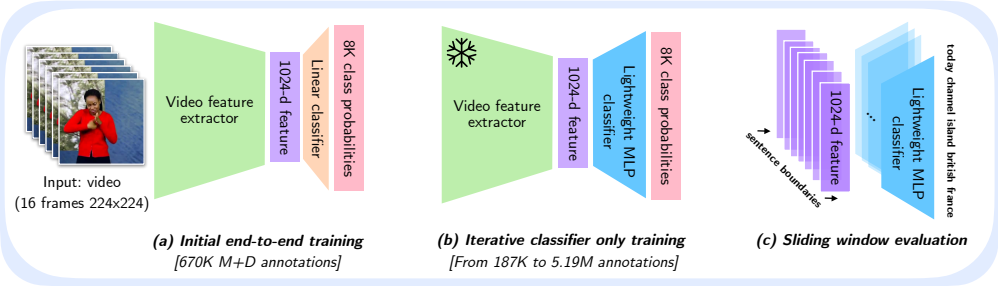


Figure 6.4: **Evaluation framework**: (a) Video features are obtained by training an I3D architecture end-to-end given M + D annotations from [Albanie et al. 2021b]. The I3D ingests 16-frames of video and has a linear classifier for 8K sign categories. The end-to-end training is a costly procedure which is not affordable to repeat for each set of our new sign spottings that are on the order of several million training samples. (b) As new sets of spottings are generated, a light weight MLP classifier is trained on the pre-extracted I3D features. This relatively inexpensive training procedure means that we benefit from new annotations without the expense of end-to-end training. (c) The MLP is applied in a sliding window fashion to the signing sequence to generate sign predictions.

6.3.5 Evaluation Framework

Our framework consists of three stages: (a) a costly end-to-end classification training to learn sign category aware video features given an initial set of sign-clip annotation pairs; (b) a lightweight classification training given pre-extracted video features for a large number of annotations; (c) a sliding window evaluation of the trained lightweight model by comparing dense sign predictions against the subtitles (see Sec. 6.4.1). These stages are illustrated in Fig. 6.4. Note that the *annotations* we refer to are always *automatically* localised sign spottings from continuous videos using subtitle information. The motivation for the video backbone and lightweight classifier is purely related to computational costs. Unlike traditional video recognition datasets, we work with untrimmed video data of 1400 hours, where the set of sign-clip pairs is not fixed. Instead, our goal is to increase the number of sign-clip pairs within the continuous stream, and assess the quality of the expanded annotation yield on the proxy task of continuous sign language recognition. Next, we describe the training stages for the video backbone and the lightweight classifier.

Improving the I3D Feature Extractor through Vocabulary Expansion. Following previous works [Joze and Koller 2019; D. Li et al. 2019; Albanie et al. 2020; Albanie et al. 2021b], we use the I3D spatio-temporal convolutional architecture to train an end-to-end sign recognition model. We input 16 consecutive

RGB frames and output class probabilities. The details about optimisation are provided in the appendix. As explained above, this model forms the basis of sign video representation which corresponds to the spatio-temporally pooled latent embedding before the classification layer. The prior work of [Albanie et al. 2021b] trains this classifier on the BOBSL dataset (see Sec. 6.4.1) with 2K categories obtained through the vocabulary of mouthing spottings. As a first step, we perform a vocabulary expansion and construct a significantly increased vocabulary of 8K categories. This is achieved by including each sign that has at least 5 training spottings above 0.7 confidence from both mouthing (M) and dictionary (D) annotations. The confidence for the mouthing annotation corresponds to the probability that a text keyword (corresponding to the sign) is mouthed at a certain time frame, as computed in [Albanie et al. 2020]. The confidence for the dictionary annotation corresponds to the cosine similarity (normalised between 0-1) between the representations of a dictionary clip of the sign and the continuous signing at each time frame, as in [Momeni et al. 2020b]. The resulting M+D training set comprises 670K annotations, with a long-tailed distribution. Furthermore, we note that the categories are noisy where multiple categories may correspond to the same sign, and vice versa. Despite this noise, we empirically show that this model provides better performance than its 2K-vocabulary counterpart. We use our improved I3D model for two purposes: as the frozen feature extractor and as the source of pseudo-labelling for sign spotting (see Sec. 6.3.3).

Lightweight Sign Recognition Model. Following [Varol et al. 2021], we opt for a 4-layer MLP module (with one residual connection) to assess the quality of different sets of annotations. Given pre-extracted features, this model is trained for sign recognition into 8K categories. We note that we do not train on a larger vocabulary to avoid the presence of many singletons in the training set. The efficiency of the MLP allows faster experimentation to analyse the value of each of our sign spotting sets. The input is one randomly sampled feature around the sign spotting location (the receptive field of one feature 16 frames). The MLP weights are randomly initialised. Additional training and implementation details are given in the appendix.

6.4 Experiments

We start by describing our dataset and evaluation metrics (Sec. 6.4.1). We then present experimental results on the contribution of each source of annotation and show qualitative examples (Sec. 6.4.2).

6.4.1 Data and Evaluation Protocol

BOBSL [Albanie et al. 2021b] is a public dataset consisting of British Sign Language interpreted BBC broadcast footage, along with English subtitles corresponding to the audio content. The data contains 1,962 episodes, which have a total duration of 1,467 hours spanning 426 different TV shows. BOBSL has a total 1,193K subtitles covering a total vocabulary of 78K words. We note that in this work we use the word *subtitle* to refer to the processed BOBSL sentences from [Albanie et al. 2021b] as opposed to the raw subtitles. There are a total of 39 signers in the dataset. Further dataset statistics can be found in [Albanie et al. 2021b]. For a subset of 36 episodes in BOBSL, referred to as SENT-TEST in [Albanie et al. 2021b], the English subtitles have been manually aligned *temporally* to the continuous signing video. We make use of this test set to evaluate the quality of our predicted automatic annotations. SENT-TEST covers a total duration of 31 hours and contains 20,870 English subtitles. The total vocabulary of English words is 13,641, of which 5,604 are singletons. The 3 signers in SENT-TEST are different to the signers in the training set, this enables signer-independent BSL recognition to be evaluated.

Evaluation protocol. Given an English subtitle and the *temporally* aligned continuous signing video clip, we evaluate our predicted signs for the clip using (i) *intersection over union* (IoU); (ii) *recall* between signs and the English word sequence; and (iii) *temporal coverage*: this is defined as the proportion of frames in the clip assigned to signs that occur in the word sequence, where a sign is given a fixed duration of 16 frames (for 25Hz video). Note that none of these metrics depend on the word order of the English subtitle, only the words it contains. All metrics are rescaled from the range 0-1 to 0-100 percentage for readability.

For this evaluation, stop words are filtered out since often they are not signed. This reduces the number of test subtitles from 20,870 to 20,547: subtitles such

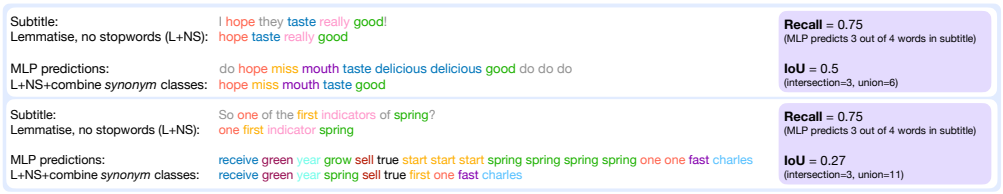


Figure 6.5: **Evaluation illustration on sample prediction:** We illustrate the processing applied to the predicted sign sequence from the MLP predictions and corresponding English subtitle for calculating our metrics. As the MLP model predicts one sign per time-step, some predictions are repeated and irrelevant words appear at transition periods between signs, decreasing the IoU. Some signs are not predicted as they are not signed, showing the limitations of using the subtitle to measure performance.

as "is it?", "Oh!", "but no" are removed. The sign and word sequences are also lemmatised. We also remove repetitions from the predicted sign sequence and allow the prediction of synonyms of words in the English subtitle. This processing is highlighted in Fig. 6.5, where the IoU and recall are computed for a pair of predicted signs and English text. While this evaluation is suboptimal due to the simplified word-sign correspondence assumption, it tests the capacity of the sign recognition model in a large-vocabulary scenario, necessary for open-vocabulary sign language technologies.

Note, the predicted signs for a clip can be produced in two ways. In the first way, the signs are obtained from the automatic annotations using knowledge of the content of the English subtitle – we refer to these as *Spottings*. In the second, signs are predicted directly from the clip using the MLP sign predictions, without access to the corresponding English subtitle. These are referred to as *MLP predictions*. Spottings are evaluated using all the words; this metric is important to monitor how dense we can automatically annotate the data. The MLP evaluation is limited to the fixed classification vocabulary (of size 8K in our experiments). We note that when different annotations are combined, the sign spotting methods are applied independently.

6.4.2 Results

Comparison of Video Features. By finetuning our Kinetics pretrained I3D model on BOBSL M+D annotations from [Albanie et al. 2021b] using an 8K vocabulary instead of a 2K vocabulary, we improve predictions on the test set, as shown in 6.1. We increase the recall from 25.5 to 26.3 and the coverage from

Table 6.1: **Comparison of I3D video features**: We highlight the improved performance of I3D on the test set (SENT-TEST) when trained on a larger vocabulary (8K instead of 2K) with more samples (670K instead of 426K).

Annot. source	Num. I3D train annot.	Vocab. size	I3D predictions (subtitle independent) Recall	IoU	Coverage
M [Albanie et al. 2020]+D [Momeni et al. 2020b]	426K	2K	25.5	6.4	15.5
M [Albanie et al. 2020]+D [Momeni et al. 2020b]	670K	8K	**26.3**	**7.9**	**16.3**

Table 6.2: **Improved mouthing and dictionary spottings:** We evaluate different sets of spottings and their respective MLP predictions. M [Prajwal et al. 2021] shows our finetuned version for all the rows in the last block. We quantify the effects of subtitle alignment and querying synonyms. We also show the oracle performance and a translation baseline.

Annotation source	Subtitle alignment	Synonyms	Training set full vocab	#ann. [full]	#ann. [8K]	Spottings [full] (subtitle dependent) Recall	IoU	Coverage	MLAP predictions [8K] (subtitle independent) Recall	IoU	Coverage
Oracle			-	-	-	-	-	-	86.7	86.3	55.2
Translation baseline [Albanie et al. 2021b]			-	-	-	-	-	-	11.7	8.3	7.6
M [Albanie et al. 2020]			13.6K	197K	187K	2.5	2.2	1.3	15.1	3.2	8.7
M [Prajwal et al. 2021] (no finetuning)			21.5K	725K	661K	9.4	8.3	4.9	20.4	4.8	11.9
M [Prajwal et al. 2021]			18.6K	445K	412K	7.1	6.5	3.9	23.6	4.8	13.8
M [Prajwal et al. 2021] (M*)	✓		19.6K	598K	552K	8.9	8.2	4.9	27.4	6.3	16.7
M [Prajwal et al. 2021]	✓	✓	19.6K	1.38M	1.25M	11.8	10.4	6.1	25.3	6.2	16.3
D [Momeni et al. 2020b]			4.4K	482K	482K	6.5	6.3	3.7	24.0	7.2	15.1
D [Momeni et al. 2020b]	✓		4.5K	535K	535K	7.0	6.9	4.0	24.2	7.3	15.3
D [Momeni et al. 2020b] (D*)	✓	✓	5.0K	1.40M	1.39M	12.5	11.6	7.0	26.0	7.3	16.9
M*+ D*	✓	✓(D-only)	20.9K	2.00M	1.94M	19.0	17.6	10.5	29.0	7.9	18.4
M*+ D*+ A [Varol et al. 2021]	✓	✓(D-only)	20.9K	2.43M	2.37M	**21.9**	**20.1**	**11.8**	**29.6**	**9.1**	**19.0**

15.5 to 16.3. We therefore use the 8K M+D model for the rest our experiments as the frozen feature extractor. We note that we restrict the M+D annotations to the high-confidence ones (over 0.8 threshold) used for the I3D baseline in [Albanie et al. 2021b], as these present an appropriate signal-to-noise ratio. We use the same threshold for subsequent automatic annotations unless stated otherwise.

Oracle. As the MLPs are trained on a restricted 8K vocabulary, it is not possible to predict the full vocabulary of 13,641 words present in the test set subtitles. Furthermore, not all words in the subtitle are signed and vice versa. This means a recall, IoU and coverage of 100% is not achievable between predicted signs and English subtitle words. However, we propose an oracle in Tab. 6.2 whereby we measure the recall and IoU assuming each word in the subtitle, which either falls within the 8K vocabulary or corresponds to a synonym of a word in the 8K vocabulary, is signed and correctly predicted. The oracle achieves a recall of 86.7 and IoU of 86.3. For the coverage metric, we assume each correctly predicted sign has a duration of 16 frames and no signs overlap. The resulting oracle coverage is 55.2. This low coverage is partly due to the signer pausing within subtitles and also due to the presence of non-lexical signs. In fact, the percentage of fully lexical

signs in three other sign language corpora (Auslan [T. Johnston 2012], ASL [T. Johnston 2012] and LSF [Belissen et al. 2020]) is estimated to be only 70-85% of total signing.

Translation Baseline. Although the goal in this work is not translation, but achieving dense annotations, we can nevertheless compare our MLP predictions to the translation baseline in [Albanie et al. 2021b]. Using the test set translation predictions from this model, we perform the same processing as highlighted in Fig. 6.5 to calculate our metrics. As shown in Tab. 6.2, all our simple MLP models clearly outperform the transformer-based translation model used in [Albanie et al. 2021b], demonstrating that we are able to recognise more signs in the English subtitle.

Improving Mouthing and Dictionary Spottings. As shown in Tab. 6.2, by using the Transpotter [Prajwal et al. 2021] for spotting mouthings M, our yield of total annotations triples from 197K to 725K. The quality of these new annotations is reflected in the increased performance of the MLP: the recall increases from 15.1 to 20.4 and the coverage from 8.7 to 11.9. Finetuning the keyword spotter on sign language data through pseudo-labelling also helps considerably despite the drop in the number of training annotations since there are less false positives; recall increases from 20.4 to 23.6 and coverage from 11.9 to 13.8. Subtitle alignment improves the yield of both mouthing and dictionary annotations, as shown in Tab. 6.2. This translates to a significant boost for mouthings on the MLP performance; the recall increases from 23.6 to 27.4 and the coverage from 13.8 to 16.7. For dictionary annotations, the improvement by using aligned subtitles is less striking. By querying synonyms when searching for mouthings, the yield more than doubles. However, these additional annotations seem to be quite noisy as they decrease the performance of our MLP. Due to the nature of sign language interpretation, it is possible that signers are far more likely to mouth a word which is actually in the written subtitle than a synonym of that word. We therefore do not query synonyms for mouthing spottings. For dictionary spottings, we observe the opposite effect. By incorporating synonyms, the yield of dictionary spottings more than doubles and the recall of the MLP predictions also increases from 24.2 to 26.0. We denote our best performing mouthing and dictionary spottings with M^* and D^*, respectively. Adding attention spottings from [Albanie et al. 2021b] (with a threshold of 0) adds around 400K additional annotations and boosts the

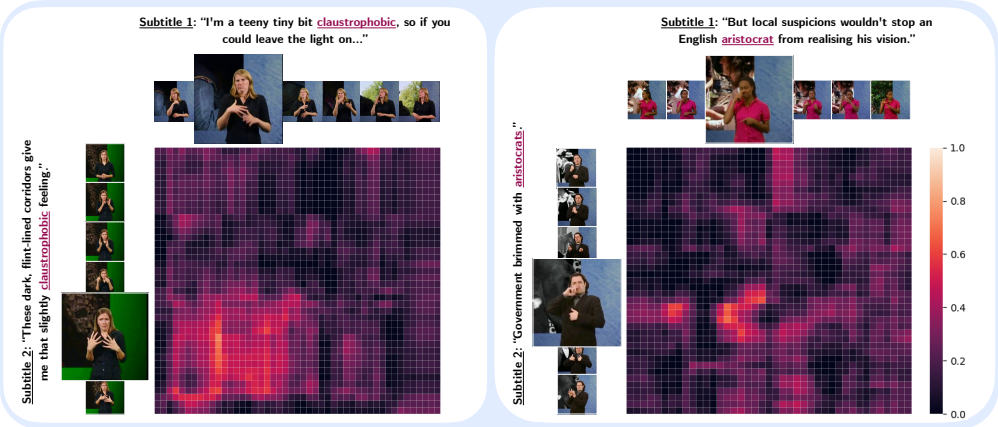


Figure 6.6: **Discovering novel sign classes (N)**: For two pairs of continuous signing sentences, we plot the score maps (as described in Sec. 6.3.1) between their feature sequences. We highlight the ability of our approach to spot novel sign classes.

MLP performance; increasing recall from 29.0 to 29.6 and coverage from 18.4 to 19.0, compared to the oracle recall of 86.7 and coverage of 55.2.

Sign Recognition as a Form of Pseudo-labelling. Pseudo-labels P are a source of over 1M new annotations (when using a threshold of 0.5) on top of our best M*, D*, A spottings. As shown in Tab. 6.4, they greatly increase the spottings recall from 21.9 to 25.4 and coverage from 11.8 to 13.9, while only marginally increasing the recall and coverage for MLP predictions. As the pseudo-labels come from our 8K I3D model in Tab. 6.1 whose frozen features are also used for training the MLP, P may not be providing additional information for our downstream evaluation. Nevertheless, they provide a great source of additional spottings (not found by previous methods) for our goal of dense annotation.

Mining more Examples of Known and Novel Sign Classes with In-domain Exemplars. By explicitly querying words in the subtitle text which are not present in our annotations, we can obtain significantly more annotations. Tab. 6.3 shows multiple methods to use exemplar signs to find additional annotations for these signs. The best performing method takes spotting exemplars from across the whole training set, irrespective of signer or episode, and uses the voting scheme described in Sec. 6.3.1 to localise signs. By using 20 spotting exemplars, we acquire 1.63M additional annotations. An MLP model trained *only* on these additional annotations achieves a recall of 28.4 and coverage of 18.1. Tab. 6.4 illustrates the impact of combining these additional annotations from spotting exemplars to M*, D*, A and P annotations. With the additional exemplar-based

Table 6.3: **Ablation on mining exemplar-based spottings for known signs E:** We perform different ablations for mining known signs which have been unannotated by previous methods (M*, D*, A, P). We experiment with the source of exemplar data (same episode, same signer, all data), the confidence of exemplar signs (0,0.5,0.8), the number of samples of exemplar data (5,10,20) and the pooling mechanism (average, max, vote). We evaluate on the test set (SENT-TEST).

					Training set			**Spottings [full] (subtitle dependent)**			**MLP predictions [8K] (subtitle independent)**		
Ann. src.	ex. data	ex. thres	ex. #	ex. pooling	full vocab	#ann. [full]	#ann. [8K]	Recall	IoU	Coverage	Recall	IoU	Coverage
E	same ep.	0	var	avg	11.6K	869K	833K	10.4	9.6	5.8	25.1	6.9	15.3
E	same signer	0	20	avg	15.9K	505K	421K	7.8	7.5	4.4	23.1	5.6	14.2
E	all	0	20	avg	16.7K	351K	252K	5.7	5.7	3.3	21.5	5.1	13.4
E	all	0.5	20	avg	16.6K	370K	261K	5.9	5.8	3.4	21.9	5.2	13.5
E	all	0.8	20	avg	16.6K	458K	358K	7.4	7.3	4.3	25.2	6.2	15.7
E	all	0.8	20	max	15.4K	1.48M	1.38M	20.2	18.6	10.8	27.6	8.4	17.7
E	all	0.8	10	max	15.4K	1.07M	982K	15.2	14.0	8.3	27.9	8.0	17.7
E	all	0.8	5	max	15.3K	740K	664K	10.7	10.0	6.0	27.6	7.6	17.4
E	all	0.8	20	vote	15.9K	1.76M	1.63M	**25.8**	**23.3**	**13.5**	**28.4**	**8.5**	**18.1**
E	all	0.8	10	vote	15.8K	1.32M	1.21M	20.0	18.1	10.7	28.4	8.3	18.1

Table 6.4: **Pseudo-label spottings P & Exemplar-based sign spottings for known E and novel classes N:** We highlight the boost in annotations by adding our pseudo-label annotations (P) as well as exemplar-based spottings of known (E) and novel (N) classes. We evaluate Spottings and MLP predictions on the test set (SENT-TEST). For the novel classes, we only show the evaluation of spottings since these are beyond the 8K training vocabulary of the MLP.

	Training set			**Spottings [full] (subtitle dependent)**			**MLP predictions [8K] (subtitle independent)**		
Annotation source	full vocab	#ann. [full]	#ann. [8K]	Recall	IoU	Coverage	Recall	IoU	Coverage
M* + D* + A [Varol et al. 2021] + P	20.9K	3.64M	3.56M	25.4	23.5	13.9	29.8	8.9	19.2
M* + D* + A [Varol et al. 2021] + P + E	20.9K	5.40M	5.19M	45.3	40.7	23.3	**30.7**	**9.5**	**19.8**
M* + D* + A [Varol et al. 2021] + P + E + N	24.8K	5.47M	-	**45.6**	**40.9**	**23.4**	-	-	-

annotations E, recall increases from 29.8 to 30.7 and coverage increases from 19.2 to 19.8, where the oracle recall and coverage are 86.7 and 55.2. Furthermore, by mining instances of novel sign classes N (see Fig. 6.6), we increase our total vocabulary to 24.8K and total number of annotations to 5.47M.

6.5 Conclusion

Progress in sign language research has been accelerated in recent years due to the availability of large-scale datasets, in particular sourced from interpreted TV broadcasts. However, a major obstacle for the use of such data is the lack of available sign level annotations. Previous methods [Albanie et al. 2020; Momeni et al. 2020b; Varol et al. 2021] only found *sparse* correspondences between keywords in the subtitle and individual signs. In our work, we propose a framework which

scales the number of confident automatic annotations from 670K to 5.47M (which we make publicly available). Potential future directions for research include: (1) increasing our number of annotations by incorporating context from *surrounding* signing to resolve ambiguities; (2) investigating *linguistic* differences between spoken English and British Sign Language such as the different word/sign ordering; (3) leveraging our automatic annotations for sign language translation.

Acknowledgements. This work was supported by EPSRC grant ExTol, a Royal Society Research Professorship and the ANR project CorVis ANR-21-CE23-0003-01. LM would like to thank Sagar Vaze for helpful discussions. HB would like to thank Annelies Braffort and Michèle Gouiffès for the support.

Chapter 7

Weakly-supervised Fingerspelling Recognition in British Sign Language Videos

The paper has been accepted for publication at the British Machine Vision Conference (BMVC), 2022.

Weakly-supervised Fingerspelling Recognition in British Sign Language Videos

K R Prajwal[1*] Hannah Bull[2*] Liliane Momeni[1*]

Samuel Albanie[3] Gül Varol[2] Andrew Zisserman[1]

[1] Visual Geometry Group, University of Oxford, UK

[2] LIGM, École des Ponts, Univ Gustave Eiffel, CNRS, France

[3] Department of Engineering, University of Cambridge, UK

Abstract

The goal of this work is to detect and recognize sequences of letters signed using fingerspelling in British Sign Language (BSL). Previous fingerspelling recognition methods have not focused on BSL, which has a very different signing alphabet (e.g., two-handed instead of one-handed) to American Sign Language (ASL). They also use manual annotations for training. In contrast to previous methods, our method only uses weak annotations from subtitles for training. We localize potential instances of fingerspelling using a simple feature similarity method, then automatically annotate these instances by querying subtitle words and searching for corresponding mouthing cues from the signer. We propose a Transformer architecture adapted to this task, with a multiple-hypothesis CTC loss function to learn from alternative annotation possibilities. We employ a multi-stage training approach, where we make use of an initial version of our trained model to extend and enhance our training data before re-training again to achieve better performance. Through extensive evaluations, we verify our method for automatic annotation and our model architecture. Moreover, we provide a human expert annotated test set of $5K$ video clips for evaluating BSL fingerspelling recognition methods to support sign language research.

*Equal contribution.

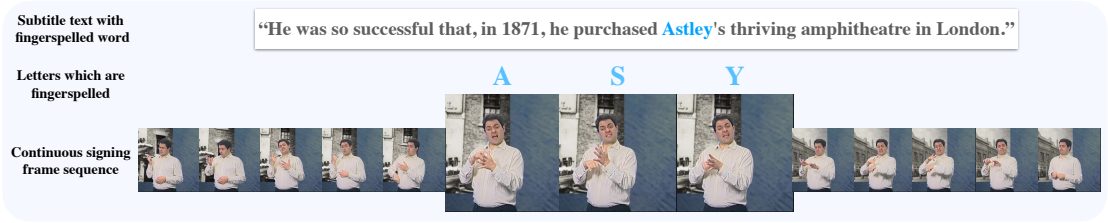


Figure 7.1: **Fingerspelling recognition.** We study the task of recognising a sequence of BSL fingerspelled letters in a continuous signing window in sign language interpreted TV broadcast data. We exploit the accompanying English subtitles to automatically collect training data. However, the task remains very challenging as (i) the signer may fingerspell only a subset of the letters – as shown above, although the subtitle contains the word 'Astley', only the letters 'A', 'S' and 'Y' are fingerspelled – and (ii) due to occlusions from the hands, different letters can visually look very similar.

7.1 Introduction

Fingerspelling in signed languages is a means to encode words from the surrounding written language into sign language via a manual alphabet, i.e. one sign per letter. Words from a written language with no known sign may be fingerspelled, such as names of people and places. Additionally, some signs are derived from fingerspelled words, for example, initialized signs in BSL such as 'F' for 'father' or 'KK' for 'kitchen' [Rachel Sutton-Spence and Woll 1999a]. Padden & Gunsauls [Padden and Gunsauls 2003] estimate that signers fingerspell 12-35% of the time in ASL. Within the BSL videos used in this work [Albanie et al. 2021b], we estimate roughly 5-10% based on the duration of automatically detected fingerspelling content. Consequently, it is important to incorporate automatic fingerspelling recognition methods to be able to *exhaustively* transcribe signs in continuous sign language videos.

Some sign languages, such as ASL and LSF, use a one-handed manual alphabet, and others, including BSL and Auslan, use a two-handed manual alphabet [Schembri and T. Johnston 2007]. One-handed manual alphabets typically do not involve significant wrist movements, while two-handed fingerspelling resembles other lexical signs, making it relatively more difficult to detect fingerspelling beginning and end times from a longer signing sequence. The presence of two-hand movements further results in occlusions, making it challenging to differentiate between certain characters, e.g. especially vowels.

In this work, we focus on BSL, i.e. the more challenging two-handed fingerspelling. Given a sign language video, our goal is to temporally locate the fin-

gerspelling segments within the video (i.e. detection) and to transcribe the fingerspelled letters to text (i.e. recognition). To this end, we design a Transformer-based model that jointly performs both detection and recognition. Our key contribution lies in the data collection procedure used to automatically obtain training data for this task, which is applicable to any sign language videos that have approximately-aligned subtitle translations. We also provide the first large-scale benchmark for BSL fingerspelling recognition based on the recently released BOBSL dataset [Albanie et al. 2021b]. Our experiments on this benchmark demonstrate promising results with a 53.3 character error rate on this challenging task while only using weak supervision.

Previous works building fingerspelling datasets rely on manual annotation, either by expert annotators resulting in limited data [B. Shi et al. 2018], or by crowdsourcing noisy large-scale annotations [B. Shi et al. 2019]. In contrast to these works on ASL fingerspelling, we introduce a practical methodology to automatically annotate fingerspelling in the presence of subtitled sign language video data, allowing to scale up the data size, and potentially to be applicable to other sign languages. Starting from a small number of manually annotated fingerspelling exemplars, we use an embedding space to find numerous similar instances of fingerspelling in the corpus. To annotate these instances, we exploit the observation that signers often simultaneously mouth the words (i.e. silent speech with lip movements) which they fingerspell. We obtain an initial set of annotations by querying potential words from the subtitles, especially proper nouns, and identifying mouthing cues [Albanie et al. 2020; Prajwal et al. 2021] which also coincide with fingerspelling instances. This initial set of annotations are further extended and enhanced by a pseudolabeling step, also making use of subtitles (see Sec. 7.4).

A key challenge with training through automatic annotations is label noise. For example, the mouthing model [Prajwal et al. 2021] may spot a single word, while the fingerspelling contains multiple words, such as name-surname pairs. In some other cases, the mouthing model may fail, associating the wrong word in the subtitle with the fingerspelling segment. Also, some letters of the word may be skipped within fingerspelling as illustrated in Fig. 7.1. To account for this uncertainty, we implement a multiple-hypotheses version of the CTC loss [Graves et al. 2006] (MH-CTC) where we consider all nouns from the subtitle, as well as bigrams and trigrams, as potential targets.

Our main contributions are: (i) Training a BSL fingerspelling detection and recognition model using only weak labels from subtitles, mouthing cues, and a small number (115) of manual fingerspelling exemplars; (ii) Employing multiple hypotheses from the subtitle words within the CTC loss to train with noisy labels; (iii) Demonstrating advantages of a pseudolabeling step incorporating the subtitle information; and (iv) Providing a large-scale manually annotated benchmark for evaluating BSL fingerspelling recognition, released for research purposes. Please check our website for more details: https://www.robots.ox.ac.uk/~vgg/research/transpeller.

7.2 Related work

Our work relates to four themes in the research literature: the broader topics of *sign language recognition* and *learning sign language from weak/noisy annotation*, and the more directly related literature on *spotting mouthings* and *fingerspelling recognition*.

Sign language recognition. Building on the pioneering 1988 work of Tamura and Kawasaki [Tamura and Kawasaki 1988], early approaches to automatic sign recognition made use of hand-crafted features for motion [M.-H. Yang et al. 2002] and hand shape [Fillbrandt et al. 2003; Vogler and Metaxas 2003]. To model the temporal nature of signing, there has also been a rich body of work exploring the use of Hidden Markov Models [Starner 1995; Vogler and Metaxas 2001; Fang et al. 2004; Cooper et al. 2011; Koller et al. 2016; Koller et al. 2017] and Transformers [N. C. Camgoz et al. 2020b; De Coster et al. 2020]. One notable trend in prior work is the transition towards employing deep spatiotemporal neural networks to provide robust features for recognition. In this regard, the I3D model of [J. Carreira and A. Zisserman 2017] has seen widespread adoption, achieving strong recognition results on a range of benchmarks [Joze and Koller 2019; D. Li et al. 2019; Albanie et al. 2020; D. Li et al. 2020b]. In this work, we likewise build our approach on strong spatiotemporal video representations, adopting the Video-Swin Transformer [Z. Liu et al. 2022] as a backbone for our model.

Learning sign language from weak/noisy annotation. Given the paucity of large-scale annotated sign language datasets, a range of prior work has sought to leverage weakly aligned subtitled interpreter footage as a supervisory signal [Cooper

and Bowden 2009; Buehler et al. 2009; Pfister et al. 2014; Momeni et al. 2020b] for sign spotting and recognition via apriori mining [Agrawal et al. 1993] and multiple instance learning [Dietterich et al. 1997]. Similarly to these works, we likewise aim to make use of subtitled signing footage. However, we do so in order to detect and recognize fingerspelling in a manner that allows for scalable training. To the best of our knowledge, this approach has not been considered in prior work.

Spotting mouthings in sign language videos. Signers often mouth the words that they sign (or fingerspell) [Rachel Sutton-Spence 2007]. The recent advancements in visual keyword spotting [Prajwal et al. 2021; Stafylakis and Tzimiropoulos 2018; Momeni et al. 2020a] have enabled the automatic curation of large-scale sign language datasets by spotting a set of query words using the mouthing cues and matching them with the corresponding sign segment. The state-of-the-art architecture for the visual KWS task is the Transpotter [Prajwal et al. 2021], which we use in Sec 7.4.1 to obtain our initial set of automatic fingerspelling annotations. Our fingerspelling architecture also partly takes inspiration from the Transpotter (and more broadly from prior works for text spotting that detect words and learn to read them via CTC [H. Li et al. 2017; Borisyuk et al. 2018]), wherein we process video features with a single Transformer encoder and then employ multiple heads to solve related tasks such as detection and classification with a $[CLS]$ token. In this work, we add also add a recognition head supervised by a novel loss function. We also show the benefits of our multi-stage training pipeline in training this model when we only have weak supervision.

Fingerspelling recognition and detection. Early work on automatic fingerspelling recognition explored the task of classification under fairly constrained settings, focusing on isolated signs and limited vocabularies (e.g. 20 words [Goh and Holden 2006], 82 words [Ricco and Tomasi 2009] and 100 words [Liwicki and Everingham 2009]). Kim et al. propose to consider instead a "lexicon-free" setting which they tackle with frame-level classifiers in combination with segmental CRFs on a newly introduced dataset, of 3,684 American Sign Language (ASL) fingerspelling instances [Taehwan Kim et al. 2017]. Moving towards more challenging data, the ChicagoFSWild (7304 fingerspelling sequences across 160 signers annotated by ASL students) [B. Shi et al. 2018] and ChicagoFSWild+ (55,232 fingerspelling sequences signed by 260 signers annotated by crowdsourcing) [B. Shi et al. 2019] datasets sourced from YouTube and Deaf social media target greater

diversity and visual variation.

From a modeling perspective, Pugealt and Bowden employ random forests on depth and intensity images for real-time recognition of 24 fingerspelled letters [Pugeault and Bowden 2011]. Shi et al. demonstrate the benefits of using a signing hand detector for fingerspelling recognition without frame-level labels [B. Shi and Livescu 2017], motivating later work to attain this benefit automatically through visual attention without an explicit region detector [B. Shi et al. 2019]. Other work has explored the feasibility of using synthetic hand training data to fine-tune a CNN for isolated Irish Sign Language (ISL) fingerspelling recognition [Fowley and Ventresque 2021].

More closer to our work, several works have considered detecting the temporal location of fingerspelling in addition to recognition. This includes efforts to segment signing into sign types (classifiers, lexical signs, and fingerspelling) prior to recognition [Yanovich et al. 2016], as well as systems for fingerspelling detection supervised with segment boundaries [B. Shi et al. 2021]. The recently proposed FSS-Net learns joint embeddings to enable fingerspelling search within and across videos [B. Shi et al. 2022b]. In contrast, our work is weakly supervised with noisy, automatic annotations. It is weakly supervised in the sense that the model lacks access to ground truth fingerspelling boundaries at training, while the annotation is noisy in the sense that it is derived from subtitles from which the signing is produced as a translation, rather than a transcription.

7.3 Fingerspelling detection and recognition

In this section, we introduce Transpeller, our Transformer-based model to recognize and detect fingerspelling (Sec. 7.3.1). Our model is trained only on automatically curated data. In order to circumvent this label noise, we also propose a new loss function in Sec. 7.3.2.

7.3.1 The Transpeller architecture

Our model ingests a video clip of a signer, encodes it with a Transformer encoder [Vaswani et al. 2017], and produces three outputs with each of its prediction heads: (i) a classification head that predicts **if** the given video clip contains fingerspelling, (ii) a localization head that produces per-frame probabilities indicating **where** the fingerspelling is in the clip, (iii) a recognition head that produces a se-

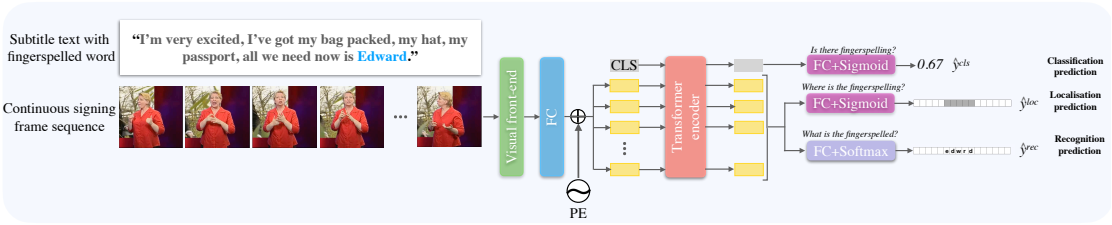


Figure 7.2: **Transpeller architecture.** Given a short clip of a signer, we extract features from a visual front-end pre-trained on the sign language recognition task. We project these features to a desired hidden dimension and add positional encodings before encoding these feature vectors using a transformer encoder. We use three heads on top of the transformer outputs to predict if the clip contains fingerspelling (classification prediction), and if so, where it is in the clip (localization prediction) and what it is (recognition prediction). The subtitle shown on the left is used for curating the training data and is shown here for illustrative purposes.

quence of letter probabilities indicating **what** is being fingerspelled. We illustrate this architecture in Figure 7.2.

Visual backbone. Our input is a sequence of RGB frames constituting a short clip of a signer. In order to extract the visual features, we follow prior sign-language works [Bull et al. 2021; Momeni et al. 2022; Varol et al. 2021] and use a pre-trained sign classification model. The sign classification model used in prior works is an I3D model [Bull et al. 2021; Momeni et al. 2022; Varol et al. 2021] pre-trained on Kinetics [Kay et al. 2017] and finetuned using a sign classification objective. We replace the I3D with a relatively modern Video-Swin-S [Z. Liu et al. 2022] architecture and finetune it for the sign classification task in a similar fashion. We pre-extract the features for all our videos and save them at a temporal stride of 4. The Transpeller operates on these feature vectors $V \in \mathbb{R}^{T \times d_f}$.

Transformer Encoder. Given a sequence of visual feature vectors $V \in \mathbb{R}^{T \times d_f}$, we first use two fully-connected layers (FC) to project the feature dimension to d, which is the hidden dimension of the transformer encoder. We add the temporal positional encodings and prepend this sequence with a learnable $[CLS]$ token embedding (such as in BERT [Devlin et al. 2018] and ViT [Dosovitskiy et al. 2021]). We encode the temporal information of this sequence by passing it through a Transformer encoder consisting of N layers:

$$V_{enc} = encoder([CLS]; [FC(V) + PE_{1:T}]) \in \mathbb{R}^{(1+T) \times d}.$$

Prediction heads. The $[CLS]$ output feature vector $V_{enc(1)}$ serves as a aggregate representation for the entire input video clip. An MLP head for binary classifica-

tion, f_c is attached to $V_{enc(1)}$ to predict the probability of a fingerspelling segment being present in the input video:

$$\hat{y}^{cls} = \sigma(f_c(V_{enc(1)})) \in \mathbb{R}^1,$$

where σ denotes a sigmoid activation. To localize the fingerspelling segment, we attach a second MLP head f_l that is shared across the encoded video feature time-steps $V_{enc(2:T+1)}$:

$$\hat{y}^{loc} = \sigma(f_l(V_{enc(2:T+1)})) \in \mathbb{R}^T.$$

The output y_t^{loc} at each feature time-step $t \in T$ indicates the probability of the time-step t being a part of a fingerspelling segment. In order to recognize what is being fingerspelt, a third MLP head f_r is attached in a similar fashion to f_l to predict $\mathbb{C}$ letter probabilities at each time-step:

$$\hat{y}^{rec} = softmax(f_r(V_{enc(2:T+1)})) \in \mathbb{R}^{T \times \mathbb{C}}.$$

Loss functions. Given a training dataset $\mathcal{D}$ consisting of video clips v, class labels y^{cls}, location labels y^{loc} and recognition labels (whenever present) y^{rec}, we define the following training objectives:

$$\mathcal{L}^{cls} = -\mathbb{E}_{(v,y^{cls})\in\mathcal{D}}\ BCE(y^{cls}, \hat{y}^{cls}) \tag{7.1}$$

$$\mathcal{L}^{loc} = -\mathbb{E}_{(v,y^{cls},y^{loc})\in\mathcal{D}}\ y^{cls}\left[\frac{1}{T}\sum_{t=1}^{T} BCE(y_t^{loc}, \hat{y}_t^{loc})\right] \tag{7.2}$$

$$\mathcal{L}^{rec} = -\mathbb{E}_{(v,y^{rec})\in\mathcal{D}}\ CTC(y^{rec}, \hat{y}^{rec}) \tag{7.3}$$

where BCE stands for the binary cross-entropy loss and CTC stands for Connectionist Temporal Classification [Graves et al. 2006] loss. The labels y^{cls} are set to 1 when the given keyword occurs in the video and 0 otherwise; the frame labels y^{loc} are set to 1 for the frames where the keyword is uttered and 0 otherwise. We optimize the total loss $\mathcal{L} = \mathcal{L}^{cls} + \mathcal{L}^{loc} + \lambda\mathcal{L}^{rec}$ where, $\lambda = 1$ if a given input video has a word label annotation, else, $\lambda = 0$ to only train the classification and localization heads.

7.3.2 Multiple Hypotheses (MH) CTC loss

As described earlier, our word labels for fingerspelling recognition are weak labels from subtitles and mouthing cues. The process for obtaining these labels is de-

scribed in Sec. 7.4.1 & 7.4.2. Since this process is automatic, it introduces a degree of label noise. For example, the mouthing model can detect false positives; the detection boundaries can be erroneous, or multiple mouthings can be spotted for a single detection interval. We observed that our automatic detection pipeline is more accurate than our process of obtaining word labels. If we assume the fingerspelling detection is correct, then it is quite likely that the fingerspelled word is among one of the words (usually a noun) in the subtitle.

With this idea in mind, we design an improved CTC loss function, termed Multiple Hypotheses CTC (MH-CTC), that allows the model to "pick" the most correct word label from a set of possible word hypotheses. This set comprises a number of words, of which one is the correct word label for the fingerspelling sequence in the input video clip. For example, the set of word hypotheses could be the proper nouns in the subtitle corresponding to the input video clip. Given a list of word hypotheses $\mathbb{H}$ for the input video clip v, our modified recognition loss $\mathcal{L}^{rec}$ is:

$$\mathcal{L}^{rec} = -\mathbb{E}_{(v,\mathbb{H})\in\mathcal{D}} \min_{\forall h\in\mathbb{H}} CTC(h, \hat{y}^{rec})$$

This corresponds to backpropagating the recognition loss for the word that achieves the minimum CTC loss among all the hypotheses. Since this allows the model to "choose" its own target, we found two strategies that help the model converge. Firstly, pretraining the model with CTC loss using Eqn 7.3 before using MH-CTC is essential. Secondly, when using MH-CTC, we found that randomly (with 50% chance) setting $\mathbb{H}$ to be a single hypothesis containing the word found by our automatic annotation also prevents the model from diverging.

7.4 Automatic annotations

Our model is completely trained with weak labels that are automatically curated. We perform two stages of automatic annotation. In the first stage (Sec. 7.4.1), we automatically annotate fingerspelling detections using exemplars and letter labels using mouthing cues. In the second stage (Sec. 7.4.2), we obtain detection and letter pseudolabels from the Transpeller model that has been trained on annotations from the first stage. The number and duration of these fingerspelling detections, as well as the number of detections associated with either a noun or proper noun letter label, is shown in Tab. 7.1.

	Train			Val		
	#labels	vocab	avg. dur.	#labels	vocab	avg. dur.
Stage 1: Exemplar detections	149k	-	1.4s	3.0k	-	1.3s
w/ mouthing labels: nouns	59k	18k	1.8s	1.2k	0.8k	1.6s
w/ mouthing labels: pr. nouns	39k	14k	1.9s	0.9k	0.5k	1.7s
Stage 2: Transpeller detections	129k	-	1.9s	2.5k	-	1.8s
w/ mouthing labels: nouns	61k	19k	2.1s	1.3k	0.9k	1.9s
w/ mouthing labels: pr. nouns	41k	15k	2.2s	0.9k	0.5k	2.0s
w/ Transpeller labels	111k	32k	2.0s	2.2k	1.4k	1.8s

Table 7.1: **Two stages of automatic annotations.** Stage 1: We use exemplars to detect fingerspelling and mouthing cues to obtain letter labels. Stage 2: We obtain detections and letter labels using Transpeller pseudolabels.

7.4.1 Exemplar and mouthing annotations

Detections using exemplars. We manually annotate a small number E of single frame exemplars of fingerspelling amongst videos from different signers in the training set ($E = 115$). Using these exemplars, we search for frames in continuous signing with high feature similarity to these frames containing fingerspelling. This exemplar-based annotation technique is inspired by [Momeni et al. 2020b; Momeni et al. 2022]. We use features from [Momeni et al. 2022] and compute cosine similarity with fingerspelling features. This simple method provides approximate annotation of fingerspelling detections. We use this method for two reasons: firstly, to help our model learn fingerspelling temporal detection, and secondly, to select video segments containing fingerspelling for manual annotation. Technical details on the computation of the feature similarity can be found in the supplementary material.

Letter labels from mouthings. To obtain word annotations for the fingerspelled segments, we use mouthing cues, as fingerspelled words are often mouthed simultaneously in interpreted data [Davis 1990]. In [Momeni et al. 2022], the authors use an improved Transpotter architecture [Prajwal et al. 2021] to query words from surrounding subtitles and localize corresponding mouthing cues. We consider all mouthing annotations from [Momeni et al. 2022] falling within the interval of the fingerspelling detections. As the fingerspelling detection boundaries are approximate, and the automatic mouthing annotations are not always accurate, these annotations are noisy. Tab. 7.2 shows that almost all fingerspelling annotations refer to nouns, and most refer to proper nouns. Thus, restricting mouthing annotations to nouns or proper nouns reduces noise.

7.4.2 Transpeller annotations

Improving detections with Transpeller pseudolabels. The model described in Sec. 7.3.1 outputs a per-frame localization score, predicting the presence of fingerspelling. After training this model, we can improve upon the exemplar-based detections using pseudolabels. Given localisation scores $s_1, ..., s_N$ for a window of N frames, we consider that the window contains fingerspelling if $\max(s_1, ..., s_N) > t_1$. We consider that a sub-interval $[i, j]$ $(1 <= i <= j <= N)$ of this window contains fingerspelling if $s_i, ..., s_j >= t_2$, where $t_2 < t_1$. To smooth the localization scores, we take the moving maximum score amongst K consecutive scores. We let $t_1 = 0.7$, $t_2 = 0.3$ and $K = 5$.

Improving letter labels with Transpeller pseudolabels. Using pseudolabels from the model in Sec. 7.3.1, we can also improve automatic letter label annotations. After decoding the CTC outputs using beam search, we can then compute a proximity score with words from neighboring subtitles, i.e. dist(w_1, w_2), where w_1 is a subtitle word and w_2 is the output of beam search decoding. The proximity score is a variant of the Levenshtein edit distance, but where deletions are not heavily penalized. This is because words such as 'Sarah Jane' can be reasonably fingerspelled as 'SJ'. Details of this proximity score are provided in the supplementary material. We can use this proximity score to find the subtitle word most likely to be fingerspelled.

7.4.3 Multi-stage training

We perform a multi-stage training strategy where we start by training on exemplar-based annotations (lines 2 and 3 of Tab. 7.1) using the vanilla CTC loss to supervise the recognition head. Upon convergence, we finetune this model further using MH-CTC, this time additionally considering proper nouns in neighboring subtitles as the hypotheses.

Using the above pre-trained model, we now extract the Transpeller annotations as detailed in Sec. 7.4.2. Using these new annotations, we repeat the process: we start with the vanilla CTC loss and then finetune this model further using MH-CTC. Given that we have pseudo-labels from the Stage 1 Transpeller model, we can restrict the MH-CTC search space. The hypotheses now only contain nouns and proper nouns from neighboring subtitles with at least one letter in common with the CTC decoded outputs from the Transpeller model of Stage 1.

7.5 Experiments

7.5.1 BOBSL Fingerspelling benchmark

We collect and release the first benchmark for evaluating fingerspelling in British Sign Language. The test set annotations are collected by adapting the VIA Whole-Sign Verification Tool [Dutta and Andrew Zisserman 2019] to the task of fingerspelling recognition. Given proposed temporal windows around the automatic exemplar and mouthing annotations, annotators mark whether there is any fingerspelling in the signing window and type out the exact letters which are fingerspelled. We use a temporal window of 2.1s before to 4s after the midpoint of the automatically detected fingerspelling instance. Since the fingerspelled letters could be a subset of the actual full word (e.g. SH for SARAH), we also obtain the corresponding full word annotations. Descriptive statistics on the test set annotations are in Tab. 7.2.

Evaluation criteria. We measure the fingerspelling recognition performance using the Character Error Rate (CER), which provides a normalized count of the substitution/deletion/insertion errors in the predicted letter sequence when compared to the ground-truth sequence. We report two CERs for the two different ground-truth annotations we have for each clip: (i) $\text{CER}_{\text{fspell}}$ - ground-truth is the actual fingerspelled letters which, as mentioned before, may only be part of a word, and (ii) CER_{full} - ground-truth is the full word annotation to which the fingerspelling refers to. Given that our model(s) are trained only with automatic weakly-supervised full-word annotations (Sec. 7.4), these two different CER scores can help us see if the model learns to pick up on the fingerspelled letters, rather than only relying on the mouthing cues or memorizing the full word annotations.

Implementation details. We use a batch size of 32 and an initial learning rate of $5e^{-5}$, which is reduced to $1e^{-5}$ after the validation loss does not improve for 3 epochs. At test time, we decode with a beam width of 30. More implementation details are provided in the supplementary material.

7.5.2 Results

We now evaluate different variations at each stage of the Transpeller recognition pipeline.

Using exemplar detections and mouthing cue letter labels with CTC

#labels	(full word) vocab.	%nouns	%pr. nouns	%full	avg. % missing
4923	3442	96%	74%	22%	34%

Table 7.2: **Statistics on test set annotations.** Most fingerspellings refer to proper nouns or nouns. Around 22% of fingerspelling instances contain all letters of the encoded word, but on average 34% of the letters of a word are omitted during fingerspelling.

Annotations	**#Recogn. ex.**	**$\text{CER}_{\text{fspell}}$**	**CER_{full}**
Exemplars + Mouthings: proper nouns	39k	58.5	62.1
Exemplars + Mouthings: nouns	59k	58.6	62.9
finetuned with MH-CTC	59k (avg. 4 hyp.)	**57.6**	64.3

Table 7.3: **Stage 1: Transpeller model with exemplar + mouthing supervision.** All rows use 149k fingerspelling detections for training. When assigning a word label for these detections from the subtitle, we choose to look at nouns, especially proper nouns, as they are more likely to be fingerspelt. We obtain the best results when using MH-CTC loss.

Annotations	**#Recogn. ex.**	**$\text{CER}_{\text{fspell}}$**	**CER_{full}**
Transpeller detect. + Mouthings	61k	57.5	63.1
Transpeller detect. + Char. labels	111k	55.4	63.0
finetuned with MH-CTC	111k + (avg. 9 hyp.)	**53.3**	**60.1**

Table 7.4: **Stage 2: Transpeller pseudolabels.** All rows use 129k fingerspelling detections for training. Using both the refined detections and word annotations from the Stage 1 model gives a clear reduction in the CER. Further, using MH-CTC gives a 2.1 CER boost. Overall, our Stage 2 achieves a final best CER of 53.3 which is 4.3 CER better than the best model of Stage 1, thus validating the impact of our multi-stage training pipeline.

loss. Our initial set of annotations from the exemplar detections gives us 149k fingerspelling instances, out of which a fraction of them can be associated with word labels using mouthing cues. In Tab. 7.3, we show how our performance depends on our choice of recognition annotations. Restricting our automatic word label annotations to proper nouns gives us a cleaner training set, as they are most likely to be fingerspelt. However, this comes at the expense of having very few training samples. In row 2 of Table 7.3, we find that we can tolerate a bit of label noise and expand to all nouns. We finetune the best CTC-based model from the above using our MH-CTC loss, which further results in an improvement of 1.0 $\text{CER}_{\text{fspell}}$. This is our best model with the initial set of annotations.

Transpeller detections. We now expand our training data, by extracting pseudolabels (Sec. 7.4.2) from the best model from the previous stage. We train on these new annotations and report our results in Tab 7.4. When we use the refined detections but still use the mouthing cues for assigning word labels, we obtain a $\text{CER}_{\text{fspell}}$ of 57.5, a similar result to the corresponding model (57.6) from Stage 1.

Transpeller letter labels. The error rates drop further when we also improve the recognition annotations. We do so by using a variant of the edit distance to match the predictions of the Stage 1 Transpeller to a word in a neighboring subtitle. These results can be seen in row 2 of Tab. 7.4.

Finetuning with MH-CTC. Finally, when finetuning further with our MH-CTC loss, our final Stage 2 model gives a $\text{CER}_{\text{fspell}}$ of 53.3 which is 4.3 points better than the best model (57.6) of Stage 1. It is evident from both these tables that both MH-CTC and our multi-stage training with pseudolabelling improve our recognition performance.

Fingerspelling recognition vs. spotting mouthings. An interesting line of thought is: if we have access to the subtitles at test time, can we use a mouthing model to accurately predict words instead of doing fingerspelling recognition? To judge this, we restrict the test set to instances where there is a noun or a proper noun mouthing annotation and compute the CER between the mouthing annotation and the ground truth annotation, and we get 55.6. On this same subset, our best model obtains 53.3 CER, demonstrating that it in fact performs better than a mouthing method with access to the subtitle text.

Error analysis of the best model on the test set. In Tab. 7.5, we show a few examples of our predictions and the corresponding ground-truth sequence. We

can see our model makes reasonable errors for most examples, where it confuses between letters that are visually quite similar. In Fig. 7.4, we show how the CER varies based on the length of the ground-truth character sequence. It is evident that the model struggles the most with very short sequences (one or two letters). This is expected because it has only been supervised with full word annotations during training and has never been trained to predict one or two letters in isolation.

Lookup-based correction at inference-time. We explore the possibility of correcting the errors in the model's outputs at test-time with the help of a pre-determined list of words, e.g. "atlanic" to "atlan**t**ic". We first curate a list of nouns present in the subtitles of the BOBSL train set and use edit distance to match the model's predictions to the closest noun in our list. If the edit distance is below a set threshold, i.e., a very close match, we replace the predicted character sequence with the matched word from our list. However, we found that this increases the $\text{CER}_{\text{fspell}}$ to 57.0 and CER_{full} to 61.1. Such a lookup-based correction method leads to several false matches because (i) it is done with no context of the surrounding words, (ii) not all the letters of the word are fingerspelled, (iii) fingerspelled words in the test set can also be novel and unseen.

7.5.3 Architecture ablations

Importance of joint recognition and detection. As described in Sec 7.3.1, our model contains three prediction heads for classification, localization, and recognition. All three heads are essential and are used to obtain the Stage 2 annotations as described in 7.4.2. We conduct an experiment to also demonstrate that it is beneficial to train these heads jointly. We find that joint training leads to a better recognition performance (55.4 $\text{CER}_{\text{fspell}}$) than training without the localization head (56.3 $\text{CER}_{\text{fspell}}$) or without the localization and classification heads (56.2 $\text{CER}_{\text{fspell}}$).

Sequence-to-Sequence vs. CTC-based models. We also compare our CTC-based recognition head with a sequence-to-sequence (seq2seq) encoder-decoder architecture supervised with a cross-entropy loss. We use the standard Transformer-Base [Vaswani et al. 2017] model, which contains a Transformer encoder similar to Transpeller and a 6-layer auto-regressive Transformer decoder. We compare with the CTC model trained on the Stage 2 annotations. The seq2seq network performs worse (57.4 $\text{CER}_{\text{fspell}}$) than the CTC model (55.4 $\text{CER}_{\text{fspell}}$). This is ex-

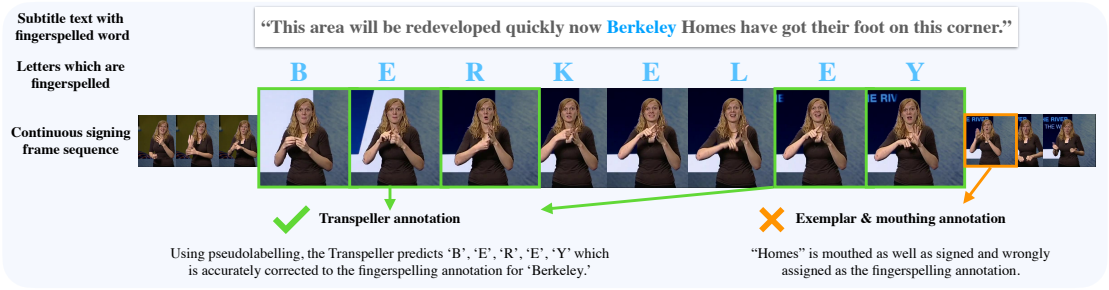


Figure 7.3: **Benefit of Transpeller annotations:** Here, we are given a fingerspelling clip of "BERKELEY". The initial annotation with exemplars + mouthing cues assigns an incorrect word label because the spotted mouthing of the word "Homes" is temporally close to the fingerspelling location, resulting in a false word assignment. In the second annotation stage, we correct this using the Transpeller's predicted characters "BEREY" that are matched to the subtitle word "BERKELEY".

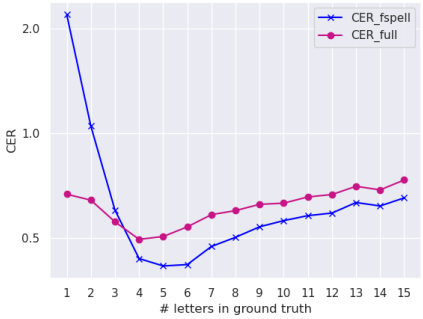


Figure 7.4: **Variation of CER vs the number of letters in the ground-truth.** Transpeller struggles to correctly predict very short fingerspelling segments, which are usually partial fingerspellings (e.g. MJ for Mary Jane). This happens because the model is only supervised with full-length words.

Ground Truth	Prediction	CER
chloride	churide	25
nurembrg	turmug	50
ivory	ener	80
elind	elinc	20
bnmm	ben	75
clove	clune	40
semnoa	samol	50

Figure 7.5: **Qualitative examples.** Model predictions on our manually verified test set. The CERs are shown for reference. We see that one of the most common error sources is confusion between letters that are visually similar: (a, e, i), (d, c), (l, n, m, v, t), (o, u).

pected because only 22% of the test samples contain no missing letters, i.e. full words, so conditioning on past letters can lead to errors in future letter predictions. However, conditioning on past letters can be very helpful if we want to actually estimate the full word and not determine the exact letters that are fingerspelled. This is indeed validated by the fact that the seq2seq model performs much better on CER_{full}, achieving 55.2 compared to the CTC model's 63.0. We also note that due to the sequential decoding in the seq2seq model, it is $\approx 20\times$ slower in terms of run-time speed than the CTC model, which decodes all the characters in parallel.

7.6 Conclusion

We presented a new BSL fingerspelling recognition benchmark and our Transpeller model designed to jointly detect and recognize fingerspelled letters in continuous sign language video. Our training data is largely constructed automatically, exploiting English subtitles and mouthing cues. The evaluation data is manually curated, and we achieve promising recognition results (Tab. 7.5). However, we note some limitations. First, there remains room for improvement in the accuracy of the model to further reduce the character error rate. Second, our training and evaluation of the Transpeller is limited to the use of interpreted data and therefore is not necessarily representative of more natural, conversational signing. Addressing these limitations would be a valuable future research direction.

Acknowledgements. This work was supported by the Oxford-Google DeepMind Graduate Scholarship, EPSRC grant ExTol, a Royal Society Research Professorship and the ANR project CorVis ANR-21-CE23-0003-01. HB would like to thank Annelies Braffort and Michèle Gouiffès for the support.

Chapter 8

A Large-Vocabulary Continuous Sign Language Recognition Benchmark

This paper is a technical report presenting ongoing works, written in 2024.

A Large-Vocabulary Continuous Sign Language Recognition Benchmark

Charles Raude[1,2*] K R Prajwal[2*] Liliane Momeni[2*] Hannah Bull[1]

Samuel Albanie[3] **Andrew Zisserman**[2] **Gül Varol**[1,2]

[1] LIGM, École des Ponts, Univ Gustave Eiffel, CNRS, France

[2] Visual Geometry Group, University of Oxford, UK

[3]Department of Engineering, University of Cambridge, UK

https://imagine.enpc.fr/~varolg/cslr2/data.html

Abstract

In this work, our goal is large-vocabulary continuous sign language recognition (CSLR) — the task of predicting a sequence of signs from a given continuous signing video clip, without constraints on the lexicon. Previous works have been unable to focus on this realistic setting due to the absence of appropriate benchmarks. We introduce here a new CSLR test benchmark, BOBSL-CSLR-TEST, consisting of manually annotated continuous sign-level data. It is the largest CSLR evaluation benchmark to date, both in terms of vocabulary (5K different signs) and duration (6 hours), containing 32K gloss annotations. To solve the CSLR task, we also propose a Transformer model, that is able to ingest a signing sequence and output in a joint embedding space between signed language and spoken language text. We demonstrate that by a careful choice of loss functions, training the model

*Equal contribution.

jointly for both CSLR and sign language retrieval tasks improves the CSLR performance by providing context. Our model significantly outperforms the previous state of the art on our newly collected CSLR benchmark, serving as a strong baseline for the community.

8.1 Introduction

Recognising continuous and large-vocabulary sign language is a vital step towards enabling real-world technologies that enhance communication and accessibility for the deaf or hard of hearing. With the availability of data that depicts continuous signing from a large vocabulary of signs [Albanie et al. 2021b; N. Camgoz et al. 2021; Duarte et al. 2021], the computer vision field has recently gained momentum towards this direction, building on previous research that had largely focused on restricted settings such as recognising single signs in isolation [Joze and Koller 2019; D. Li et al. 2019] or signs covering relatively small vocabularies [Koller et al. 2015a; H. Zhou et al. 2020a].

Our goal in this paper is to enable large-vocabulary continuous sign language recognition (CSLR) – providing time-aligned and dense word predictions for each sign within a signing sequence. This is an essential first step towards translation, as English sentence-level annotations have been shown to be difficult to use directly as targets for sign language translation [Varol et al. 2021; Albanie et al. 2021b; N. Camgoz et al. 2021].

However, there are several challenges to achieving this goal, primarily due to the lack of suitable data for training and evaluation. For CSLR, ideally, *each* individual sign within a continuous video should be associated to a symbolic category. One challenge is that current training supervision sources are restricted by their *weak* or *sparse* nature. For instance, in the largest dataset BOBSL [Albanie et al. 2021b], the available annotations are either (i) at sentence-level, weakly associating the entire signing sequence in the video to an English sentence, rather than breaking the video into individual sign-word correspondences, or (ii) at sign-level, but sparse with gaps in the temporal timeline (despite the densification efforts in [Momeni et al. 2022] to scale up the number and vocabulary of annotations). At the same time, it is not possible to assess and compare the performance of large-vocabulary CSLR algorithms at scale, as there is no evaluation set with dense ground truth

sign-level annotations in the BOBSL dataset.

In this paper, we introduce a CSLR evaluation benchmark comprising six hours of manually collected new sign-level annotations that are continuous on the timeline. Since we focus on the large-vocabulary setting, we collect annotations on the BOBSL test set. As shown in Fig. 8.1, we also provide accurate sign-level temporal segments. When compared to PHOENIX-2014 [Koller et al. 2015a], which is the largest existing CSLR test set collected in non-laboratory settings, our test set contains 6× more hours with 4× more gloss labels spanning a vocabulary of 5000 words, which is also 10× larger than the current best test set in non laboratory settings.

We also introduce a simple Transformer encoder model [Vaswani et al. 2017], CSLR^2, that ingests a signing video sequence and outputs tokens in a joint embedding space between signed and spoken[1] languages. The output space enables training the CSLR^2 model jointly with CSL**R** and sentence **R**etrieval tasks (i.e. retrieve the most similar text sentence given a signing video or vice versa), which we find improves CSLR performance. The joint language space may also help to overcome another challenge of sign recognition: polysemy, where the same word may correspond to several sign variants, and conversely the same sign may correspond to several different words. The benefits of a joint embedding space for spoken and sign language has been explored before. The LCC paper [Wong et al. 2023] bridges the gap between sign and spoken language word embeddings. Our approach has a similar motivation, but at the sentence-level, specifically to help CSLR. For training our model, we note that we only use noisy annotations from the large-scale BOBSL dataset for supervision.

In summary, our contributions are the following: (i) a new benchmark of substantial size for evaluating large-vocabulary CSLR; (ii) extensive experimental analyses to highlight the challenge of the new benchmark; and (iii) a new method in the form of a multi-task Transformer network that leverages weak training supervision and outperforms existing methods on the new test for large-vocabulary CSLR. Our code and data is publicly available for research, linked from our project page.

[1]We refer to the written form of spoken language, not the speech audio.

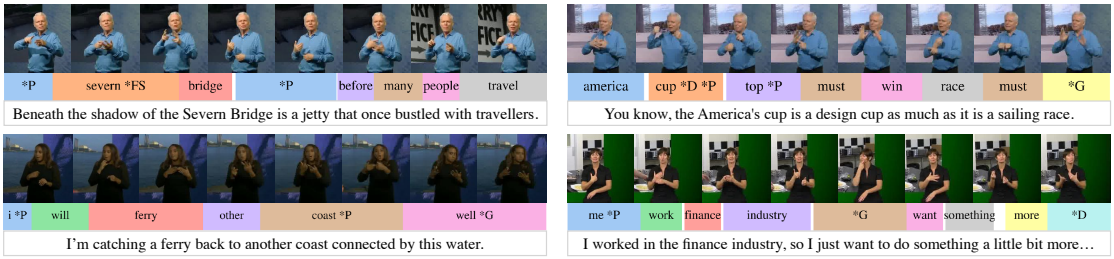


Figure 8.1: **Annotation examples from the BOBSL-CSLR-test dataset:** As well as assigning the English word(s) corresponding to a sign (i.e. 'gloss'), the annotations provide the temporal segment of each sign, and indicate the type of sign when appropriate. For example, '*P' for pointing, '*FS' for fingerspelling, '*G' for gesture sign.

Table 8.1: **Recent CSLR training and evaluation sets:** Our manually-curated BOBSL-CSLR-TEST set is larger in the number of annotated signs and vocabulary, compared to other standard CSLR test sets from the literature. In addition, it also comes with sign segmentation annotations. *Note that the BOBSL training has different vocabulary sets of varying sizes: 72K words spanned by subtitles and 25K words spanned by 5.5M automatic annotations generated in [Momeni et al. 2022].

		segmented	#sentences	hours	vocab.	#glosses	source
train	PHOENIX-2014 [Koller et al. 2015a]	✗	6K	11	1K	65K	TV
	CSL-Daily [H. Zhou et al. 2020a]	✗	18K	21	2K	134K	lab
	BOBSL [Albanie et al. 2021b]	✗	993K	1220	72K*	5.5M*	TV
	BOBSL-CSLR-TrainVal	✓	2.6K	3.0	3.1K	15.3K	TV
test	PHOENIX-2014 [Koller et al. 2015a]	✗	629	1.0	0.5K	7.1K	TV
	CSL-Daily [H. Zhou et al. 2020a]	✗	1176	1.4	1.3K	9.0K	lab
	BOBSL-CSLR-TEST	✓	4518	6.0	5.1K	32.4K	TV

8.2 Related Work

We briefly discuss relevant works on CSLR benchmarks and methods that operate on continuous sign language video streams.

CSLR benchmarks. Early works with continuous signing videos provided very small vocabularies in the order of several hundreds (104 signs in Purdue RVL-SLLL [R. B. Wilbur and Kak 2006] and BOSTON104 [Dreuw et al. 2008] ASL datasets, 178 in CCSL [Huang et al. 2018b], 310 in GSL [Adaloglou et al. 2020], 455 in the SIGNUM DGS dataset [von Agris et al. 2008], and 524 in the KETI KSL dataset [Ko et al. 2019]). BSL Corpus [Schembri et al. 2017] represents a large-vocabulary collection; however, it is mainly curated for linguistics studies, and has not been used for CSLR.

Relatively large collections made it possible to train CSLR methods based on neural networks. RWTH-PHOENIX-Weather 2014 [Koller et al. 2015a], which is the most widely used dataset, contains around 11 hours of videos sourced from

weather forecasts. CSL-Daily [H. Zhou et al. 2020a] provides 20K videos with gloss and translation annotations from daily life topics, covering a 2K sign vocabulary, and 23 hours of lab recordings.

In Tab. 8.1, we provide statistics to compare against our new benchmark, mainly on their evaluation sets. Recently released large-vocabulary continuous datasets (such as BOBSL [Albanie et al. 2021b], How2Sign [Duarte et al. 2021], Content4All [N. Camgoz et al. 2021], and OpenASL [B. Shi et al. 2022a]) do not provide sign-level gloss annotations due to the prohibitive costs of densely labeling.

Continuous sign language recognition in video streams. In recent years, the community has started to move beyond isolated sign language recognition (ISLR) [Joze and Koller 2019; D. Li et al. 2019], which only seeks to assign a category (typically also expressed as a word) to a short video segment trimmed around a single sign without context. As well as CSLR, several recognition tasks require ingesting a continuous video stream exist, including sign spotting [Albanie et al. 2020; Momeni et al. 2020b; Varol et al. 2021; Momeni et al. 2022], sign tokenization [Renz et al. 2021a; Renz et al. 2021b], translation [N. C. Camgoz et al. 2018; N. C. Camgoz et al. 2020b; N. C. Camgoz et al. 2020a; B. Zhou et al. 2023; A. Yin et al. 2023], subtitle alignment [Bull et al. 2021], subtitle segmentation [Bull et al. 2020], text-based retrieval [Duarte et al. 2022; Y. Cheng et al. 2023], fingerspelling detection [Prajwal et al. 2022b], active signer detection and diarization [Albanie et al. 2021a]. Our work is related to some of these tasks in that they also operate on a large-vocabulary setting [Albanie et al. 2020; Momeni et al. 2020b; Varol et al. 2021; Bull et al. 2021; Momeni et al. 2022]; however, these papers do not tackle CSLR, mainly due to lack of continuous sign annotations. While [Duarte et al. 2022] addresses retrieval, their method is not suitable for CSLR – our work differs in that we show training for both isolated sign recognition and text-based retrieval tasks benefits CSLR performance.

State-of-the-art CSLR methods have so far focused on PHOENIX-2014 [Koller et al. 2015a] or CSL-Daily [H. Zhou et al. 2020a] benchmarks, where the performances are saturated. These methods typically consider a fully-supervised setting, and train with RNN-based [Huang et al. 2018b; R. Cui et al. 2019; Huaiwen Zhang et al. 2023], or Transformer-based [N. C. Camgoz et al. 2020b] models. Due to lack of sign segments (i.e., start and end times of signs are unknown), many works use the CTC loss [N. C. Camgoz et al. 2020b; K. L. Cheng et al. 2020; Jiao et al.

2023; Wei and Y. Chen 2023; Zuo and Mak 2022]. Our work differs from these previous works on several fronts. We consider a weakly-supervised setting, where the training videos are *not* annotated for CSLR purposes, but are accompanied with weakly-aligned spoken language translation sentences. We also study the benefits of joint training with CSLR and retrieval objectives. In a similar spirit, the works of [N. C. Camgoz et al. 2020b; Zuo and Mak 2022] jointly train CSLR with sentence-level objectives (translation in [N. C. Camgoz et al. 2020b], margin loss for gloss-sequence text retrieval in [Zuo and Mak 2022]), but in significantly different settings (e.g., 8× smaller vocabulary, and with manually-annotated CSLR labels for training).

8.3 A New CSLR Evaluation Benchmark

In this section, we describe the new continuous sign annotations that we collected for evaluating CSLR. We first describe the data splits in BOBSL-CSLR, what the BOBSL-CSLR-TEST is, and then how it was annotated. Finally, we provide the evaluation protocol for benchmarking.

BOBSL-CSLR. We collect continuous sign-level annotations for a subset of the BOBSL dataset [Albanie et al. 2021b]. We follow the official train-val-test splits of the dataset, and annotate a small subset of train and val splits, and a larger portion of the test split. The statistics of the collected data for each split is specified in Table 8.2. Our main focus, though, is on the test split, BOBSL-CSLR-TEST, which we describe in detail below.

Table 8.2: **CSLR annotations on BOBSL:** We provide statistics for our collected CSLR annotations on BOBSL in terms of the number of subtitle sentences, the duration in hours, the number and vocabulary of subtitle words, the number and vocabulary of annotated glosses, and the number of annotated sign types. While our focus is to collect a large evaluation split (BOBSL-CSLR-TEST), we also collect annotations in a similar way for a small proportion of the train and validation splits of BOBSL.

	#sentences	#hours	#subtitle words	subtitle vocab.	#glosses	gloss vocab.	#sign type annot.
all	7134	8.9	81.3K	9.0K	47.7K	7.1K	17.1K
train	1463	1.7	15.0K	3.1K	8.7K	2.6K	3.4K
val	1153	1.3	13.4K	2.7K	6.6K	2.1K	2.0K
test	4518	6.0	52.9K	6.9K	32.4K	5.1K	11.7K

BOBSL-CSLR-test. The continuous annotations are provided for a subset of the SENT-TEST partition of BOBSL [Albanie et al. 2021b]. SENT-TEST contains 31 hours of BSL signing sequences that have been manually aligned temporally with their corresponding English subtitle sentence translations.

The BOBSL-CSLR-TEST annotations consist of a time aligned sequence of sign 'glosses'[2], where each sign is annotated with its temporal interval, the type of the sign, and its word equivalent if that exists. In addition to lexical signs (i.e. signs that have an English word equivalent), a wide range of sign types such as fingerspelling, pointing, depicting or no-signing are annotated. These are marked with special characters such as *FS and *P for fingerspelling and pointing, respectively. The various sign types are illustrated in Fig 8.2. We briefly describe the major sign types in our benchmark. Pointing (*P) contains the pointing *hand-shape*, e.g. depicting pronouns by pointing to oneself ("I"/"me") or another person ("you"). Fingerspelling refers to signing letter by letter using the manual alphabet, e.g. to spell proper names. We also have depicting signs (*D), which depict shape and size, and gestures (*G) which are hand or body poses that non-signers also do such as shrugging shoulders. We also annotate signs that are depicted through meaningful facial expressions (*FE). A few more sign types and their explanations are given in Sec. 8.8 of the Appendix.

Note, that there exists no universally accepted writing system for sign languages today [Filhol 2020], though attempts have been made with descriptive languages such as HamNoSys [Hanke 2004] and SignWriting [Sutton 1990]. Also, careful glossing that is linguistically consistent (e.g., enumerating each sign variant [Schembri et al. 2017]) is a tedious process which hinders scaling up. For these reasons, we make a compromise when annotating for CSLR and use English words for the glosses, assigning 'any' reasonable English word for a sign segment, but prioritising words in the surrounding subtitle. For example, if the 'natural' English word for the sign is 'laugh' but a synonym word such as 'giggle' is in the subtitle, then the gloss would be 'giggle' (with 'laugh' also provided as a more general translation). However, one should keep in mind that associating words to signs is a lossy and error-prone process in any case.

Fig. 8.1 shows example ground truth gloss annotations for BOBSL-CSLR-

[2]We abuse the gloss terminology, despite our sign-level annotations *not* being careful linguistic glosses, but rather free-form sign-level translations.

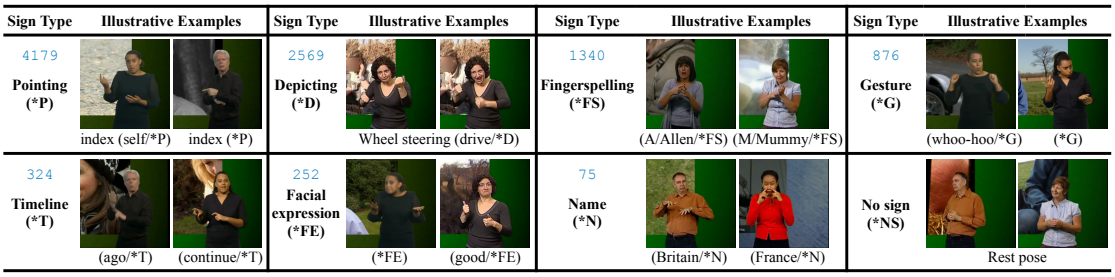
TEST. In total, we curate these continuous labels for 6 hours of video, comprising 32.4K individual signs from a vocabulary of approximately 5.1K glosses. The BOBSL-CSLR-TEST annotations evenly cover all 35 episodes in SENT-TEST. About 3% of the the gloss vocabulary in the test split are completely new – they do not occur in the subtitles of the BOBSL train set. Additional statistics for the dataset are given in Sec. 8.8 of the Appendix. All annotations, together with a datasheet, are publicly released from our project page.

Dataset annotation. The annotation procedure uses a web based annotation tool that is built from the VIA video annotation software [Dutta and Andrew Zisserman 2019]. Annotators are provided with a video sequence with 10 time aligned subtitle sentences. Annotators enter free-form text for each sign token, taking into account the context by watching the full video around a given subtitle. The subtitle is also displayed on the video and the annotators are encouraged to prioritise assigning words that appear in the corresponding subtitle. The annotators additionally assign sign types when appropriate (see Fig. 8.1) and temporally align each gloss to the duration of the sign.

We facilitate faster annotation iterations by incorporating several strategies. We adopt a semi-automatic labeling technique where we initialise the sign boundaries by using an automatic sign segmentation method [Renz et al. 2021a]. Annotators are thus given an initial set of sign intervals, and are instructed to refine these sign boundaries (or add/remove sign intervals) if necessary. We further show a dropdown menu for each sign and prioritise at the top of the list words from the corresponding subtitle. Two native BSL users worked on this task for over one year. Further details on the annotation procedure and quality control are provided in Sec. 8.7 of the Appendix, as well as the annotation instructions which are linked from our project page.

Evaluation protocol. The CSLR evaluation is on the manually annotated test set (BOBSL-CSLR-TEST) as described above, which contains video sequences corresponding to 4518 unseen test subtitles. As in other CSLR benchmarks [Koller et al. 2015a; H. Zhou et al. 2020a], we report word error rate (**WER**) as our main performance measure. We also monitor **mIoU** (mean intersection over union) between predicted and ground truth sequences' words, without considering any temporal aspect. In all metrics, similar to [Momeni et al. 2022], we do not penalise output words if they are synonyms.

Figure 8.2: **Sign type examples:** We show 2 illustrative samples for each non-lexical sign type annotation, together with the number of occurrences in BOBSL-CSLR-TEST in blue. See Tab. A.1 of the Appendix for a complete list of statistics and explanation of each sign type. We note that we omit the number of occurrences for no signing segments as these do not have a meaningful unit like signs.



In order to assess the model's ability to correctly predict sign segments at the right temporal location, we also report the F1 score. We define a segment to be correctly predicted if (i) the predicted sign matches, up to synonyms, the ground-truth gloss, and (ii) the IoU between the predicted segment's boundaries and the ground truth gloss segment's boundaries is higher than a given threshold. We compute the F1 score as the harmonic mean of precision and recall, based on this definition of correct segment detection. We report the F1 score at different thresholds values, namely, $\mathbf{F1@\{0.1, 0.25, 0.5\}}$.

Given a video sequence, the evaluation compares the predicted gloss sequence (after post-processing the raw per-frame outputs with the optimal θ and m heuristics – see Sec. 8.9.1 of the Appendix for post-processing strategy) with the corresponding ground-truth gloss sequence.

We only evaluate on segments associated with a word in the ground-truth sequence, e.g. a segment annotated as *G without a word label will be filtered out from the ground-truth sequence. This gives us about 28K remaining gloss instances. Note that we keep sign types like pointing and fingerspelling that *have a word* associated with them. In addition, if several annotation words are associated to one sign (e.g. 'giggle' and 'laugh'), predictions are considered to be correct if one of these words is predicted correctly.

8.4 CSLR^2: A Joint Model for CSLR and Retrieval

We start by describing the CSLR^2 model design that goes from raw sign language video pixels to a *joint embedding space* with spoken language text (Sec. 8.4.1). We then present the losses of our joint training framework with sentence-level and sign-level objectives (Sec. 8.4.2). The details about our sources of supervision which consists of (noisy) sign-level pseudo-labels and weakly-aligned subtitles are given in Sec. 8.9.1 of the Appendix.

8.4.1 Model overview and inference

Our model, shown in Fig. 8.3, consists of 3 main components: (i) a sign video encoder $\mathcal{V}_{enc}^{\text{Sign}}$ based on a pretrained Video-Swin [Z. Liu et al. 2022], (ii) a sentence video encoder $\mathcal{V}_{enc}^{\text{Sent}}$ based on a randomly initialised Transformer encoder, and (iii) a text encoder $\mathcal{T}_{enc}$ based on a pretrained T5 model [Raffel et al. 2020]. Given raw RGB video frame pixels V for a signing sentence, we obtain a sequence of isolated sign video embeddings from $\mathcal{V}_{enc}^{\text{Sign}}$ as $\{\hat{\mathbf{v}}_f\} = \mathcal{V}_{enc}^{\text{Sign}}(V)$. In practice, such a sequence is obtained by feeding 16 consecutive frames to the sign video encoder in a sliding window fashion, with a 2-frame stride. These are then fed through the sentence video encoder $\mathcal{V}_{enc}^{\text{Sent}}$ to have *context-aware* sign video embeddings $\{\mathbf{v}_f\}$, as well as a single sentence video embedding $\mathbf{V}$, denoted $(\{\mathbf{v}_f\}, \mathbf{V}) = \mathcal{V}_{enc}^{\text{Sent}}(\{\hat{\mathbf{v}}_f\})$. Similarly, for the text side, we embed the sentence T into $\mathbf{T} = \mathcal{T}_{enc}(T)$. We also define sign-level text embeddings for each sign in the sentence as $\{\mathbf{t}_w\}_{w=1}^{W} = \{\mathcal{T}_{enc}(t_w)\}_{w=1}^{W}$, where W is the number of signs in the sentence. In practice, we get these embeddings by independently feeding the word(s) corresponding to each sign to the text encoder. Please refer to the Sec. 8.9.2 of the Appendix for details on each component of our model.

CSLR inference. Sign-level recognition predictions are obtained by using the sequence of sign video embeddings in $\mathbf{v}_f$ that lies in the same space as spoken language. To associate each feature frame f to a word (or phrase), we perform nearest neighbour classification by using a large text gallery of sign category names, as illustrated in Fig. 8.3b. In our experiments, we observe superior performance of such retrieval-based classification over the more traditional cross-entropy classifi-

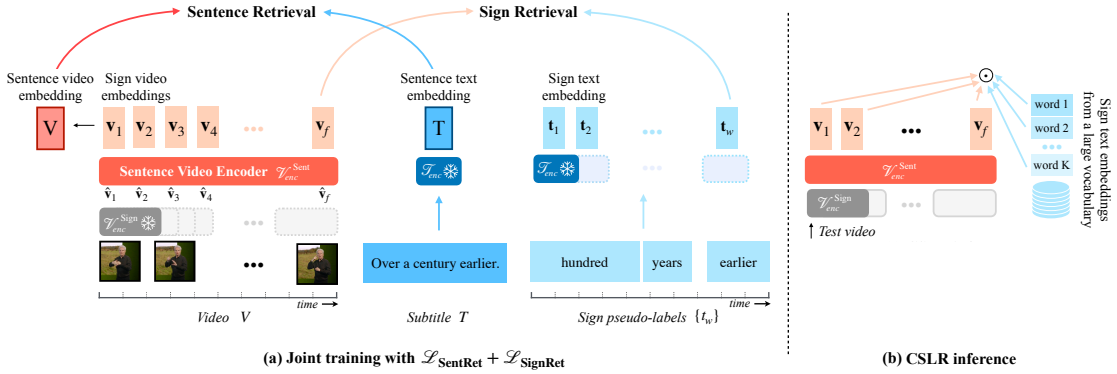


Figure 8.3: **CSLR^2 method overview:** (a) We show a simplified view for our model architecture which consists of both video and text streams. On the video side, features are extracted from a signing video clip V by running $\mathcal{V}_{enc}^{\text{Sign}}$ in a sliding window fashion and passed through a Transformer model $\mathcal{V}_{enc}^{\text{Sent}}$. A video embedding $\mathbf{V}$ and sign video embeddings $\{\mathbf{v}_f\}$ are subsequently extracted. On the text side, we input an English subtitle sentence T and sign pseudo-labels $\{t_w\}$ to the text encoder $\mathcal{T}_{enc}$ and obtain sentence and sign text embeddings ($\mathbf{T}$, $\{\mathbf{t_w}\}$), respectively. While we illustrate only one triplet data point $(V, T, \{t_w\})$, in practice, we operate on a minibatch of triplets, and employ two contrastive losses to jointly train on sentence retrieval $\mathcal{L}_{\text{SentRet}}$ and sign retrieval $\mathcal{L}_{\text{SignRet}}$. (b) For CSLR inference, each sign video embedding is matched to the top-ranked word from a large vocabulary of size 8K. A post-processing strategy is applied on frame-level predictions to produce final outputs. For visibility, we omit linear layers which project embeddings into the learnt joint-space. See Sec. 8.4.1 for a detailed description of the architecture and inference procedure.

cation [M. Wang et al. 2021], with the advantage that it is potentially not limited to a closed vocabulary. In order to go from per-feature classification to continuous sign predictions, we perform a post-processing strategy detailed in Sec. 8.9.1. We note that this same post-processing strategy is used for obtaining our sign-level pseudo-labels for training.

8.4.2 Training with sentence- and sign-level losses

We train the Transformer-based model, that operates on sentence-level sign language videos, for two tasks, namely, CSLR and sign language Retrieval (CSLR^2). As illustrated in Fig. 8.3a, we employ two retrieval losses: (i) a sentence-level objective, supervised with weakly-aligned subtitles, and (ii) a sign-level objective, supervised with pseudo-labels obtained from a strong ISLR model [Prajwal et al. 2022b] (see Sec. 8.9.1 of the Appendix for details on supervision sources). Note that we do not employ the manual annotations from BOBSL-CSLR-TrainVal for training our model. We next formulate each objective individually before introducing our joint framework that leverages both sentence-level and sign-level

information.

Sentence-level objective: sign language sentence retrieval (SentRet). We explore the task of retrieval as a means to obtain supervision signal from the subtitles. Following the success of vision-language models building a cross-modal embedding between images and text [Radford et al. 2021; Junnan Li et al. 2022], we employ a standard contrastive loss, and map sign language videos to spoken language text space.

Sign language sentence retrieval is made of two symmetric tasks, that is, V2T and T2V retrievals. For the former, given a query signing video V, the goal is to rank a gallery of text samples (here subtitles) such that the content of V matches the content in the highest ranked texts. Symmetrically, in the latter, given a text query T, the goal is to rank a gallery of signing videos. Formally, given a dataset $\mathcal{D} = \{(V_i, T_i)\}_{i=1}^{N}$ of video-subtitle pairs, the goal is to learn two encoders ϕ_V, ϕ_T mapping each signing video V and subtitle T into a joint embedding space. The encoders are trained using a recently proposed Hard-Negative variant of InfoNCE [van den Oord et al. 2018], HN-NCE [Radenovic et al. 2023], that re-weighs the contribution of each element in the contrastive loss.

Sign-level objective: sign classification via sign retrieval (SignRet). Given a continuous signing video, the goal of CSLR is to recognise a sequence of individual signs. The continuous video is encoded into a sequence of context-aware sign video embeddings $\{\mathbf{v}_f\}_{f=1}^{F}$, with F the number of video frames. Again, since these embeddings are in the same joint space as the text embeddings, a contrastive loss can be used as a proxy for sign retrieval.

Similarly to the sentence-level retrieval, we use the HN-NCE contrastive formulation for the sign retrieval (SignRet) loss. However, instead of the *full sentence* video-text embedding pair $(\mathbf{V}, \mathbf{T})$, we map *individual* sign video-word embedding pairs $(\mathbf{v}, \mathbf{t})$ (see Fig. 8.3b).

Overall loss. Our model is trained jointly using a weighted sum of the two retrieval terms:

$$\mathcal{L} = \lambda_{\text{SentRet}}\mathcal{L}_{\text{SentRet}} + \lambda_{\text{SignRet}}\mathcal{L}_{\text{SignRet}}$$

with $\mathcal{L}_{\text{SentRet}}, \mathcal{L}_{\text{SignRet}}$, i.e. two contrastive losses for sentence and sign retrieval, respectively. The training hyperparameters can be found in Sec. 8.9.2 of the Ap-

pendix.

8.5 Experiments

In this section, we first describe baselines (Sec. 8.5.1). Next, we provide ablations on the CSLR^2 model, to assess the contribution of important components (Sec. 8.5.2). We then report CSLR performance, comparing to the baselines (Sec. 8.5.3), and give qualitative results, illustrating the current challenges (Sec. 8.5.4).

8.5.1 Baselines

Subtitle-based automatic annotations CSLR baseline. The first baseline for CSLR is obtained using spotting methods that search for signs corresponding to words in the spoken language subtitles. The initial set of sparse sign annotations along with timestamps was released with the BOBSL dataset [Albanie et al. 2021b], using sign spotting methods from [Albanie et al. 2020; Momeni et al. 2020b; Varol et al. 2021], followed by denser spottings in [Momeni et al. 2022]. We evaluate these existing sequences of spottings, in particular the ones corresponding to our BOBSL-CSLR-TEST subtitles. In practice, we filter these automatic annotations using the same sets of thresholds as in [Albanie et al. 2021b; Momeni et al. 2022]. Note that these spottings make use of the weakly-aligned subtitles (and cannot go beyond words in the subtitles), and therefore cannot be used as a true CSLR method. They are also point annotations, without precise temporal extent, thus we omit F1 scores.

ISLR baselines for CSLR. This set of baselines uses ISLR models in a sliding window fashion to obtain continuous frame-level predictions. We aggregate the sliding window outputs by performing the post-processing strategy described in Sec. 8.9.1 (for optimal θ and m parameters, tuned on unseen manually annotated CSLR sequences from the validation set). We use the I3D [J. Carreira and A. Zisserman 2017] and Video-Swin [Z. Liu et al. 2022] models trained from spotting annotations of [Albanie et al. 2021b; Momeni et al. 2022]. We build on prior works for these baselines: we use the I3D weights released in [Albanie et al. 2021b] and train Video-Swin-Tiny similarly to [Prajwal et al. 2022b].

Table 8.3: **CSLR² ablations on BOBSL-CSLR-test:** We ablate the loss component of our model. Only using CTC, cross entropy (CE), or SignRet does not perform well, remaining below or comparable to the ISLR baseline. We observe best results when incorporating joint sentence retrieval training.

SentRet	Sign-level loss	WER ↓	mIOU ↑	F1@{0.1, 0.25, 0.5} ↑		
ISLR Baseline		71.7	30.1	40.0	37.9	27.2
✗	CTC	75.0	27.8	-	-	-
✗	CE	72.9	28.6	37.9	37.3	29.1
✗	SignRet	71.2	30.4	40.7	39.9	31.0
✓	CE	71.0	30.4	39.7	38.9	30.9
✓	SignRet	**65.5**	**35.5**	**47.1**	**46.0**	**37.1**

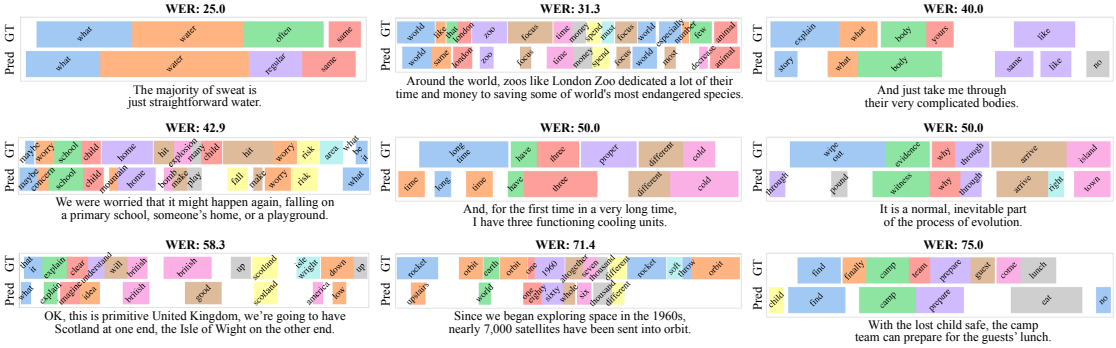


Figure 8.4: **Qualitative CSLR results:** We compare our CSLR² model's predictions (Pred) against the ground truth (GT), providing examples from several error ranges (sorted by WER). The subtitles displayed below each example are not used by the model. While we observe that our model correctly predicts a large portion of signs, handling both English synonyms as well as sign language polysemy (two visually similar signs with different meanings) makes the CSLR task challenging. Synonyms are depicted with the same color coding, e.g. 'earth' and 'world' in 3rd row, middle.

8.5.2 CSLR² ablation study

In Tab. 8.3, we experiment with the choice of training objectives for CSLR performance. In particular, we train with the standard CTC or cross-entropy (CE) losses, as well as our sign retrieval (SignRet) loss alone. Without an additional sentence retrieval loss, i.e. if we only optimise for a sign-level objective, we observe that the performance is worse than the strong ISLR baseline (i.e. with Video-Swin) for CTC and CE, and comparable for SignRet. In the final two rows, we observe clear gains by combining the SentRet loss with either (i) our SignRet loss or (ii) the standard CE loss. While the joint training with the CE loss brings a performance boost, from 72.9 to 71.0 WER, it does not significantly surpass the competitive ISLR baseline. Our model CSLR², which jointly trains sentence and sign retrieval, brings a major improvement by reducing the WER by 5.7 points, from 71.2 to 65.5.

Table 8.4: **Comparison to baselines:** The CSLR^2 model significantly outperforms the existing baselines. Note that the automatic spotting annotations [Albanie et al. 2021b; Momeni et al. 2022] have *access to the subtitles* at inference (unlike our fully automated approach). We also compare to raw ISLR outputs from sign classification models from [Albanie et al. 2021b; Momeni et al. 2022; Prajwal et al. 2022b] with various backbones (I3D or Swin) and with various vocabularies (2K or 8K categories). Our optimal filtering and post-processing strategy at inference is denoted with † (see. 8.9.1). We note that for the ISLR I3D-2K baseline without †, we still remove consecutive repetitions.

CSLR Model	BOBSL-CSLR-TEST WER ↓	mIOU ↑	F1@{0.1, 0.25, 0.5} ↑		
Sub-based spotting [Albanie et al. 2021b]	93.8	7.1	-	-	-
Sub-based spotting [Momeni et al. 2022]	85.5	18.1	-	-	-
ISLR I3D-2K [Albanie et al. 2021b]	453.5	8.6	10.6	8.5	5.4
ISLR I3D-2K [Albanie et al. 2021b] †	82.5	18.1	25.7	23.8	16.0
ISLR I3D-8K [Momeni et al. 2022] †	74.7	27.0	36.7	35.4	26.7
ISLR Swin-8K [Prajwal et al. 2022b] †	71.7	30.1	40.0	37.9	27.2
CSLR^2 (OURS) †	**65.5**	**35.5**	**47.1**	**46.0**	**37.1**

8.5.3 Comparison to baselines

First, in terms of the baselines, it can be seen from Tab. 8.4, that our post-processing strategy significantly strengthens the original ISLR I3D-2K [Albanie et al. 2021b] performance by removing significant noise (with/without †) – we reduce the WER by more than a factor of 5 (453.5 vs 82.5). Also, our post-processing strategy combined with the 8K vocabulary ISLR models, delivers models of higher performance (by more than 10 WER) than all subtitle-based spottings methods, even though the ISLR models do not have access to the subtitles. Second, our joint model, CSLR^2, outperforms all CSLR baselines by a significant margin on all metrics. Indeed, CSLR^2 surpasses the best subtitle spotting method by 20 WER and the strongest ISLR baseline by 6.2 WER. Please refer to Sec. 8.10.3 of the Appendix for a breakdown of performance based on different sign types.

8.5.4 Qualitative analysis

In Fig. 8.4, we show several qualitative examples of our CSLR predictions (Pred rows) against the corresponding ground truth (GT rows) on BOBSL-CSLR-TEST. Note that we display the corresponding ground truth subtitles below each example to give context to the reader, but they are not given as input to the model. We illustrate examples from several error ranges, sorting them by WER per sample (reported at the top). These timelines show that our model is able to

predict a large proportion of the annotations, in the correct order, with approximate sign segmentation. For instance, even though the bottom-right example in Fig. 8.4 has a high error rate of 75 WER, our predictions correctly identify 4 out of the 8 ground truth words, and catch the meaning of the sentence.

However, we also observe several challenges: (i) our model has difficulty predicting several words for a single sign, as the 8K training vocabulary of pseudo-labels primarily comprises of individual words (e.g. the phrase 'long time' is associated to a single sign in the 2nd row - middle, but our model predicts two separate signs 'long' and 'time', leading to an extra insertion); (ii) our model performance is sensitive to the synonyms list, which must be carefully constructed to not unfairly penalise predictions (e.g. in the top left example, 'regular' is counted as a substitution since 'often' is not present in the synonyms list of 'regular'); (iii) our model still struggles with visually similar signs in BSL which correspond to different English words (e.g. 'upstairs' and 'rocket' signs are visually similar, both signed by pointing upwards, 3rd row - middle); (iv) finally, our model is more likely to fail in recognising names of places and people as these are often fingerspelled in BSL and may therefore not be in our 8K sign vocabulary of pseudo-labels (e.g. 'isle wright' is predicted as 'america' in the bottom left). Future directions include addressing such limitations. We include more visual examples in Sec. 8.12 of the Appendix and in the video linked from our project page.

8.6 Conclusion

In this work, we carefully collect a large-vocabulary CSLR benchmark, consisting of 6 hours of continuous sign-level annotations. This is the largest publicly available CSLR test benchmark. We also propose a strong baseline in the form of a multi-task Transformer network that learns with weak supervision and outperforms strong baselines on our new CSLR benchmark. We hope that BOBSL-CSLR-TEST encourages future research on large-vocabulary CSLR and modeling non-lexical signing classes such as pointing and gesture-based signs.

Societal impact. The CSLR task can have positive implications by bridging the gap between spoken and sign languages. It can enable more seamless communication, content creation and consumption by breaking down the language barriers that are prevalent today. At the same time, the ability to automatically recognise

signs and search a large volume of signing videos can lead to risks such as surveillance of signers. We believe that the positives outweigh the negatives.

Acknowledgements. This work was granted access to the HPC resources of IDRIS under the allocation 2023-AD011013569 made by GENCI. The authors would like to acknowledge the ANR project CorVis ANR-21-CE23-0003-01. The work was funded by a Royal Society Research Professorship RP\R1\191132.

APPENDIX

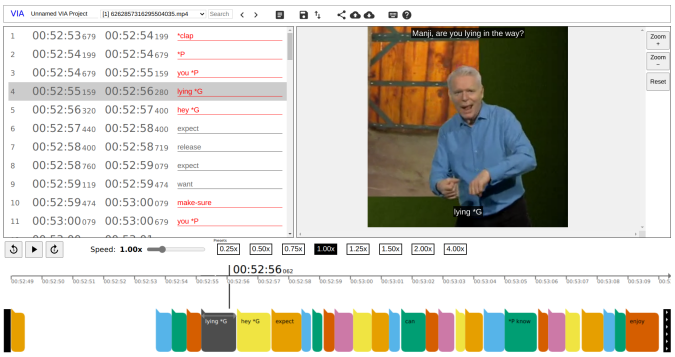
8.7 BOBSL-CSLR: Annotation

In this section, we provide details on the annotation collection procedure for BOBSL-CSLR.

In Fig. A.1, we show two screenshots of the web-based annotation tool used for collecting BOBSL-CSLR. It is based on the VIA video annotation software [Dutta and Andrew Zisserman 2019; Renz et al. 2021a; Renz et al. 2021b]. Specifically, we start from VIA-SLA[3] and extend it for our purposes (by adding the signing-aligned subtitle, adjusting the dropdown vocabulary, and removing the Top-1 sign class prediction from a sign recognition model).

Annotators are provided with a video sequence with 20 signing-aligned subtitle sentences spanning roughly 2-3 minutes (this takes about 1 hour to annotate densely). The subtitles are displayed at the top of the video. Annotators are given an initial set of sign intervals (shown at the bottom of the tool with a colored timeline) from an automatic sign segmentation method [Renz et al. 2021b], but are instructed to refine these sign boundaries (or remove/add sign intervals) if necessary. For each sign interval, the annotators enter free-form text which best corresponds to the meaning of the sign, by taking into account context of the surrounding video. The annotators are encouraged to prioritise assigning words that appear in the corresponding subtitle. For example, in Fig. A.1a, the signing-aligned subtitle is 'Manji, are you lying in the way?' and the highlighted gray interval is annotated with the subtitle word 'lying'. We further show a drop down menu for each sign (with the full English vocabulary of the subtitle words in the

[3] https://github.com/RenzKa/VIA_sign-language-annotation



(a) Final sign annotations

(b) Zoom on dropdown menu before any entry

Figure A.1: **Annotation tool:** We show two screenshots of the web based annotation tool used for collecting BOBSL-CSLR. **(a)** The annotators use (i) the right-hand side of the tool to view the signing video and aligned English subtitle, (ii) the colored timeline at the bottom of the tool to adjust/add/remove sign intervals, (iii) the left-hand side to write free-form text to assign English word(s) to sign intervals, as well as the sign type (for example, *P for pointing) when appropriate and (iv) a dropdown menu for each sign with the vocabulary spanned by subtitle words in BOBSL. We note that on the bottom colored timeline, some sign intervals appear as if unlabelled simply because there is not enough space to fit the annotation text (however these are shown on the left hand side of the tool). The segments are first initialised from automatic sign segmentation [Renz et al. 2021b] and are refined by annotators. **(b)** When no annotation is entered yet, subtitle words are displayed at the top of the drop down menu.

BOBSL dataset, together with a search functionality), with the subtitle words listed at the top of the list, to enable faster annotation (see Fig. A.1b). Besides assigning the English word(s) corresponding to a sign, the annotators indicate the *type* of sign when appropriate. For example, as shown on the left of Fig. A.1a, the third sign is annotated as 'you *P' meaning the sign for 'you' is a **p**ointing sign, and the fifth sign is annotated as 'hey *G' meaning the sign for 'hey' is a **g**esture sign. The full list of sign types annotated is given in the following section.

Quality control and error correction. The annotations were provided by two native BSL signers. In the early stages of the annotation, each video sequence was labelled by one annotator, and then checked (and modified if necessary) by the other. The annotations were also reviewed by a subset of the authors on this paper to ensure consistency and high quality.

We note that initially the annotators heavily relied on pre-estimated sign boundaries, and mainly entered words for each segment. This resulted in numerous ambiguous (misaligned) segments in between signs which were labelled as unknown (*U), leading to a sparse timeline. After reviewing the initial set of annotations and discussing with the annotators (using sign interpreters for the meetings), the

instructions were refined to ensure a dense timeline: by adjusting the segment boundaries if necessary, including all the various sign types in additional to the lexical signs, and prioritizing subtitle words to resolve some ambiguities.

8.8 BOBSL-CSLR-test Dataset Analysis and Statistics

In this section, we provide further dataset statistics.

Sign types. As mentioned in Sec. 8.3 of the main paper, in addition to lexical signs that have an English word equivalent, a wide range of sign types are annotated. This is mainly motivated by signs that cannot easily be associated to lexical words: pointing signs or depicting signs are such examples.

In Tab. A.1, we present the different sign type annotations collected and their occurrence in BOBSL-CSLR-TEST. We note that for the CSLR evaluation, we only use signs that are associated to words (or phrases), filtering out the non-lexical special sign types since these are outside the scope of this work. We highlight that unknown, i.e. unrecognisable signs denoted by *U, are also filtered out for the CSLR evaluation since they are not associated to a lexical word. In Fig. 8.2 of the main paper, we show several examples per sign type.

We are also aware that our *P pointing annotations' definition differs from standard pointing as seen in the linguistic literature (e.g. BSL-Corpus [Schembri et al. 2017]). Annotators are encouraged to mark lexical signs that contain the pointing *handshape* (such as the sign for 'rocket' which is signed with index finger pointing upwards) even when they are not pronominal reference [Cormier et al. 2013]. This can potentially be beneficial in future work to recognise the role of pointing in context. Our pointing annotations are therefore a superset of the linguistic definition. We approximate the linguistic pointing annotations containing locative pointing or pronominal reference [Cormier et al. 2013] (shown in parenthesis in Table 8.2) by counting pointing annotations (i) without any lexical word associated to it or (ii) annotations associated to words in the following list: 'me', 'this', 'I', 'here', 'in', 'that', 'you'. These words were determined by inspecting the most frequent words assigned to *P signs. We refer to this subset of *P signs as *P1, and the rest as *P2.

Table A.1: **Sign types:** We present the different sign type annotations collected and their occurrence in BOBSL-CSLR-TEST. We also provide a brief description of each sign type and their annotation short form. See Fig. 8.2 of the main paper for visual examples. Note we add in parenthesis, the subset of pointing annotations referred to as *P1 in Section 8.8, to approximate the linguistic pointing such as pronominal referencing.

Type of partially-lexical or non-lexical signs	Description	Short form	#occurences in BOBSL-CSLR-TEST
Name sign	countries or people that have a well-known BSL sign	*N	75
Facial expression sign	meaningful facial expression	*FE	252
Slang sign	sign corresponding to slang word	*S	258
Timeline sign	signs which relate to time (present, past, future, calendar units, action continuity)	*T	324
Unknown sign	unrecognisable signs	*U	680
Gestures	meaningful body poses that non-signers also do	*G	876
Fingerspelling	signing letters of the BSL alphabet to spell a word	*FS	1340
Depicting signs	signs which describe shape, size, movement or handling	*D	2569
Pointing	sign with index finger or flat hand	*P	(2477 *P1) 4179
Total			11K

Annotation duration. In Fig. A.2, we show three plots on the duration of BOBSL-CSLR-TEST annotations. First, the left plot shows the histogram of durations: we see that a large portion of the annotations last between 0.4 and 0.6 seconds, which is in accordance to the estimates of previous works [Buehler et al. 2008]. Second, the middle plot shows the duration distribution per sign type; we see that fingerspelled signs (*FS) last longer than other sign types, potentially because fingerspelling involves multiple handshapes (one per letter) to express a word. On the other hand, pointing signs (*P1) seem to have shorter duration than others. Third, the right plot shows the proportion of sign types per duration, and we see that purely lexical signs dominate other annotation types. We also observe that the proportion of fingerspelled and depicting signs increases with longer duration.

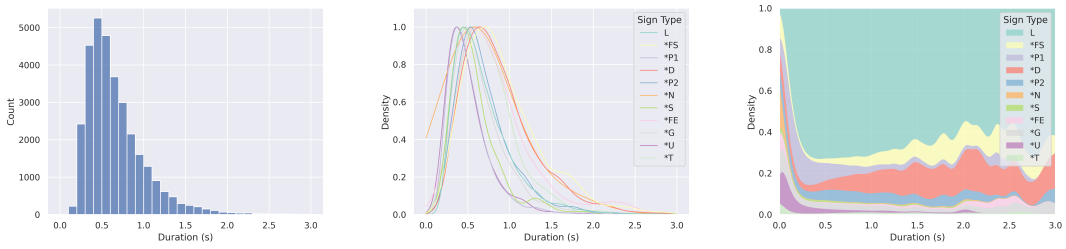


Figure A.2: **Statistics on BOBSL-CSLR-test annotations' duration:** We show **(left)** the histogram of durations for all annotations; **(middle)** the kernel density estimation of duration for each annotation type; and **(right)** the distribution over annotation types for different durations. L denotes purely lexical signs. See text for interpretation.

8.9 CSLR^2 Implementation Details

In this section, we provide implementation details for our training strategy and the model architecture.

8.9.1 Sources of supervision

Leveraging weak and noisy text labels for the CSLR and retrieval training constitutes one of the key contributions of this work. Next, we present our two sources of text supervision, namely, sign-level pseudo-labels and sentence-level weakly-aligned subtitles.

Sign-level pseudo-labels. We start with (V, T) video-subtitle pairs that do not contain sign-level annotations. In order to obtain sign-level supervision to train for CSLR, we perform sign-level pseudo-labelling. Specifically, we apply an ISLR model in a sliding window fashion with a stride of 2 frames, and perform post-processing of sign predictions as an attempt to reduce noise. Our post-processing strategy consists of 3 steps: (i) we first combine confidence scores of synonym categories for the Top-5 predictions from the ISLR model (using the synonym list defined in [Momeni et al. 2022]); (ii) we then filter out low confidence predictions (below a threshold value of $\theta = 0.6$); (iii) finally, we remove non-consecutive predictions – since each sign spans several video frames, we expect repetitions from the ISLR model (we keep predictions with at least $m = 6$ repetitions). We note that this same post-processing strategy is used during CSLR inference (see Section 8.4.1 of the main paper).

Weakly-aligned subtitles. The source of our large-scale video-subtitle pairs is

from sign language interpreted TV shows, where the timings of the accompanying subtitles correspond to the audio track, but not necessarily to signing [Albanie et al. 2021b]. For better sign-video-to-text alignments, we use automatic signing-aligned subtitles from [Bull et al. 2021] (described in [Albanie et al. 2021b]) to train our models. We restrict our training to subtitles spanning 1-20 seconds, resulting in 689K video-subtitle training pairs.

Table A.2: **Model summary:** F is the maximum number of frames for all video sequences in a batch of size B. Zero-padding is applied to inputs. W is the maximum number of tokens obtained after tokenisation of B sentences. Zero-padding is also applied here.

Notation	Description	Input	Output
$\mathcal{V}_{enc}^{\text{Sign}}$	Video-Swin-Tiny (across T clips)	$T \times 16 \times 224 \times 224 \times 3$	$T \times 768$
$\mathcal{V}_{enc}^{\text{Sent}}$	Transformer Encoder with 6 encoder layers, 8 attention heads and 768 hidden dimensions	$B \times F \times 768$	$B \times F \times 768$
$\mathcal{T}_{enc}$	Pre-trained T5 [Raffel et al. 2020] ("t5-large") encoder	$B \times W$	$B \times 1024$ (sentence-level) $B \times W \times 1024$ (word-level)
FC	Two linear projections from 768-d space to 256-d space	$B \times F \times 768$	$B \times F \times 256$
	Two linear projections from 1024-d space to 256-d space	$B \times F \times 1024$	$B \times F \times 256$

8.9.2 Architecture

Model summary is displayed in Tab. A.2 (please also refer to Fig. 8.3 of the main paper).

Sign video encoder ($\mathcal{V}_{enc}^{\text{Sign}}$). Similar to [Prajwal et al. 2022b], our sign video features are obtained by training a Video-Swin model [Z. Liu et al. 2022], on ISLR. Specifically, we finetune the Video-Swin-Tiny architecture on ISLR, starting from Kinetics-400 pre-trained weights[4], using automatic annotations released in [Momeni et al. 2022]. These annotations provide individual sign labels along with timestamps of where they occur in the video. Note that the annotations have been *automatically* obtained with the help of subtitles (by exploiting cues such as mouthing), and can thus be noisy.

In practice, the network ingests a short video clip (16 frames, $<$ 1 second) and outputs a single vector $\hat{\mathbf{v}} \in \mathbb{R}^d$ ($d = 768$), followed by a classification head to recognise isolated signs. The input videos are of resolution 224×224, with

[4] https://github.com/SwinTransformer/Video-Swin-Transformer

the pixel values normalized by the ImageNet mean and standard deviation as also done in [Z. Liu et al. 2022]. We only perform two data augmentations: (i) random horizontal flipping, (ii) random color jitter by scaling the original pixel values by a random factor between $[0.8, 1.2]$. We train for ISLR with a batch size of 256 distributed across 4 NVIDIA A40 GPUs over a period of 3 days. We use the Adam optimizer with the default parameters, except for the learning rate. The learning rate is linearly warmed up to a maximum of 1×10^{-3} over 8000 iterations, followed by a cosine decay schedule. We stop training when the model's validation accuracy (Top-1 and Top-5) does not improve for 2 epochs.

Once trained for ISLR, we freeze the parameters of this relatively expensive backbone, and extract isolated sign video embeddings $\hat{\mathbf{v}}$ in a sliding window manner with a stride of 2 frames. Note we use RGB-based embeddings, instead of body keypoint estimates, due to their more competitive performance in the large-vocabulary setting, where sign differences are subtle and nuanced [Albanie et al. 2021b].

Sentence video encoder ($\mathcal{V}_{enc}^{\mathbf{Sent}}$). We adopt a Transformer encoder architecture, similar to BERT [Devlin et al. 2018], with 6 encoder layers, 8 attention heads and 768 hidden dimensionality. It ingests the sign sentence video as isolated sign video embeddings $\{\hat{\mathbf{v}}_f\}$ and outputs context-aware sign video embeddings $\{\mathbf{v}_f\}$. We obtain a single sentence video embedding by simply max-pooling over the temporal dimension, i.e. $\mathbf{V} = \text{MaxPool}_f(\{\mathbf{v}_f\}_{f=1}^{F})$, with F video features, and experimentally validate this choice. We restrict our training to video clips shorter than 20 seconds. The parameters of the Transformer encoder are learnt using the sentence retrieval loss $\mathcal{L}_{\text{SentRet}}$ between sentence text T and video V embeddings, and the sign retrieval loss $\mathcal{L}_{\text{SignRet}}$ between sign text embeddings $\mathbf{t}_w$ and the corresponding sign video embeddings $\mathbf{v}_f$, as described in Sec. 8.4.2.

Text encoder ($\mathcal{T}_{enc}$). We use the encoder part of a pre-trained T5 [Raffel et al. 2020] (`t5-large`), and keep its weights frozen. Note we do not use its decoder. The output text embeddings have dimensionality 1024.

Projection heads. We additionally learn projection layers, mainly to reduce the joint embedding dimensionality to 256 before contrastive loss computations. Specifically, we have a total of 4 projection heads: two for reducing the *text* dimensionality ($1024 \rightarrow 256$) with a separate projection for sign categories and sentences, two for reducing the *video* embedding dimensionality ($768 \rightarrow 256$) separately for

sign and sentence video embeddings.

CSLR^2 training details. We perform the contrastive training with batch size 512 and learning rate lr $= 5 \times 10^{-5}$ using Adam [Diederik P Kingma and Ba 2015] optimizer. The HN-NCE loss uses temperature $\tau = 0.07$, $\alpha = 1.0$ and $\beta_{\text{SignRet}} = 0.5$, $\beta_{\text{SentRet}} = 1.0$. We use $\lambda_{\text{SignRet}} = 0.0075$, $\lambda_{\text{SentRet}} = 0.90$ to weigh the different terms in our joint loss. Training takes less than 20 hours on 4 NVIDIA 32 GB V100 GPUs. We choose the model checkpoint that performs best on the validation set.

Data augmentation is performed on both the video and text side. On the text side, random words from the subtitles are dropped using two probabilities. First, we choose with probability $p_{\text{sub,drop}} = 0.8$ whether words will be dropped in a given subtitle. Second, in subtitles selected by the previous step, we drop each word with probability $p_{\text{word,drop}} = 0.4$. On the video side, we perform dropping in a similar fashion to dropping words in subtitles, and drop features in sequences with two probabilities: $p_{\text{seq,drop}} = 0.8$ and $p_{\text{frame,drop}} = 0.5$. We also temporally shift subtitles with a random offset, which is uniformly sampled in the range ± 0.5 seconds.

8.10 Experiments

8.10.1 CSLR baselines

In Tab. A.3, we provide a more detailed version of Tab. 8.4 of the main paper. In particular, we give results with/without our post-processing filtering † on ISLR-based methods from [Albanie et al. 2021b; Momeni et al. 2022; Prajwal et al. 2022b], and show the impact of thresholds on spotting-based methods from [Albanie et al. 2021b; Momeni et al. 2022] that use subtitles. We see that without the post-processing heuristics, the ISLR-based methods and our CSLR-based method do not obtain reasonable results, mainly due to the noise from many false positives. Our model performs the best both in the settings (i) with and (ii) without post-processing, amongst all the approaches that do not use subtitle-cues.

Table A.3: **Analysis on CSLR baselines:** † denotes the optimal subset of the annotations that gives the highest performance. The values without † are obtained using the raw model predictions and collapsing continuous repetitions. We observe that performing post-processing is essential to obtain reasonable results. Our model gives a clear improvement over the current best CSLR method.

	WER ↓	mIOU ↑
Sub-based annots. [Albanie et al. 2021b] [$M_{.5}D_{.7}A_{.0}$]	115.8	13.0
Sub-based annots. [Albanie et al. 2021b] [$M_{.8}D_{.8}$]	93.8	7.1
Sub-based annots. [Momeni et al. 2022] [$M^*_{.8}D^*_{.8}A_{.0}P_{.5}$E N]	90.7	20.0
Sub-based annots. [Momeni et al. 2022] [$M^*_{.8}D^*_{.8}A_{.0}P_{.5}$]	85.5	18.1
Sub-based annots. [Momeni et al. 2022] [$M^*_{.7}D^*_{.9}A_{.4}P_{.2}$]	85.9	15.3
ISLR I3D-2K [Albanie et al. 2021b]	453.5	8.6
ISLR I3D-2K [Albanie et al. 2021b] †	82.5	18.1
ISLR I3D-8K [Momeni et al. 2022]	343.6	14.9
ISLR I3D-8K [Momeni et al. 2022] †	74.7	27.0
ISLR Swin-8K [Prajwal et al. 2022b]	293.9	17.6
ISLR Swin-8K [Prajwal et al. 2022b] †	71.7	30.1
CSLR^2 (Ours)	201.4	22.4
CSLR^2 (Ours)†	**65.5**	**35.5**

8.10.2 Oracle CSLR performance

We conduct an oracle experiment to determine an upper-bound for: "What is the best performance we can obtain if we remove all the false positives predictions from our model after post-processing?". To do this, we compare the post-processed predictions of our best model with the words in the ground-truth sequence and remove the ones that are false positives. In Tab. A.4, we see that we can obtain a large improvement in scores, but the WER is still relatively high. This indicates that our model should recognise more signs in the sequences for it to lower the WER even more. We also note that about 25% of the test set glosses are out-of-vocabulary for the model. Thus, increasing the vocabulary coverage of the model can substantially boost the recognition performance.

8.10.3 CSLR performance breakdown and analysis

In Tab A.5, we report the CSLR performance breakdown by excluding specific sign types from both ground truth and prediction *timelines*, i.e. removing corresponding frames. We further examine the oracle performance by assuming we perfectly recognise the words that correspond to certain sign types, such as pointing and

Table A.4: **Oracle analysis:** We compute oracle results by removing all false positives (i.e. words in the post-processed predicted sequence that are not in the ground-truth sequence). We see that despite removing false positives, the WER is still relatively high, implying that we either miss words in the ground-truth sequence or we predict them at the incorrect location.

	WER ↓	mIOU ↑
CSLR^2 (Ours)	65.5	35.5
Oracle	61.0	42.3

Table A.5: **CSLR performance breakdown:** We experiment with (i) filtering out certain annotation types from both ground truth and prediction timelines (**Filter** column), and (ii) replacing predictions with ground truth labels for these annotations (**Oracle** column). We observe that pointing and fingerspelling, in particular, contribute to a significant WER reduction (-14.8).

	Filter		Oracle	
Sign Type	WER ↓	mIOU ↑	WER ↓	mIOU ↑
CSLR^2 (Ours)	65.5	35.5	-	-
*FS	64.9	36.5	54.4	48.0
*P	65.0	35.7	56.0	47.3
*P/*FS	64.2	36.9	50.7	50.6
all *	64.3	37.0	49.3	51.7

fingerspelling. Specifically, we investigate the impact of (i) all * annotations, (ii) pointing *P annotations, and (iii) fingerspelling *FS annotations. We see that in an ideal scenario where pointing or fingerspelled annotations are perfectly predicted, the overall CSLR performance is substantially improved, leading to a significant WER reduction of -14.8 (65.5 vs 50.7), and even more if all * sign types are correctly recognised (49.3).

8.11 CSLR^2 Ablations

In this section, we provide complementary ablations to what is done in Sec. 8.5. of the main paper. We first look at the impact of changing the frozen text backbone, from pre-trained T5 model to pre-trained MPNet [Song et al. 2020] (Sec. 8.11.1). Next, we show the impact of evaluating with synonyms at test time on CSLR performance (Sec. 8.11.2). We then provide ablations on: (i) loss weights (Sec. 8.11.3), (ii) and subtitle alignment (Sec. 8.11.4).

Table A.6: **Impact of the text encoder:** We experiment with the choice of the text encoder in our model. More specifically, we compare T5 (`t5-large`) [Raffel et al. 2020] and MPNet (`all-mpnet-v2`) [Song et al. 2020]. We observe gains on all reported metrics by opting for T5 over MPnet.

Text Backbone	SentRet	SignRet	CSLR WER ↓	CSLR mIoU ↑	CSLR F1@{0.1, 0.25, 0.5} ↑		
ISLR Baseline			71.7	30.1	40.0	37.9	27.2
MPNet	✓	✓	66.3	34.9	46.4	45.3	36.3
T5	✓	✓	**65.5**	**35.5**	**47.1**	**46.0**	**37.1**

Table A.7: **Impact of synonyms at test time:** We observe major gains on all reported metrics when considering synonyms as correct for CSLR evaluation. This shows that having a precise and exhaustive list of synonyms is important for CSLR.

Model	Synonyms	CSLR WER ↓	CSLR mIOU ↑	CSLR F1@{0.1, 0.25, 0.5} ↑		
ISLR Baseline	✗	78.0	22.6	32.6	30.9	21.6
ISLR Baseline	✓	71.7	30.1	40.0	37.9	27.2
$CSLR^2$ (Ours)	✗	73.3	26.5	37.8	36.8	29.4
$CSLR^2$ (Ours)	✓	**65.5**	**35.5**	**47.1**	**46.0**	**37.1**

8.11.1 Text encoder

In Tab. A.6, we experiment with replacing the frozen T5 [Raffel et al. 2020] text encoder with MPNet [Song et al. 2020]. We first see that opting for T5 instead of MPNet yields performance gains on all reported metrics. Moreover, we observe that our joint training helps, for both text encoders, improving over the strong ISLR Baseline for CSLR.

8.11.2 Synonyms for CSLR

We have shown through a series of qualitative examples the importance of having a more comprehensive synonyms list in the evaluation process. Here, we look at the impact of considering synonyms as correct when evaluating for CSLR in terms of performance. In Tab. A.7, we observe major gains on all reported metrics, when considering synonyms as correct, despite our synonyms list being far from perfect as explained in our qualitative analysis in the main paper.

8.11.3 Weighting of losses on CSLR performance

We study the impact of loss weighting on CSLR performance. More specifically, in Fig. A.3, we plot WER performance obtained by jointly training with our two losses

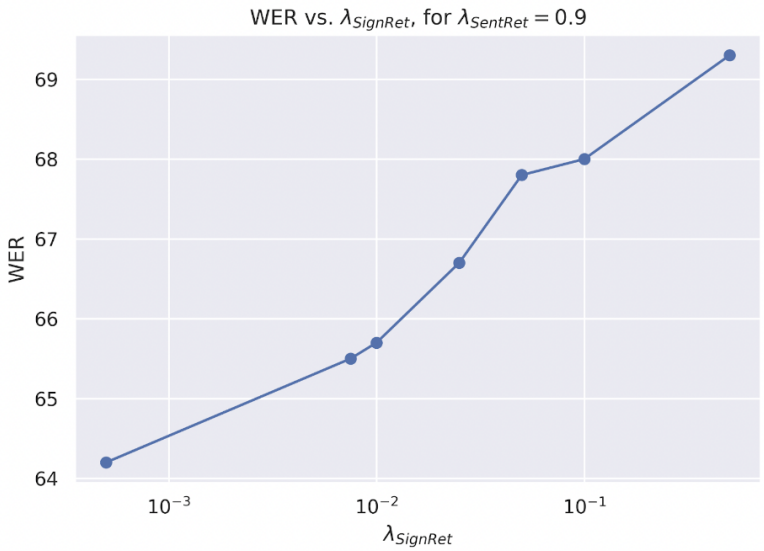


Figure A.3: **Weighting of losses on CSLR performance:** We look at the impact of changing λ_{SignRet} for fixed $\lambda_{\text{SentRet}} = 0.9$. We observe that WER decreases with lower λ_{SignRet} values.

Table A.8: **Subtitle alignment:** Using automatic signing-aligned subtitles [Bull et al. 2021] instead of audio-aligned subtitles improves CSLR performance.

	BOBSL-CSLR-test				
Subtitles	WER ↓	mIOU ↑	F1@{0.1, 0.25, 0.5} ↑		
Audio-aligned	68.0	33.5	44.5	43.3	33.5
Signing-aligned [Bull et al. 2021]	**65.5**	**35.5**	**47.1**	**46.0**	**37.1**

(SentRet and SignRet) against varying loss weights for the sign-level loss, fixing the weight given to the sentence-level loss. Surprisingly, we observe that giving less weight to the CSLR-relevant SignRet loss actually brings improvement to CSLR. We see that WER decreases with lower λ_{SignRet} values. A potential explanation could be that the SentRet loss regularizes the training, as we observed SignRet alone is prone to overfitting.

8.11.4 Subtitle alignment

In Tab. A.8, we measure the impact of using audio-aligned subtitles instead of automatic signing-aligned subtitles [Bull et al. 2021]. We observe a notable gain in CSLR performance: this is expected as audio-aligned subtitles are not necessarily aligned with the signed content in videos, but instead with the audio soundtrack accompanying the original TV broadcasts.

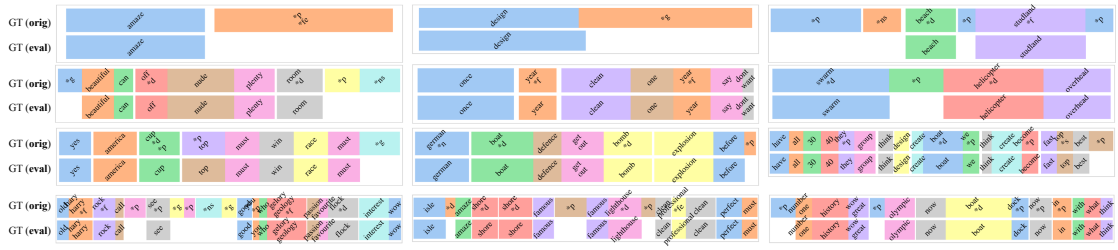


Figure A.4: **Ground truth processing:** We provide examples of the original ground truth annotations as collected by annotators (GT orig), and the version we use at evaluation time (GT eval).

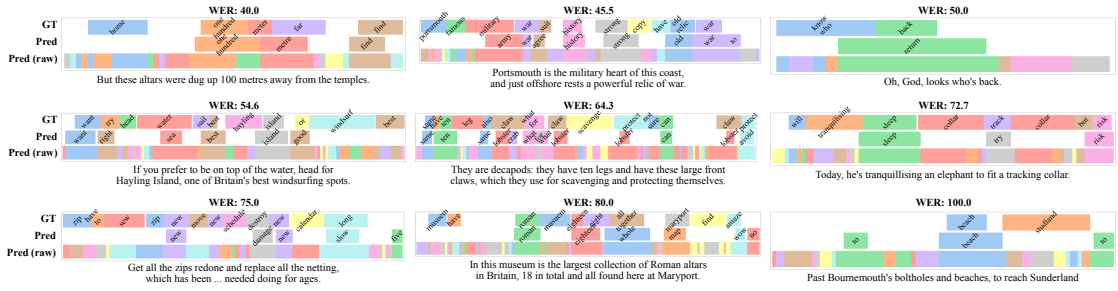


Figure A.5: **Effect of post-processing the predictions:** We show that processing the frame-level sign recognitions into segment-level ones (as explained in Sec. 8.9.1 of the main paper) effectively cleans up a large portion of the noise. For each sample, top row corresponds to the ground truth (GT), middle row to post-processed CSLR predictions from CSLR^2 model, and bottom to unfiltered raw predictions.

8.12 Qualitative Examples

Ground truth processing. We show in Fig A.4 several sequences illustrating how we obtain the ground truth for evaluation, in particular with the removal of * annotations and keeping only the words associated to each annotation.

Effect of post-processing the predictions. We show in Fig A.5 qualitative results with and without the post-processing procedure described in Sec. 8.9.1. These examples particularly highlight the importance of the post-processing we propose in our method.

Additional results. We show qualitative examples of CSLR performance in Fig. A.6. For each signing video sample, we show the ground truth and predicted glosses, along with the English subtitle. We also show the WER and IOU per sample. We observe that our model is able to correctly predict an important proportion of the ground truth annotations.



Figure A.6: **Additional qualitative CSLR results.** We provide additional examples for our model's CSLR predictions (bottom row per sample) on BOBSL-CSLR-TEST, as well as the the ground truth (top row per sample). As in Fig. 5.3 of the main paper, we display WER for each example, along with the corresponding subtitle to provide context for the reader.

Part IV

Learning Co-speech Gesture Representations

Chapter 9

Understanding Co-speech Gestures in-the-wild

This paper has been accepted for publication at the International Conference on Computer Vision (ICCV), 2025.

Understanding Co-speech Gestures in-the-wild

Sindhu Hegde K R Prajwal Taein Kwon Andrew Zisserman
Visual Geometry Group, University of Oxford
{sindhu,prajwal,taein,az}@robots.ox.ac.uk

Abstract

Co-speech gestures play a vital role in non-verbal communication. In this paper, we introduce a new framework for co-speech gesture understanding in the wild. Specifically, we propose three new tasks and benchmarks to evaluate a model's capability to comprehend gesture-text-speech associations: (i) gesture-based retrieval, (ii) gestured word spotting, and (iii) active speaker detection using gestures. We present a new approach that learns a tri-modal speech-text-video-gesture representation to solve these tasks. By leveraging a combination of global phrase contrastive loss and local gesture-word coupling loss, we demonstrate that a strong gesture representation can be learned in a weakly supervised manner from videos in the wild. Our learned representations outperform previous methods, including large vision-language models (VLMs), across all three tasks. Further analysis reveals that speech and text modalities capture distinct gesture-related signals, underscoring the advantages of learning a shared tri-modal embedding space. The dataset, model, and code are available at: https://www.robots.ox.ac.uk/~vgg/research/jegal.

9.1 Introduction

Humans gesture when they talk – gesturing is an integral part of human communication, together with speech and facial expressions. Gestures can vary from *beats*

– two phase hand movements (up/down, left/right etc) that emphasize particular words or phrases and match the rhythm of the speech, but do not carry semantic content – to *iconic* and *deictic* gestures that are representational and illustrate the *content* of the speech [Andric and Small 2012; McNeill 1992]. For example, hands and arms moving apart can accompany a speech segment indicating that something is "huge", or as illustrated in Fig 9.1, an inward pointing gesture to depict the uttered word "my".

Non-verbal communication accounts for 55% of overall communication[1], highlighting the need for machines to understand non-verbal gestural elements in order to have a holistic understanding of human communication. A clear application is enriching human-computer interaction (HCI) through gestures, and this requires machines to comprehend the semantics of the user's hand gestures. Another application is to detect if a person is speaking based on their gestures, or spot specific words or phrases in a video based on gestures alone. More generally, being able to recognize gestures and determine their semantic and temporal alignment with speech enables human communication to be studied at scale [Kendon 2004].

In this paper, our objective is to learn and evaluate co-speech gesture representations. To this end, we propose three tasks and evaluation benchmarks that act as a proxy for assessing real world applications: (1) gesture-based cross-modal retrieval, (2) gesture word spotting, and (3) active speaker detection via gestures. We perform large-scale training featuring $\approx$ 7000 speakers and evaluate on in-the-wild videos from the AVSpeech dataset [Ephrat et al. 2018].

All three tasks require models to learn to associate gesture clips with speech segments or their corresponding textual transcripts. For this, we propose a model termed **J**oint **E**mbedding space for **G**estures, **A**udio, and **L**anguage (**JEGAL**)[2] that facilitates matching gestures to words and phrases in the accompanying speech. The matches can be based on the style of the speech (intonation, stress, prosody) or the semantic content of the phrase or particular words. However, learning a rich joint gesture-audio-language embedding space is a very challenging task. The associations between gestures and speech are typically sparse and ambiguous, with a high degree of variability across speakers. Usually, only a few of the spoken

[1] https://online.utpb.edu/about-us/articles/communication/how-much-of-communication-is-nonverbal/

[2] JEGAL, known as Zhuge Liang in Chinese, was a prominent historical figure from China's Three Kingdoms period and is regarded as a symbol of wisdom.



Figure 9.1: Co-speech gestures supplement the spoken language – we show examples for three phrases here, with common words. Learning to associate gestures with the uttered phrases is essential for a holistic understanding of human communication.

words are clearly gestured. Additionally, the same sentence can be gestured very differently in different contexts and by different people. Gestures also depend on the speaker's emotion, culture, and social scenario (formality, private vs. public, with friends or strangers, etc.). Furthermore, some types of gestures, such as beat gestures, carry no semantic information, resulting in no direct mapping from the gesture to words. The sparse and weak cross-modal correlations makes gesture representation learning a very unique research problem.

We make three key design choices in our approach that result in a strong tri-modal gesture representation. To start with, we learn gesture video representations from large-scale *weak* cross-modal supervision. The supervision is weak because we only use phrase-level speech audio and transcripts – since we do not have any information on which words in the speech are gestured for videos collected 'in-the-wild'. Second, we obtain cross-modal supervision in the form of both audio and the corresponding text transcript (in Section 9.6.1, we demonstrate that speech and text modalities capture complementary gesture-related signals). Third, we introduce a new gesture-word alignment and spotting loss that strongly encourages learning of word-level correspondences.

To summarize, we make the following contributions: (i) we propose a new framework for co-speech gesture understanding with three new tasks and evaluation benchmarks; (ii) we learn a joint tri-modal embedding space in a weakly-supervised manner with a combination of global phrase-level objective, and a local word-level gesture coupling loss; (iii) we demonstrate that the learned JEGAL representation outperforms previous vision-language foundation models on the three

gesture-centric tasks and is useful for practical applications.

9.2 Related Work

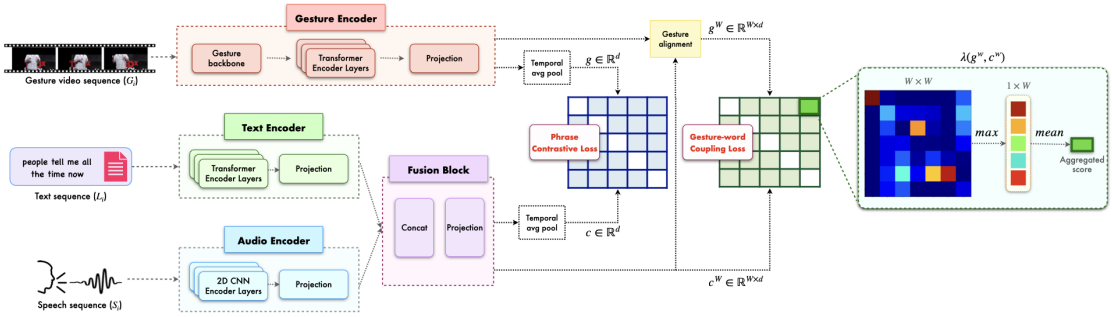


Figure 9.2: The **JEGAL** architecture. The three input modalities (video, text, speech) are each encoded with a modality-specific encoder, followed by a fusion block to merge speech and text representations. The encoder outputs are average-pooled to obtain global (phrase-level) gesture and speech-text embeddings. During training, these provide the inputs for the global 'phrase contrastive loss'. The gesture alignment module aggregates the relevant video frames to obtain a local word-level gesture embedding for each speech-text word. During training, these provide the inputs for the local 'gesture-word coupling loss'. The two losses encourage the learning of global and local correspondences between the three modalities.

Spectrum of Human Gestures. Gestures can be classified broadly into four major classes [McNeill 1992]. Emblematics convey a clear symbolic meaning, e.g. a "thumbs-up". Iconic gestures are used to convey meaning co-occurring with the articulated speech, e.g. "revolving door" is accompanied with the hands moving in a circular motion. Deictic gestures are pointing gestures using the index finger. The most common are the "beat" gestures which are co-speech gestures that are temporally aligned with the prosodic characters of human speech. They are prominent on lexically stressed syllables. Past works [Wagner et al. 2014; Kendon 2004] have studied how these gesture classes relate to the speech. Several papers have been attempted to recognize hand, head, and facial emblematic gestures [Freeman and M. Roth 1995; Darrell et al. 1996]. Recognizing deictic gestures can help in identifying the referring object being pointed to in a conversation, an essential part of human-robot interaction [Matuszek et al. 2014]. The other two gesture classes, i.e., beat and iconic, are quite diverse and are difficult to associate with a well-defined set of words or hand movements. This work takes a data-driven approach to learn gesture representations with the help of speech and natural language.

Co-speech Gesture Understanding. Spoken discourse consists of multiple

streams of information: language, lip movements, facial expressions, and hand gestures. As opposed to the advancements in the other streams [Radford et al. 2022; Prajwal et al. 2022a; Barrault et al. 2023b], co-speech gesture understanding is a relatively under-explored area. One possible reason for this could be that gestures are only sparsely correlated with the speech. As a result, some of the works have resorted to developing models for specific cases where the gestures are clear, e.g. weather narration [Rajeev Sharma et al. 2000; Kettebekov et al. 2003]. Some works [Molchanov et al. 2016; Morency et al. 2007; Ghaleb et al. 2024] have tried to detect and recognize gestures in laboratory settings while using multiple stereo and IR cameras. Recently, GestSync [Sindhu B Hegde and Andrew Zisserman 2023] learns gesture representations by solving for cross-modal synchronization with speech. This objective can lead to the model capturing low-level associations rather than high-level semantics, leading to poor performance on tasks like retrieval, and word spotting. Our work is the first one to learn gesture representations which capture the semantics, style and also learn word-level associations.

Understanding Gestures in Sign Language. Sign language understanding and recognition is another body of work where models need to understand and associate gestures to words and phrases to solve tasks like sign recognition [Albanie et al. 2021b; Prajwal et al. 2022b; Raude et al. 2024; Zuo and Mak 2022; Min et al. 2021; H. Zhou et al. 2020b; N. C. Camgoz et al. 2020b], sign language retrieval [Y. Cheng et al. 2023; Duarte et al. 2022], sign language translation [Tarrés et al. 2023; A. Yin et al. 2023; N. C. Camgoz et al. 2018; N. C. Camgoz et al. 2020b] and sign language production [Stoll et al. 2020; Saunders et al. 2020; Saunders et al. 2022]. Sign language gesture understanding is quite different compared to co-speech gesture understanding. In sign language, text transcription is a *translation* of what is being signed/gestured. Thus, the words in the text can be a summary or paraphrasing of what is being gestured, with even a mismatch in the temporal ordering. In co-speech gestures, the speaker is the gesturer, and the gestures are being made by the speaker to directly accompany each word he (she) utters. Hence, these two tasks require very different approaches.

Co-speech Gesture Generation. Several works have focused on generating natural gestures that match a given speech segment. This task has the advantage of being "freely supervised" – large-scale datasets can be curated for this

task with almost no manual effort as it only requires unlabeled videos of people talking. Speech2Gesture [Ginosar et al. 2019] trains speaker-specific models to generate hand skeleton motion for a given speech segment. Recent papers [Yoon et al. 2020; Zhi et al. 2023; Ao et al. 2023] have moved towards more speaker-independent approaches while also using text to obtain strong semantic supervision. In particular, GestureDiffuCLIP [Ao et al. 2023] learns a joint gesture-text embedding to improve gesture generation. As will be seen in the results, one clear distinction from the work presented in this paper is that [Ao et al. 2023] does not learn word-level correspondences, which makes a significant difference to the gesture understanding tasks that we evaluate.

Gesture Recognition Datasets. ChaLearn ConGD and IsoGD [Wan et al. 2016] are two gesture recognition datasets, providing benchmarks for the ChaLearn challenges. However, these datasets are of people using gestures for a task, (e.g. playing a game or controlling an appliance), and are not suitable for learning or evaluating co-speech gestures. Montalbano II [Escalera et al. 2013] is another dataset covering gestures from a vocabulary of 20 Italian sign gesture categories. Again, this is not suitable for our task.

Learning Video Representations. Representation learning in videos [Tong et al. 2022; L. Wang et al. 2023; Oquab et al. 2023] has gained significant attention driven by the availability of large-scale video datasets. Learning video representations from text offers a promising advantage by incorporating interpretability via language. Recent works on vision language representation learning [Radford et al. 2021; Alayrac et al. 2022; Junnan Li et al. 2023; C. Sun et al. 2019; Yi Wang et al. 2022] highlight this potential. On the other hand, since audio is naturally paired with video, other studies [S. Lee et al. 2021; B. Shi et al. 2022c] have explored learning video representations from audio. More recently, multimodal approaches have emerged for video representation learning. LanguageBind [Zhu et al. 2023] utilizes depth, infrared, audio, and video to enhance video representations, while Video-LLaMA [Hang Zhang et al. 2023] learns video representations from free-form text and audio. Following these successes, our work aims to leverage multimodal data - video, audio and text - to advance the understanding of gestures.

9.3 Method

Our goal is to learn co-speech gesture representations from speech and text supervision. Given a dataset *G, S, L* of gesture clips, the accompanying speech segments and their corresponding transcriptions, our goal is to learn gesture representations that capture the rich semantics (from text, usually) and utterance style (usually from speech) of what is being spoken.

9.3.1 Overview

JEGAL learns gesture representations by solving two multimodal contrastive objectives between the gesture video and the two other modalities, i.e., speech and text. Each of the three modalities are first encoded using separate encoders $\mathbb{G}, \mathbb{S}, \mathbb{L}$ to get modality-specific embeddings $\mathbf{g^T} \in \mathbb{R}^{T \times d}$, $\mathbf{s^w} \in \mathbb{R}^{W \times d/2}$, $\mathbf{l^w} \in \mathbb{R}^{W \times d/2}$ to obtain frame-level (T) and word-level (W) representations. The speech and text embeddings are fused into joint speech-text embeddings $\mathbf{c^w} \in \mathbb{R}^{W \times d}$ as depicted in Fig 9.2.

We learn these representations using (1) a *global phrase-level contrastive objective*, and (2) a *local gesture-word coupling loss*. The first one encourages the model to learn global semantics to match a gesture clip to a speech/text segment. The second objective enforces the model to find the strongest word-level matches between the gesture clip and the other two modalities. We describe our architecture and loss functions in detail below.

9.3.2 Gesture Encoder

Gesture backbone. Given a gesture clip $G \in (T, h, w, 3)$ of T frames, we encode it using a stack of 3D convolution layers, similar to previous audio-visual networks [Afouras et al. 2020c; Prajwal et al. 2022a; Prajwal et al. 2022b], where the first layer has a temporal receptive field of 5 frames to capture the motion information. We obtain a sequence of T visual feature vectors of dimension d. These feature vectors are further encoded using a stack of Transformer encoder layers to get $\mathbf{f_g} \in \mathbb{R}^{T \times d}$. We initialize the backbone weights from GestSync [Sindhu B Hegde and Andrew Zisserman 2023] and keep it frozen.

Gesture head. We use a Transformer encoder followed by a projection layer to get the gesture embeddings, $\mathbf{g^T} \in \mathbb{R}^{T \times d}$.

9.3.3 Text Encoder

Text backbone. Given a text transcription L corresponding to the gesture video clip, we use the final layer outputs from a pre-trained bi-directional language model, multilingual Roberta XLM-Base [Conneau et al. 2019] to obtain text representations. The output of the text backbone is a sequence of sub-word feature vectors, f_l.

Text head. The text head is similar to the gesture head that uses a stack of Transformer layers. The sub-word embeddings f_l are encoded and projected to get the final sub-word embeddings l^{sw} of feature dimension $d/2$ each. We aggregate these sub-word tokens to word-level tokens later in the fusion block.

9.3.4 Audio Encoder

Given a speech waveform S, we convert it into melspectrograms which is encoded using a stack of 2D-CNN layers following previous works [J. S. Chung and Andrew Zisserman 2016b; Afouras et al. 2020c]. The output of the audio encoder is a sequence of T' speech feature vectors, $\mathbf{s} \in \mathbb{R}^{T' \times d/2}$.

9.3.5 Fusion Block

Before we fuse the speech and text embeddings, we aggregate the text and audio embeddings to obtain word-level feature vectors. We average the sub-word embeddings for each word to get word-level text embeddings $\mathbf{l^w} \in \mathbb{R}^{W \times d/2}$. Using the start and end times of each word, we average the speech features for each word to get word-level features $\mathbf{s^w} \in \mathbb{R}^{W \times d/2}$. We fuse the speech and text features by concatenating along the feature dimension to get joint word-level representations, $\mathbf{c^w} \in \mathbb{R}^{W \times d}$.

9.3.6 Gesture-Word Alignment

The word boundaries are aligned with the speech, but not necessarily with the gestures in the video. The gestures can be longer or shorter than the window in which the word is uttered, and can also be offset. To handle this discrepancy, we propose an attention-based pooling mechanism to obtain the gesture embedding corresponding to the word. We first pad the speech-based word start-end times

with $p = 10$ video frames on either side. Let S, E be the start and end frames for the padded window for the word c^{w_i}. We obtain the word-level gesture embedding, g^{w_i} by using the word embedding c^{w_i} for attention-pooling over the extended temporal interval of the word:

$$g^{w_i} = \sum_{j=S}^{E} \left(\frac{\exp(\gamma \cdot g^{T_j} \cdot c^{w_i})}{\sum_{j=S}^{E} \exp(\gamma \cdot g^{T_j} \cdot c^{w_i})} \right) \cdot g^{T_j} \tag{9.1}$$

9.3.7 Training Objective

We only have reliable supervision at the phrase level — meaning we know that a specific text or speech segment corresponds to a gesture clip. However, we do not know which individual words are gestured. Keeping this in mind, we employ two loss functions.

Global Phrase Contrastive Loss. To obtain the global phrase embeddings, we average pool the speech-text embeddings and the video frame embeddings to give us $\mathbf{c}$ and $\mathbf{g}$ respectively. Given a batch of N samples, we employ the contrastive Info-NCE loss [Oord et al. 2018] to encourage similarity between the N positive triplets, and dissimilarity between the $N^2 - N$ negative triplets:

$$\mathcal{L}_{seq} = \frac{1}{N} \sum_{i=1}^{N} \left(\log \frac{\exp(\gamma \cdot cos(g_i, c_i))}{\sum_{j=1}^{n} \exp(\gamma \cdot cos(g_i, c_j))} \right) \tag{9.2}$$

Local Gesture-word Coupling Loss. Pooling the word-level representations to compute the global phrase loss can lead to weak gesture-word associations (Tab 9.6, row 1). However, directly training to match gestures to words is not possible – very few words are gestured in a given phrase, and we do not know which of them are. Thus, we devise a new strategy to learn word-level correspondences using phrase-level supervision. Given a pair of word-level gesture and speech-text $(\mathbf{g^w}, \mathbf{c^w})$ embeddings, we first find the closest gesture g^{w_j} for each speech-text word c^{w_i}. Our hypothesis is that a matching $(\mathbf{g^w}, \mathbf{c^w})$ will have a higher number of strong word couplings than a non-matching pair. With this idea, we define the following scoring function and the gesture-word coupling loss:

$$\lambda(g_n^w, c_n^w) = \frac{1}{W} \sum_{i=1}^{W} \max_{j=1,2..W} cos(g_n^{w_i}, c_n^{w_j}) \tag{9.3}$$

$$\mathcal{L}_{couple} = \frac{1}{N} \sum_{i=1}^{N} \left(\log \frac{\exp(\gamma \cdot \lambda(g_i^w, c_i^w))}{\sum_{j=1}^{N} \exp(\gamma \cdot \lambda(g_i^w, c_j^w))} \right) \tag{9.4}$$

The gesture-word coupling loss simply maximizes λ for matching gesture-speech-text samples while minimizing λ for the negative ones. In other words, the model is encouraged to find more strong word-level couplings for positive gesture-speech-text phrases in the batch.

Our final loss function is a weighted sum of the two losses:

$$\mathbb{L} = \beta \cdot \mathcal{L}_{seq} + (1 - \beta) \cdot \mathcal{L}_{couple} \tag{9.5}$$

9.3.8 Implementation Details

We now describe the essential implementation details, more details can be found in the supplementary material.

Training data. We train our model, JEGAL, on triplets of gesture clips, speech segments, and text transcriptions. For the gesture frame inputs, we resize the frames to 270×480 pixels. We extract melspectrograms with a hop length of 10ms. The word-aligned text transcriptions are tokenized into wordpiece tokens. Using the start-end time of the word boundaries, we randomly sample a video clip between $2 - 10$ seconds in length.

Modality drop. In order to encourage the model to learn both speech and text representations equally well, we randomly set one of these modality inputs to zero 50% of the time. This is commonly done in audio-visual speech recognition models [B. Shi et al. 2022c; Afouras et al. 2018a]. This also allows us to use only one modality (speech or text) during inference, if necessary.

Model hyper-parameters. For the text and gesture heads, we set the number of Transformer layers to 3 and 6 respectively. The Transformer uses a hidden dimension of 512 and a feed-forward dimension of 2048 with 8 attention heads.

Training hyper-parameters. We use the AdamW optimizer [Loshchilov and Hutter 2017] with a learning rate of $5e^{-5}$, weight decay of $1e^{-4}$ and betas $(0.9, 0.98)$. We reduce the learning rate by a factor of 5 when the validation performance does not improve for 2 epochs.

9.3.9 Training Datasets

We train our model using the following datasets: (i) PATS [Ginosar et al. 2019], (ii) LRS3 [Afouras et al. 2018b], and (iii) LRS3-Lang [Afouras et al. 2020b; Sindhu B

Table 9.1: We train and evaluate on multiple datasets consisting of 720 hours of gesture clips spanning 5 languages and 7000+ speakers. For evaluation, we curate task-specific benchmarks from the publicly available AVSpeech [Ephrat et al. 2018] dataset.

Dataset	**split**	**# hours**	**# spk.**	**avg. clip duration (s)**	**# videos**
PATS [Ginosar et al. 2019]	train	162.3	24	11.37	51390
LRS3 [Afouras et al. 2018b]	train	189.6	4292	11.83	57337
LRS3-lang [Afouras et al. 2020b]	train	366.5	2642	18.03	73173
Combined	train	718.4	6958	14.2	181900
AVS-Ret	test	0.26	420	1.85	500
AVS-Spot	test	0.38	391	2.73	500
AVS-Asd	test	0.53	500	3.80	500

Hegde and Andrew Zisserman 2023]. By merging these three datasets, we compile a comprehensive multilingual gesture dataset suitable for both training and evaluation. The specifics of each dataset are outlined in Table 9.1. PATS [Ginosar et al. 2019] is a publicly available dataset comprising 250 hours of video content from 25 speakers sourced from diverse platforms such as lectures, talk-shows, YouTube, and televangelists. The LRS3 [Afouras et al. 2018b] dataset is composed of 400 hours of TED talks spanning a broad spectrum of speakers and a rich vocabulary. We also use a subset of the LRS3-lang [Afouras et al. 2020b] dataset, containing 366 hours across 4 distinct languages. We acquired this dataset directly from the authors, as it has not been made publicly available yet.

Pre-processing: We resample all videos to 25 FPS, and the speech to 16kHz. We leverage WhisperX [Bain et al. 2023] in cases where datasets lack word-aligned text transcripts. Additionally, using the $L2$ distance between consecutive frame body keypoints, we filter out samples with minimal gesture activity. We also make sure to mask out the face region to avoid leakage from lip movements. Table 9.1 presents the final statistics of all the datasets.

9.4 Downstream Tasks and Evaluation

We describe our newly curated evaluation benchmarks and the different downstream tasks to evaluate the quality of our learned gesture representations. The first is cross-modal retrieval, the second is spotting gestured words, and the third is active speaker detection. Note that in all the tasks, while we use the joint speech-text embedding, we can obtain uni-modal scores by inputting zeros to omit

a modality during inference.

9.4.1 Cross-modal Retrieval

Given a gallery of gesture-speech-text samples, the task is to retrieve a gesture clip given a speech segment and/or text and vice-versa. Concretely, given a speech or text as query, we obtain a speech-text embedding, $c \in \mathbb{R}^d$ and rank the gesture embeddings $g \in \mathbb{R}^d$ in the gallery by cosine similarity, highest being at the top. We do the same process for the gesture to speech-text retrieval as well.

Retrieving relevant gestures for a text or speech segment enables several practical applications. For digital avatars, we can retrieve most plausible hand gesture clips to accompany what the avatar is speaking, leading to a more immersive and engaging experience. In gaming applications, given a database of gesture sequences, the developer can automatically select the most relevant gestures to go with the in-game dialogues. Gestures can assist in language learning by improving word-level memory retention (e.g. eat, kick, clap). Language teaching apps will be able to retrieve gesture clips for sentences to improve the speed of foreign language learning.

9.4.2 Spotting Gestured Words

Given a gesture clip with the accompanying speech/text segment and a word of choice from this segment, the goal is to localize the word in the gesture clip. Concretely, we obtain word-level speech-text (c^w) embeddings and frame-level gesture embeddings, g^T. To localize the i-th word, c^{w_i}, we compute the cosine similarity of the word embedding with all the gesture frame embeddings. The localization of the word in the video is simply obtained by keeping only the locations with similarity scores $\geq \delta = 0.5$.

Spotting can be useful to enhance transcriptions by supplementing the plain words with stress and emotion labels. Another application would be to create word-level gesture databases, e.g. a thousand different ways the word “big" is gestured by people all over the world, which will be useful for language and communication analysis.

9.4.3 Active Speaker Detection

Given gesture clips of P different speakers, and a speech (S) and/or text segment (T), the goal is to predict the active speaker A who is uttering the queried speech/text. To do this, we extract the sequence-aggregated gesture features $g_i \in \mathbf{R}^d, i \in 1, 2, ..., P$ for each of the P clips. Given the query speech or text, we obtain the speech-text feature, c. The active speaker A is the one whose gesture and speech-text cosine similarity is maximum:

$$A = \underset{i \in 1,2,...,S}{\arg\max}\ cos(c,\ g_i) \tag{9.6}$$

The majority of audio-visual models, encompassing tasks like speech recognition, generation, and translation, primarily operate on inputs containing a single speaker. Thus, there arises a necessity to identify the speaker within a video segment. To determine the active speaker in a multi-speaker scenario, previous works [Afouras et al. 2018c; Rahimi et al. 2022] have shown the benefits of resorting to the face for lip-sync with the audio, and text subtitles when the audio is corrupted. We extend this thread even further. What happens if the lip region is occluded or unclear? Another important use-case is privacy preserving [R. Xu et al. 2021] active speaker detection: what if the active speaker detection needs to be done without leaking the face identity of the speaker? We show that we can successfully do this – with very little identity information, i.e. by only using the hand gestures, we can determine who is speaking.

Table 9.2: Cross-modal retrieval performance on the AVS-Ret benchmark (Sec 9.4.4). JEGAL outperforms the baselines by a large margin.

Method	Mod.		Speech-text to Gesture retrieval					Gesture to Speech-text retrieval				
	T	A	R@5 ↑	R@10↑	R@25↑	R@50↑	MR ↓	R@5 ↑	R@10↑	R@25↑	R@50↑	MR ↓
Random	✓	✓	1.00	2.00	5.00	10.00	250	1.00	2.00	5.00	10.00	250
Zero-shot												
Clip4Clip [Luo et al. 2022]	✓	✗	1.90	2.70	5.90	15.10	194	2.00	3.10	7.20	12.80	205
Language-Bind [Zhu et al. 2023]	✓	✗	1.70	2.30	6.00	14.00	207	2.00	3.20	6.60	12.20	212
GestSync [Sindhu B Hegde and Andrew Zisserman 2023]	✗	✓	1.80	2.60	6.60	14.40	200	1.80	3.60	11.00	20.20	159
Fine-tuned												
Clip4Clip [Luo et al. 2022]	✓	✗	2.40	4.60	8.90	14.00	214	2.40	5.00	9.20	15.60	219
Language-Bind [Zhu et al. 2023]	✓	✗	2.40	4.40	9.00	15.60	187	2.80	4.60	9.00	14.40	199
GestSync [Sindhu B Hegde and Andrew Zisserman 2023]	✗	✓	3.60	7.20	18.00	30.20	113	6.20	9.20	21.00	31.20	109
GestureDiffuClip [Ao et al. 2023]	✓	✗	8.90	11.67	22.80	33.20	96	9.70	14.30	24.10	36.00	94
Ours												
JEGAL	✓	✗	15.80	30.00	50.20	58.80	25	10.80	19.20	34.60	47.80	55
JEGAL	✗	✓	14.20	26.00	49.40	58.80	26	11.60	20.20	33.20	45.00	62
JEGAL	✓	✓	**17.40**	**28.40**	**53.00**	**69.80**	**22**	**16.40**	**26.40**	**43.60**	**55.20**	**38**

9.4.4 AVSpeech Test Benchmarks

Using the AVSpeech official test set [Ephrat et al. 2018], we manually curate three separate evaluation benchmarks for the three downstream gesture tasks. The statistics for the evaluation test sets are summarized in Table 9.1.

AVS-Ret. We create a new cross-modal retrieval benchmark containing diverse gesture clips of hundreds of unique speakers. We choose a gallery of 500 clips, which also contain isolated clean speech and accurate text transcriptions. We verify that the clips contain reasonable gesture activity and transcripts with at least two nouns or verbs or adjectives. For evaluation, we use the standard metrics used in other video-text retrieval works [Zhu et al. 2023; Yi Wang et al. 2022], i.e. Recall@K and Median Rank. We evaluate both gesture (g) to content (speech-text c) retrieval and vice-versa and show both unimodal and multimodal retrieval performance.

AVS-Spot. To quantitatively evaluate the gesture spotting task, we manually curate a new test dataset where we search and annotate clips that clearly contain a word that is gestured. We obtain 500 such clips, each containing a target word that is clearly gestured. The manual annotation process removes all kinds of label noise in the test set, allowing for a faithful evaluation of our newly defined gesture spotting task. Additionally, we also manually annotate these target words with binary “stress/emphasis" labels, which can have important cues about the gesture (Table 9.5).

AVS-Asd. To build the evaluation dataset for active speaker detection, we first choose 500 “target" clips. For these target clips, we create three evaluation subsets, where we choose $P-1$ clips from different speakers, where $P = 2, 4, 6$. We report the accuracy of detecting the correct target speaker out of the P different speakers.

9.5 Results

9.5.1 Baselines

For baselines, we report performance of zero-shot pre-trained video-language models [Zhu et al. 2023; Yi Wang et al. 2022] and pre-trained GestSync [Sindhu B Hegde and Andrew Zisserman 2023], which learns gesture-audio correspondences by solving for audio-visual synchronization. We also report scores after fine-tuning

all these models further on our training data for a fair comparison. In addition, we compare with the semantic encoder of GestureDiffuCLIP [Ao et al. 2023], by training it on our dataset.

9.5.2 Cross-modal Retrieval

In Table 9.2, we compare the performance of JEGAL against other baselines on the cross-modal retrieval task. Zero-shot evaluation of foundational vision-language models like LanguageBind [Zhu et al. 2023] and Clip4Clip [Luo et al. 2022] leads to slightly higher than chance performance. These models are designed to capture different kinds of features: they cannot handle a large number of frames, and learn non-gesture attributes like identity and scene. Fine-tuning these models nearly doubles their performance on the task, but it is still far from the performance of JEGAL. GestSync [Sindhu B Hegde and Andrew Zisserman 2023] is trained with a gesture-related task, and hence performs nearly twice as well as the foundational vision-language models. However, since this network is trained to detect synchronization offsets in speech and video, its representations perform poorly for global semantic tasks like retrieval. This is also partly true for our model when we turn off the global phrase loss (Table 9.6 row 2 vs row 3). GestureDiffuCLIP's semantic encoder [Ao et al. 2023] performs the best among the baselines, however, the lack of local word-level semantic supervision leads to an inferior performance compared to JEGAL.

Furthermore, none of the baseline approaches ingest and fuse multi-modal speech-text inputs. Our JEGAL model outperforms previous methods by a large margin. JEGAL can retrieve gestures from speech or text queries with similar performance. The opposite direction is also true, i.e. retrieving speech or text for a query gesture clip. Finally, retrieving with the fused speech-text representation, c is clearly better than the unimodal variants, showing that the speech and text embeddings each encode information that is not present in the other modality.

9.5.3 Spotting Gestured Words

Table 9.3 compares the spotting accuracy of different methods on the AVS-Spot benchmark. Unlike JEGAL that uses the word-level L_{couple} loss, all the baseline methods, including the semantic encoder of GestureDiffuCLIP [Ao et al. 2023], use

Table 9.3: Gesture-based word spotting performance on the AVS-Spot benchmark (Sec 9.4.4).

Method	Mod.		Accuracy ↑
	T	A	
Zero-shot			
Clip4Clip [Luo et al. 2022]	✓	✗	7.62
Language-Bind [Zhu et al. 2023]	✓	✗	9.11
GestSync [Sindhu B Hegde and Andrew Zisserman 2023]	✗	✓	5.12
Fine-tuned			
Clip4Clip [Luo et al. 2022]	✓	✗	14.79
Language-Bind [Zhu et al. 2023]	✓	✗	15.68
GestSync [Sindhu B Hegde and Andrew Zisserman 2023]	✗	✓	10.41
GestureDiffuClip [Ao et al. 2023]	✓	✗	21.50
Ours			
JEGAL	✓	✗	60.40
JEGAL	✗	✓	40.40
JEGAL	✓	✓	**63.00**

only a phrase-level loss and hence, struggle to learn fine-grained word-level associations. Through this task, we also see the big advantage of using text modality in addition to speech – text-based gesture spotting is more accurate than using audio. This is expected, since word-level semantic correspondences are easier to learn in text space. Having said that, using speech alongside text still gives a clear improvement even in the gesture spotting task.

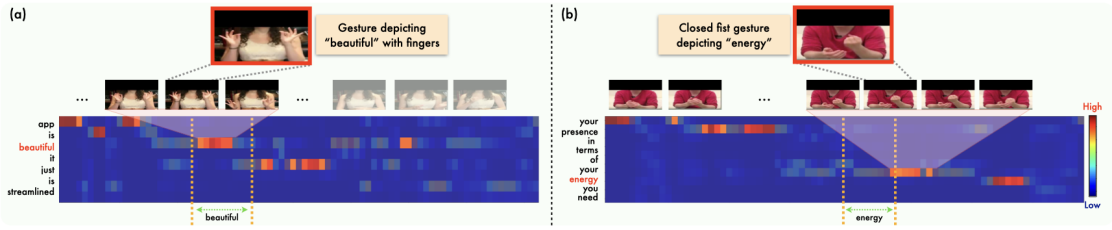


Figure 9.3: JEGAL can spot the gestured words in a video clip. Here, we show a similarity heatmap of words vs video frames. The vertical yellow lines indicate the speech-based word boundaries of 'beautiful' and 'energy'. The red triangles zoom into the corresponding frames where JEGAL detects the words, clearly aligning with the gestures. For the word "beautiful" the gestured segment is smaller than the spoken word boundary and for the word "energy" it extends well beyond to the right of the word boundary. The alignment layer (Sec 9.3.6) allows the model to look beyond just the speech-based boundaries. Note that our model learns to perform gestured-word spotting without using any training labels on which words are gestured.

In Fig 9.3, we show two examples of spotting gestured words. The heatmaps show the similarity of the word (red is higher) along the video frames. In the first example, the lady gestures "beautiful" with her fingers, and in the second example,

the speaker clenches the fist to show "energy". We can see that not all words get a high similarity score, only ones with distinctive gestures are spotted. Furthermore, the spotting does not exactly align with the speech-based word boundary (indicated by yellow vertical lines). Our alignment layer (Sec 9.3.6) allows the model to look beyond the speech-based boundary and find the exact frames where the word is gestured.

9.5.4 Active Speaker Detection

In Table 9.4, we show the accuracy of identifying the target speaker for a given text and/or speech segment. This task is different from our other tasks – it does not need a strong holistic understanding of the gesture sequence like retrieval, nor does it need semantic word-level understanding. It can simply be solved by checking for frame-level synchronization, which is exactly why GestSync [Sindhu B Hegde and Andrew Zisserman 2023], LanguageBind [Zhu et al. 2023] baselines perform well in this task. Also, compared to the previous tasks, speech information is more useful here compared to text. While the text-only variant of JEGAL lags behind GestSync [Sindhu B Hegde and Andrew Zisserman 2023], the speech and speech-text variants actually emerges as the best performing approaches. This task shows that JEGAL learns gesture features that are useful for a wide variety of tasks, across both speech and text modalities.

Table 9.4: Performance of active speaker detection on AVS-Asd benchmark. We report the mean class accuracy of predicting the active speaker among a set of S speakers, where $S = 2, 4, 6$.

Method	Mod.		2 spk.	4 spk.	6 spk.
	T	A			
Random	✓	✓	50.0	25.0	16.7
Zero-shot					
Clip4Clip [Luo et al. 2022]	✓	✗	56.4	31.2	23.0
Language-Bind [Zhu et al. 2023]	✓	✗	60.1	38.9	27.2
GestSync [Sindhu B Hegde and Andrew Zisserman 2023]	✗	✓	57.3	35.1	26.2
Fine-tuned					
Clip4Clip [Luo et al. 2022]	✓	✗	60.0	40.1	28.8
Language-Bind [Zhu et al. 2023]	✓	✗	62.5	41.4	28.2
GestSync [Sindhu B Hegde and Andrew Zisserman 2023]	✗	✓	72.8	57.2	48.0
GestureDiffuClip [Ao et al. 2023]	✓	✗	66.8	43.9	34.8
Ours					
JEGAL	✓	✗	69.4	48.6	37.4
JEGAL	✗	✓	77.2	60.0	52.2
JEGAL	✓	✓	**80.8**	**66.4**	**59.0**

9.6 Insights and Ablations

In this section, we provide additional insights into the gesture signals learned from the speech and text modalities, the impact of the two loss functions on the downstream tasks, and the choice of speech-text fusion.

9.6.1 Speech v/s Text Modalities

We have already seen how our model can flexibly leverage speech or text modality to solve three different kinds of tasks. In Fig 9.4, we show an example where audio-based gesture word spotting is successful, but text-based spotting is not. And we see that the speech clearly has the word “action" being emphasized. This example leads us to do a deeper analysis of gesture word spotting based on stress cues. We divide the binary stress labels to split the AVS-Spot test set (500 samples) into two subsets, one containing only samples with stressed/emphasized words (100 samples) and the other subset containing the remaining words (400 samples). In Table 9.5, we report the same spotting accuracy metric on these subsets separately. While we saw previously in Table 9.3 that speech-based gesture spotting is clearly inferior compared to text-based gesture spotting, it is not always the case. We see that the difference in spotting accuracy between the stressed and non-stressed words is higher for the speech modality. This indicates that speech modality pays

more attention to emphasis of the word compared to the text modality.

Table 9.5: Stressed words are more likely to be spotted with speech-based spotting than non-stressed words. As seen in the column "Δ", the difference between stressed vs. non-stressed word spotting is higher for the speech modality.

Modality	With stress	W/o stress	All words	Δ
Text	68.00	58.50	60.40	16.2%
Speech	51.00	37.75	44.00	35.1%

9.6.2 Impact of loss functions

In Table 9.6, we study the impact of our two loss functions. In the first row, we only train with the phrase contrastive loss, which captures the global semantics. The second row is the variant trained only with gesture-word coupling loss, which captures local word-level semantics. Compared to the dual loss model, the results from these individual loss models are significantly worse. Specifically, the variant without the word coupling loss performs poorly on the spotting task and the one without the sequence contrastive loss performs poorly on retrieval and active speaker detection. The combination of the two losses performs the best across all the tasks, thus demonstrating the complementary nature of the two training objectives.

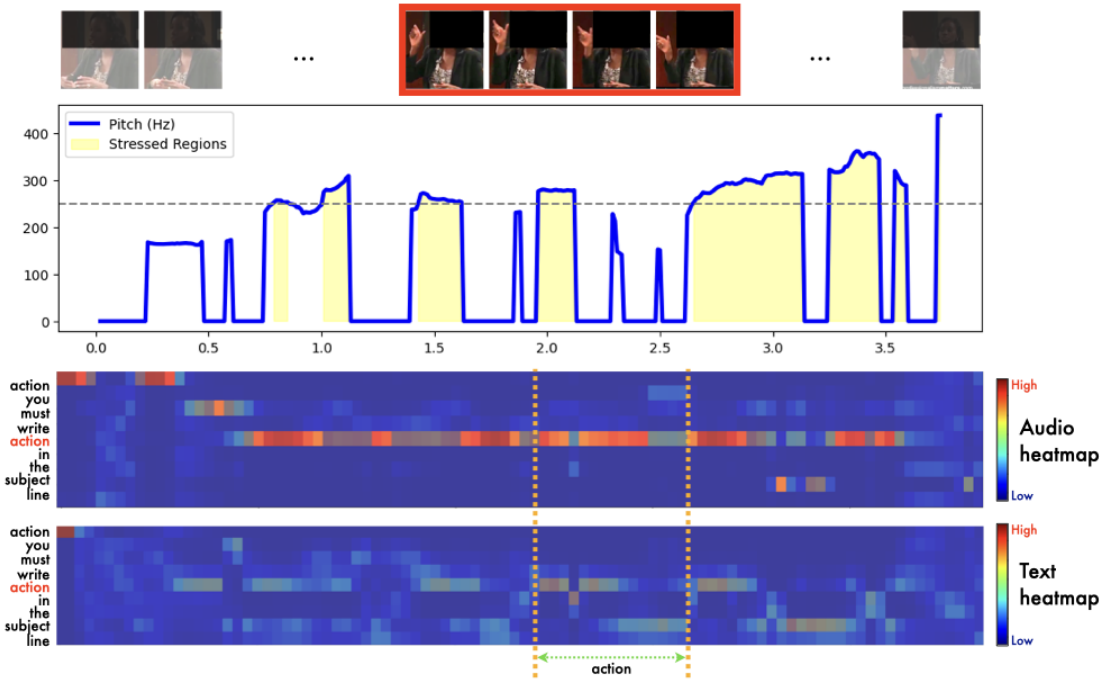


Figure 9.4: Audio and text heatmap examples when words are stressed.

Table 9.6: Each loss encourages feature learning at different temporal granularity, and combination of the two loss functions performs the best.

Loss	**Retrieval**			**Spotting**	**ASD**
	R@5 ↑	**R@10 ↑**	**MR ↓**	**Acc. ↑**	**Acc. ↑**
Seq. contrastive	11.60	21.40	43	18.83	42.8
Word coupling	7.90	13.40	78	51.98	19.6
Seq. + Word coupling	**17.40**	**28.40**	**22**	**63.00**	**59.0**

9.6.3 Fusion techniques

In Table 9.7, we ablate different ways to fuse the speech and text features. The first case is to not fuse at all and have two separate pairwise contrastive losses: gesture-audio and gesture-text. We can see in the first two rows that this is an inferior design. In fact, it is better to train with a single contrastive head after fusing the speech and text embeddings (rows 3, 4). The fusion strategy of choice would be to concatenate, rather than average.

Table 9.7: Ablation study on fusing speech and text modalities. Training without fusing is far worse, as the model cannot perform the tasks by using multiple information streams at the same time.

	Retrieval			**Spotting**	**ASD**
	R@5 ↑	**R@10 ↑**	**MR ↓**	**Acc. ↑**	**Acc. ↑**
Pairwise (with text)	9.39	15.58	70	34.31	29.6
Pairwise (with audio)	9.80	16.60	72	23.67	31.4
Late fusion (avg.)	16.00	24.60	30	57.27	54.0
Late fusion (concat.)	**17.40**	**28.40**	**22**	**63.00**	**59.0**

9.7 Conclusion

In this work, we learn a joint embedding space that captures cross-modal relationships with gestures, speech, and language. We show that we can learn such an embedding space with weak supervision using a careful design of two loss functions. We evaluate these new representations on three new downstream tasks and manually curated test sets. We observe that the two modalities, i.e., speech and text, learn complementary features that can be useful for different kinds of gesture-related tasks. One promising future direction would be to explore 2D and 3D keypoint based inputs to make the network computationally lighter and less susceptible to distracting features.

Acknowledgements. The authors would like to thank Piyush Bagad, Ragav Sachdeva, and Jaesung Hugh for their valuable discussions. They also extend their thanks to David Pinto for setting up the data annotation tool and to Ashish Thandavan for his support with the infrastructure. This research is funded by EPSRC Programme Grant VisualAI EP/T028572/1, and a Royal Society Research Professorship RP \R1 \191132.

APPENDIX

9.8 Most Gestured Words

In Figure 9.5, we show the most commonly spotted gestured words that are spotted by JEGAL on the AVS-Spot test set: pointing gestures (you, my, we), adjectives/adverbs (little, open, whole, gigantic, broad), direction words (forward, here, below) and numbers (one, two, first).



Figure 9.5: Word cloud for the most commonly gestured words.

9.9 Model Details

In Table 9.8, we provide detailed description of the model architecture. We will release the code and models to support future research.

Table 9.8: Overview of the model architecture, detailing the input modalities, network components, and key parameters used in each stage of our framework.

Branch	Layer/Module	Input Shape	Output Shape
Visual Branch			
	Vision backbone	3 × T × 270 x 480	T × 1024
	Projection MLP		
	- Linear	T × 1024	T × 512
	- LayerNorm	T × 512	T × 512
	- ReLU	T × 512	T × 512
	- Linear	T × 512	T × 512
	Positional Encoding	T × 512	T × 512
	Transformer (N=6 layers)		
	- Self-Attention (h=8)	T × 512	T × 512
	- Feed Forward	T × 512	T × 512
	Output Projection	T × 512	T × 512
Text Branch			
	mRoberta Text backbone	W	W × 768
	Transformer (N=3 layers)		
	- Self-Attention (h=8)	W × 768	W × 768
	- Feed Forward	W × 768	W × 768
	Output Projection	W × 768	W × 256
Audio Branch			
	Melspectrogram Input	1 × 80 × 4T	-
	Conv2D + BN + ReLU (k=5, s=1, p=2)	1 × 80 × 4T	32 × 80 × 4T
	Conv2D + BN + ReLU (k=3, s=2, p=1)	32 × 40 × 2T	64 × 40 × 2T
	Conv2D + BN + ReLU (k=3, s=2, p=1)	64 × 40 × 2T	128 × 20 × T
	Conv2D + BN + ReLU (k=3, s=(3,1), p=1)	128 × 7 × T	256 × 7 × T
	Conv2D + BN + ReLU (k=3, s=(3,1), p=1)	256 × 3 × T	256 × 3 × T
	Conv2D (k=1, s=(3,1), p=0)	256 × 3 × T	256 × 1 × T
	Output Projection + reshape	256 × 1 × T	T × 256
Late Fusion			
	Encoded Features		
	- Visual	T × 512	-
	- Text + sub-word pooling	W × 256	W × 256
	- Audio + sub-word pooling	T × 256	W × 256

9.10 Dataset Visualization

In Figure 9.6, we present examples from our manually annotated AVS-Spot test set (curated from the publicly available AVSpeech test dataset [Ephrat et al. 2018]), designed to evaluate downstream gesture spotting performance. As shown, the dataset includes a diverse collection of unique words, carefully curated to ensure clear and contextually appropriate gestures. For instance, in row-1, the word "little" is accompanied by a gesture where two fingers move close together to indicate a small size; in row-2, the speaker points backward to represent the word "back"; and in row-6, an inward motion is used to indicate "my".



Figure 9.6: Visualization of the **AVS-Spot** dataset, showcasing video frames from different samples. Each row corresponds to a single video, with the highlighted keyword indicating the annotated gestured word for spotting. The figure illustrates the dataset's diversity, featuring a wide range of unique keywords, various speakers, and distinct gestures.

9.11 Qualitative Results

In Figure 9.7, we show additional qualitative examples for gesture spotting. In the left text panel, the red-highlighted word represents the keyword to be spotted, as curated in the AVS-Spot test set. The word-labeled vertical columns, separated by yellow lines, indicate the word boundaries derived from speech-text alignment. JEGAL successfully spots most of these keywords, as shown by the red heatmaps. Notably, the boundaries may vary slightly since speakers often gesture and speak at slightly different times, highlighting the inherent challenges of our weakly-supervised gesture representation learning task.

In Figure 9.8, we present additional examples demonstrating that audio-based gesture spotting tends to focus on "stressed regions" in speech, unlike text-based spotting. This difference is evident from the audio and text heatmaps for each sample. In Figure 9.8, our model detects the stressed keywords "every" and "respond", whereas the text-only model misses these words. Evidently, the audio-only model looks for word emphasis cues (indicated by high pitch) as such words are more likely to be gestured. This would be difficult to infer from text modality alone. These examples illustrate the advantages of leveraging audio cues for gesture spotting.

9.12 Limitations and Areas of Improvement

Our work is the first to tackle large-scale co-speech gesture understanding. We highlight some of the limitations of our approach here. Having trained predominantly on TED/TEDx talks, our models can struggle with co-speech conversational gestures. Further, the TED/TEDx talk setting inherently encourages continuous gesturing and moving the arms, which might not always be the case for other settings (like conversational interactions). Another aspect the model struggles with is when there are limited gesture actions or hand movements that are unrelated to speech. Finally, given that we learn with only weak sequence-level supervision, the model can "find shortcuts" by focusing on simple rhythmic hand movements that occur in certain gestures classes like the beat gestures. This can affect the representation quality of iconic and deitic gestures that contain clear semantic meaning. While we still show that our models can spot such gestures, future works can focus

on improving this imbalance in gesture classes.

9.13 Potential Negative Societal Impacts

While our research significantly contributes to advancing gesture understanding, there are some potential risks of surveillance, as the system could infer conversations from a distance by identifying words/phrases. Nonetheless, we believe the benefits outweigh these risks, as the technology enhances human-machine interaction by integrating non-verbal cues. According to the 55% rule, non-verbal communication constitutes 55% of overall communication. This highlights the importance of enabling machines to engage in holistic, natural interactions with humans by understanding non-verbal elements like gestures.

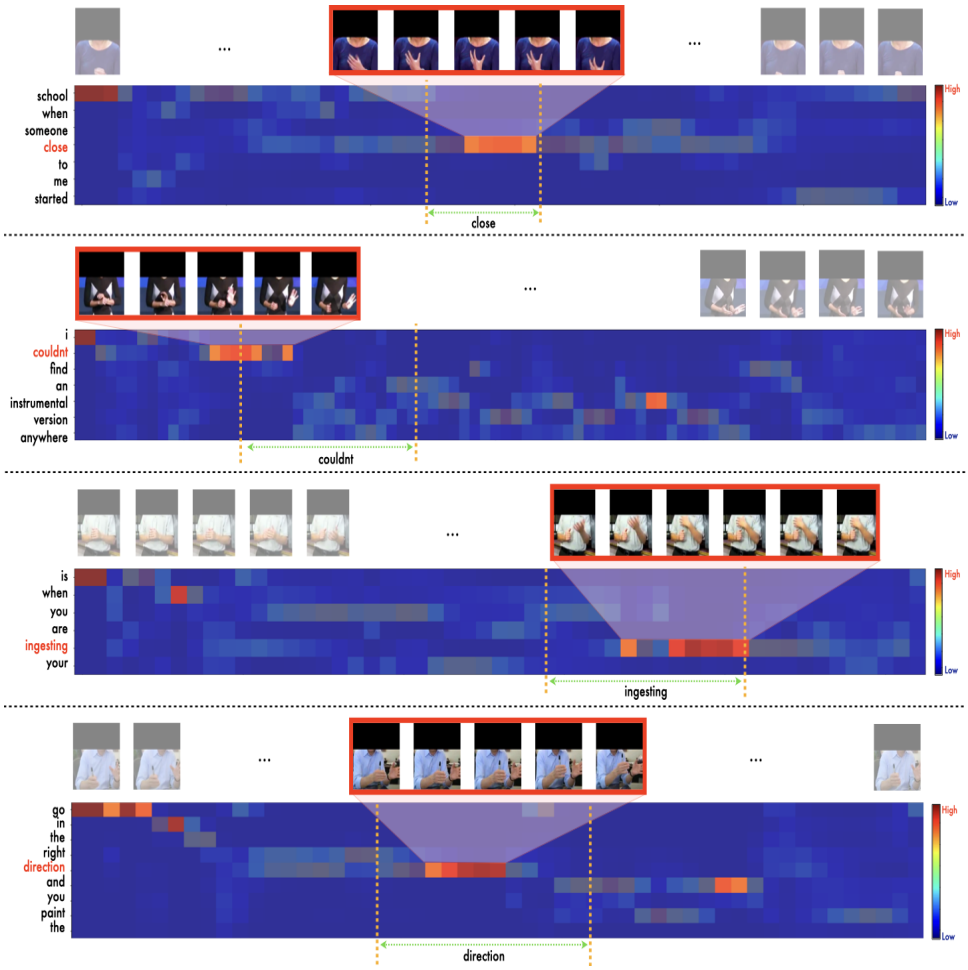


Figure 9.7: Additional gestured word spotting results on AVS-Spot dataset. Keywords are highlighted in red on the left panel and the speech-based force alignment word boundaries are marked by yellow lines. JEGAL successfully spots the gestured keywords, demonstrating its robustness across diverse gestures and speakers. The red triangles zoom into the corresponding frames where JEGAL detects the keywords, clearly aligning with the gestures. Note that in some cases (e.g., rows 2 and 3), ground-truth boundaries may slightly differ, as the speaker can gesture and utter the same word at slightly different times. JEGAL effectively estimates the approximate intervals where the target word is gestured.

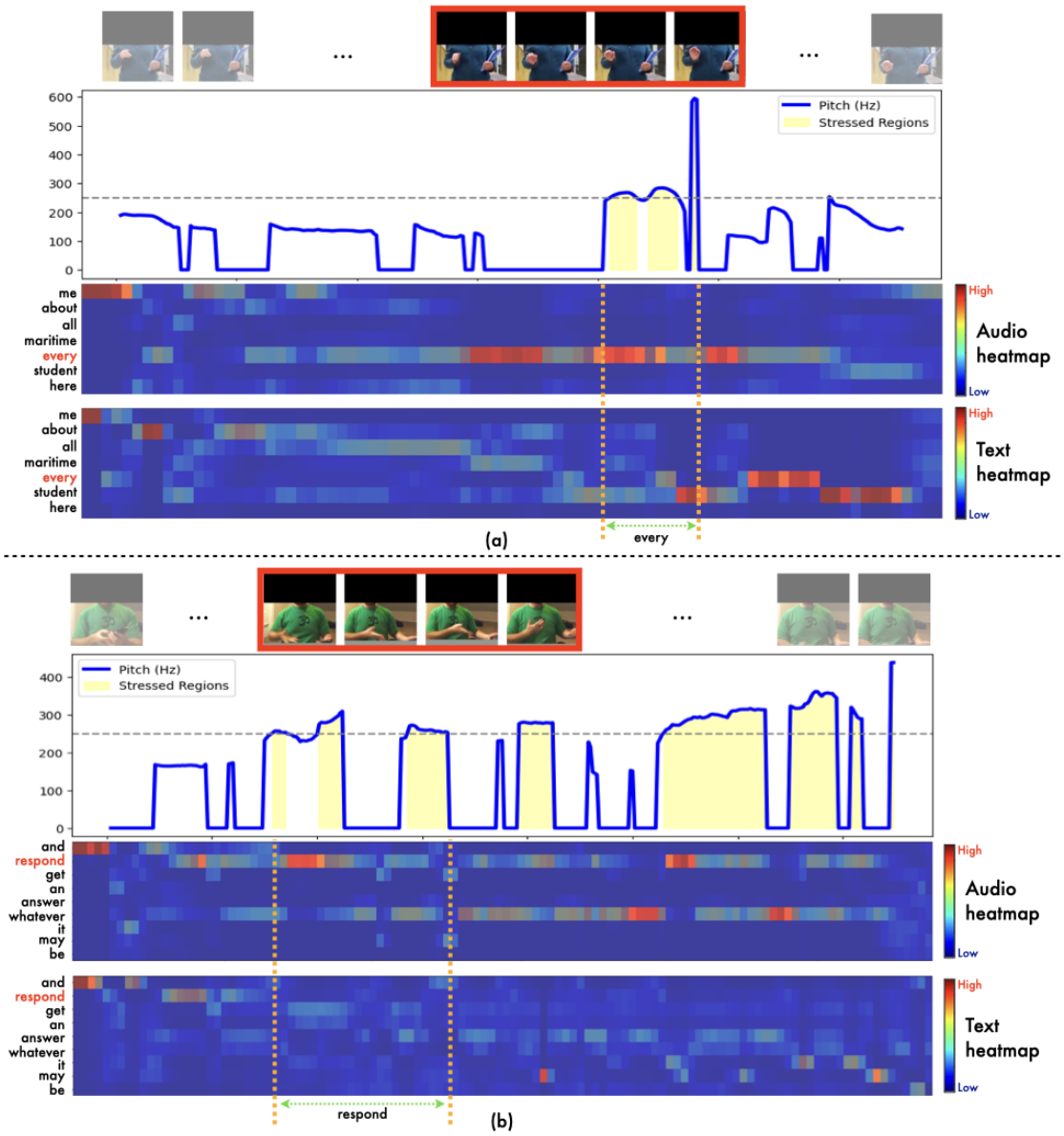


Figure 9.8: Examples highlighting the role of stressed speech regions in audio-based gesture spotting. The audio-only model successfully detects the stressed keywords "every" and "respond", whereas the text-only model misses these words. Evidently, the audio-only model looks for word emphasis cues (indicated by high pitch) as such words are more likely to be gestured. This would be difficult to infer from text modality alone. These examples illustrate the advantages of leveraging audio cues for gesture spotting.

Chapter 10

Discussion

In this chapter, we summarize our achievements and the impact of the presented works in this thesis (Section 10.1). We then briefly discuss the ethical considerations (Section 10.2) and highlight directions for future works (Section 10.3).

10.1 Achievements and Impact

Lip reading

In Chapter 2, we introduce a novel visual backbone for lip reading. The proposed transformer-based pooling method can attend and effectively track the lip region in dynamic talking head videos. We pioneer the use of sub-word units for lip reading, showing improvements in both WER and inference efficiency. Our model and training pipeline outperforms proprietary models that use an order of magnitude more data for training. Following our work, most of the lip reading models [Ma et al. 2023; B. Shi et al. 2022c; Lian et al. 2023; Haliassos et al. 2022] today have moved away from character-based generation. The visual features from this work have proven instrumental in other tasks like keyword spotting [Prajwal et al. 2021], spotting sign language mouthings [Prajwal et al. 2022b], and lip-to-speech synthesis [S. Hegde et al. 2023].

In Chapter 3, we show that we can re-purpose a large-scale, frozen, pre-trained ASR model to perform lip reading to achieve state-of-the-art results on standard lip reading benchmarks. We demonstrate that we can extend our approach to the visual language identification task, with clear improvements over previous methods. In Chapter 4, we introduce a multi-task, multilingual model that is capable of both

language identification and transcription from silent lip movements. We scale our approach using only automatic, noisy labels from a pre-trained ASR [Radford et al. 2022] on a large corpus on unlabeled videos, and outperform previous monolingual methods trained on small amount of manually annotated data. Our multilingual model can be fine-tuned for new low-resource languages achieving word error rates that are 15-20 points lower than training on these languages from scratch. We will release our training data and models to support future research in lip reading.

Visual Keyword Spotting.

In Chapter 5, we build upon the lip reading visual features from Chapter 2 and introduce a new architecture for spotting keywords in silent videos. The model fuses visual and keyword information through full cross-modal attention, and outperforms prior methods by a large margin on standard benchmarks. The attention-based information fusion has inspired multiple other works in keyword spotting [Y.-H. Lee and N. Cho 2023; K. Zhang et al. 2024] and video-text alignment tasks [M. Kim et al. 2023a].

Our new keyword spotting model is particularly important for the downstream task of spotting mouthings in sign language videos, achieving 2× higher accuracies compared to previous methods. Consquently, it has been used to curate large-scale sign language datasets [Momeni et al. 2022; Prajwal et al. 2022b].

Sign Language Recognition.

In Chapter 6, we use the new keyword spotting model (Chapter 5), along with other approaches to help curate large-scale sign recognition datasets. We demonstrate the use of sign recognition models as a way of densely labeling sign language clips. We also introduce a novel approach of leveraging in-domain exemplars, further enhancing the density of annotations in sign language data. These dense annotations act as the foundational dataset for multiple sign recognition tasks (Chapter 7, 8) in this work. The dataset has also supported several other sign language alignment [Walsh et al. 2023], sign recognition [Shen and anc Yi Yang 2022; Wong et al. 2023] and translation [Guo et al. 2024; Sincan et al. 2023; Sincan et al. 2024] papers.

In Chapter 7, we focus on recognizing a specific class of signs called finger-

spelling, typically used to sign spoken language words that do not have a sign, e.g. names of people and places. Our multi-task model identifies the presence of fingerspelling, localizes it in time, and recognises the sequence of signed letters, all while using only weak, noisy annotations. We curate a test set of 5K video clips, annotated by human experts for evaluating this new fingerspelling recognition task. We release this publicly to support sign language research. The data and model predictions have been for continuous sign language recognition in the subsequent chapter. Our fingerspelling architecture has also been used for pose-based fingerspelling recognition [Fayyazsanavi et al. 2024].

In Chapter 8, we design a multi-task transformer model, capable of performing both large vocabulary continuous sign language recognition (CSLR), and sign language retrieval. This marks an important step towards translation in open vocabulary settings. We also construct the largest CSLR test set, with continuous sign-level human annotations spanning six hours of videos, which is also publicly available for future research. Our results in CSLR marks a key step towards sign language translation.

Learning Gesture Representations.

In Chapter 9, we shift our focus from signing gestures to co-speech gestures. We learn a joint embedding space for speech transcriptions and the accompanying gestures. This allows us to do various tasks using only hand gestures: retrieving matching gestures for a given transcription, detecting who is speaking, and aligning videos to subtitles.

10.2 Ethical Considerations

In this section, we discuss some of the opportunities and limitations of the works in this thesis.

Lip reading and Visual Keyword Spotting

Narrowing the gap between lip reading and ASR performance opens up opportunities for useful applications: (i) improving speech recognition when the audio is corrupted in some manner; (ii) enabling silent dictation; (iii) transcribing archival silent films; (iv) helping speech-impaired individuals, e.g. people suffering from

Lou Gehrig's disease speak [Shillingford et al. 2019]; and (v) enabling people with aphonia (loss of voice) to communicate just by using lip movements. Further, spotting keywords in silent clips enables the use of personal assistants (wake word recognition), searching through archival content and spotting isolated mouthings, particularly in signing videos.

But, these models also raise privacy issues and the risk of potential malign uses. One issue that is often raised is the potential for malign surveillance, e.g. using CCTV footage from public spaces to eavesdrop on private civilian conversations. However, this is in fact very low risk due to a number of factors: we achieve a low WER on benchmarks containing video material that is professionally produced and at high resolutions and frame-rates, and under good lighting conditions. Moreover the speakers are aware of being filmed and collaborate, most of the time speaking while frontally facing the camera. In contrast, CCTV usually operate at much lower resolution and frame rates and from unusual angles. As shown in prior work [J. S. Chung and Andrew Zisserman 2017; Shillingford et al. 2019; Makino et al. 2019], lip reading performance greatly deteriorates with lower frame rate or input resolutions, or when non-frontal (e.g. profile or overhead viewpoints) rather than frontal speaker views are considered.

We have made the code and pre-trained models of our works publicly available. This technology is already available to a small handful of corporations that have access to enough data and compute resources for training. We believe that open access is important in order to accelerate progress in the field, as well as to enable research on defences against potential adversarial attacks [Ma et al. 2021b; Gupta et al. 2021; Jethanandani and D. Tang 2020]. Finally, releasing our code and models leads to democratisation of research – making strong models publicly available rather than to only a handful of companies can accelerate research while also making the entire process more transparent.

Overall, we believe that the benefits of the positive applications of lip reading that we have discussed (e.g. medical) greatly outweigh the risk of malevolent uses, the latter ones being inflated, therefore transparent research into this field should be continued and encouraged by the community.

Sign Recognition

Our research on sign language understanding is a promising step towards enabling applications that are useful for deaf communities. The involvement of members from the deaf community is crucial to reaching an end goal that actually results in practical value-addition for the deaf community. Applications such as efficient search and indexing of sign language videos, and the integration of sign-reading capabilities into virtual assistants such as Siri and Alexa are now closer to reality than ever before. There are also more applications that are a bit more futuristic: real-time sign language interpretation in video calls, sign language translation for critical scenarios such as hospitals, police stations, and airports. As the sign language recognition and translation models improve in accuracy, these applications will be a massive boon to the deaf communities.

10.3 Limitations and Future Work

We discuss some of the shortcomings of this thesis and some the promising directions for the future.

10.3.1 Lip reading and Visual Keyword Spotting

Despite the massive drop in word error rates, especially in English, the impact of these improvements is not clearly evaluated for some of the most impactful applications. Lip reading is particularly useful for transcribing videos that are inherently devoid of vocal speech such as silent movies, old archival videos with poor audio, and conversations of patients with vocalisation problems. However, all the standard evaluation benchmarks of today contain videos where people are vocally speaking. Even though only the lip movements are used in the evaluation, the domain gap has not been extensively studied and benchmarked. The silent clips can differ from the vocal counterparts in terms of utterance speed, style, continuity, sentence structure and length. One of the promising future directions is to evaluate the lip reading models for such practical applications with social impact. Curating a manually-verified test set similar to the LRS3 test set [Afouras et al. 2018b] will be a good step forward. The next goal would be to reduce the domain gap, either with self-supervised approaches that are already used in lip reading today [B. Shi

et al. 2022c; Haliassos et al. 2022] or with the pseudo-labeling approach that we employed in Chapter 6 to improve keyword spotting for sign language mouthings.

Another limiting factor of the keyword spotting and lipreading models in this thesis is that they are trained with tightly-cropped utterance segments – the speech starts and ends at the first and last frame respectively. This can affect the performance on silent videos, where the segment boundaries cannot be determined accurately. The models in this thesis are also further limited by the training data diversity, given that the entire training data is mostly TED/TEDx talks. Models do not generalize well [Djilali et al. 2024] beyond a limited range of emotions, can struggle with conversational, casual vocabulary, informal utterances, and so on. The goal would be to expand the training data domains and train on different types of video data (as opposed to just TED/TEDx videos).

Another avenue for improvement would be train self-supervised representations for multilingual lip reading. Self-supervised visual representations have had a massive impact on English lip reading, and given the scarcity of multilingual data, this would a fruitful path towards improving multilingual lip reading.

10.3.2 Sign Language

In spite of creating the largest publicly available BSL dataset, the sign recognition models today are severely limited by data quality and biases due to interpreted signing, especially from a limited set of speakers. The BOBSL dataset [Albanie et al. 2021b] only contains 39 signers, severely limiting the ability to generalize to real-world signing applications. While the interpreted data provides a foundation for learning strong visual representations of a large vocabulary of signs, it is still not clear how well these representations generalize to conversational signing, which brings its own additional challenges [Bragg et al. 2019]. It would be hard to realize the goal of sign language translation for "practical use", without training on more diverse, large-scale datasets.

Another limitation of our current sign recognition models is that they struggle to recognize non-lexical/partially-lexical signs such as gestures, depicting signs, facial expressions, slang, pointing signs, etc. These sign classes are relatively more difficult, particularly due to the lack of training data. However, recognizing these would be an essential step in the goal towards achieving sign language translation.

In this thesis, we circumvent the lack of training labels with weak and noisy automatic labels. A promising future direction would be to also explore self-supervised approaches to learn sign representations.

10.3.3 Learning Co-speech Gesture Representations

This thesis touches the surface of understanding co-speech gestures. JEGAL is the first model to be able to associate gesture sequences to textual transcriptions on such a large scale. One of its big limitations is that it fails to learn representations that show a "monotonic alignment" between the video and text sequence. This is partially due to the fact that we are average-pooling the representations for the contrastive loss. The next goal should be to learn more fine-grained dense gesture-text associations to facilitate tasks such as spotting and searching notable gestures (e.g. tiny, big, tall, hug, pointing, you, I, we, etc.). Effective fine-grained temporal modeling is crucial for gesture understanding (similar to sign language and lip reading) – an feature not highly prioritized in existing video-text contrastive frameworks [Sevilla-Lara et al. 2021; Buch et al. 2022]. In the current training strategy, the loss can be minimized by learning to associate a single gesture cue to the textual phrase, while ignoring the other cues. Stronger negative sampling, training with loss functions that encourage local-to-global information aggregation will be instrumental for learning a rich semantic embedding space for co-speech gestures and text.

10.4 Conclusion

In this thesis, we proposed various approaches to understand three important areas of visual communication. We proposed several approaches to scale lip reading "for-free" to large datasets and multiple languages. The learned lip representations also enabled tasks such as visual speech detection, visual language identification, and visual keyword spotting. A major performance improvement in the keyword spotting task, helps us to curate large-scale sign language datasets, allowing us train weakly-supervised sign recognition models. We propose new benchmarks and approaches to move from isolated sign recognition to recognizing sequences of signs in continuous video. Finally, we explore the area of co-speech gestures and

learn gesture representations that can solve several downstream gesture-text tasks. We believe that these works pave the way for future research on understanding the visual modality of communication, and how to scale these models in spite of the lack of labeled training data.

References

Sami Abu-El-Haija, Nisarg Kothari, Joonseok Lee, Paul Natsev, George Toderici, Balakrishnan Varadarajan, and Sudheendra Vijayanarasimhan (2016). "Youtube-8m: A large-scale video classification benchmark". In: *arXiv preprint arXiv:1609.08675*.

Nikolas Adaloglou, Theocharis Chatzis, Ilias Papastratis, Andreas Stergioulas, Georgios Th Papadopoulos, Vassia Zacharopoulou, George J Xydopoulos, Klimnis Atzakas, Dimitris Papazachariou, and Petros Daras (2020). "A Comprehensive Study on Sign Language Recognition Methods". In: *arXiv*.

Triantafyllos Afouras, Joon Son Chung, Andrew Senior, Oriol Vinyals, and Andrew Zisserman (2019). "Deep Audio-Visual Speech Recognition". In: *IEEE TPAMI*.

Triantafyllos Afouras, Joon Son Chung, and Andrew Zisserman (2018a). "Deep Lip Reading: a comparison of models and an online application". In: *INTERSPEECH*.

Triantafyllos Afouras, Joon Son Chung, and Andrew Zisserman (2018b). "LRS3-TED: a large-scale dataset for visual speech recognition". In: *arXiv preprint arXiv:1809.00496*.

Triantafyllos Afouras, Joon Son Chung, and Andrew Zisserman (2018c). "The Conversation: Deep Audio-Visual Speech Enhancement". In: *INTERSPEECH*.

Triantafyllos Afouras, Joon Son Chung, and Andrew Zisserman (2020a). "ASR is all you need: Cross-modal distillation for lip reading". In: *International Conference on Acoustics, Speech, and Signal Processing*.

Triantafyllos Afouras, Joon Son Chung, and Andrew Zisserman (2020b). "Now you're speaking my language: Visual language identification". In: *INTERSPEECH*.

Triantafyllos Afouras, Andrew Owens, Joon Son Chung, and Andrew Zisserman (2020c). “Self-Supervised Learning of Audio-Visual Objects from Video”. In: *ECCV*.

Rakesh Agrawal, Tomasz Imieliński, and Arun Swami (1993). “Mining association rules between sets of items in large databases”. In: *ACM SIGMOD International Conference on Management of Data*.

Roee Aharoni, Melvin Johnson, and Orhan Firat (2019). “Massively multilingual neural machine translation”. In: *arXiv preprint arXiv:1903.00089*.

Jean-Baptiste Alayrac et al. (2022). “Flamingo: a visual language model for few-shot learning”. In: *arXiv preprint arXiv:2204.14198*.

Samuel Albanie, Gül Varol, Liliane Momeni, Triantafyllos Afouras, Andrew Brown, Chuhan Zhang, Ernesto Coto, Necati Cihan Camgöz, Ben Saunders, Abhishek Dutta, Neil Fox, Richard Bowden, Bencie Woll, and Andrew Zisserman (2021a). “Signer Diarisation in the Wild”. In: *Technical Report*.

Samuel Albanie, Gül Varol, Liliane Momeni, Triantafyllos Afouras, Joon Son Chung, Neil Fox, and Andrew Zisserman (2020). “BSL-1K: Scaling up Co-articulated Sign Language Recognition Using Mouthing Cues”. In: *ECCV*.

Samuel Albanie, Gül Varol, Liliane Momeni, Hannah Bull, Triantafyllos Afouras, Himel Chowdhury, Neil Fox, Bencie Woll, Rob Cooper, Andrew McParland, and Andrew Zisserman (2021b). “BOBSL: BBC-Oxford British Sign Language Dataset”. In: *arXiv*.

Juan León Alcázar, Fabian Caba, Long Mai, Federico Perazzi, Joon-Young Lee, Pablo Arbeláez, and Bernard Ghanem (2020). “Active speakers in context”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*.

Dario Amodei, Sundaram Ananthanarayanan, Rishita Anubhai, Jingliang Bai, Eric Battenberg, Carl Case, Jared Casper, Bryan Catanzaro, Qiang Cheng, Guoliang Chen, et al. (2016). “Deep speech 2: End-to-end speech recognition in english and mandarin”. In: *International conference on machine learning*. PMLR.

Michael Andric and Steven L Small (2012). “Gesture’s neural language”. In: *Frontiers in psychology*.

Mohamed Anwar, Bowen Shi, Vedanuj Goswami, Wei-Ning Hsu, Juan Pino, and Changhan Wang (2023). "MuAViC: A Multilingual Audio-Visual Corpus for Robust Speech Recognition and Robust Speech-to-Text Translation". In: *arXiv preprint arXiv:2303.00628*.

Tenglong Ao, Zeyi Zhang, and Libin Liu (2023). "Gesturediffuclip: Gesture diffusion model with clip latents". In: *ACM Transactions on Graphics (TOG)*.

Sercan Arik, Markus Kliegl, Rewon Child, Joel Hestness, Andrew Gibiansky, Chris Fougner, Ryan Prenger, and Adam Coates (2017). "Convolutional Recurrent Neural Networks for Small-Footprint Keyword Spotting". In: *INTERSPEECH*.

Yannis M. Assael, Brendan Shillingford, Shimon Whiteson, and Nando de Freitas (2016). "LipNet: Sentence-level Lipreading". In: *arXiv:1611.01599*.

Andrew Aubrey, Bertrand Rivet, Yulia Hicks, Laurent Girin, Jonathon Chambers, and Christian Jutten (2007). "Two novel visual voice activity detectors based on appearance models and retinal filtering". In: *2007 15th European Signal Processing Conference*.

Andrew J. Aubrey, Yulia A. Hicks, and Jonathon A. Chambers (2010). "Visual voice activity detection with optical flow". In: *Iet Image Processing*.

Kartik Audhkhasi, Andrew Rosenberg, Abhinav Sethy, Bhuvana Ramabhadran, and Brian Kingsbury (2017). "End to-end ASR-free keyword search from speech". In: *IEEE Journal of Selected Topics in Signal Processing*.

Alexei Baevski, Wei-Ning Hsu, Qiantong Xu, Arun Babu, Jiatao Gu, and Michael Auli (2022). "Data2vec: A general framework for self-supervised learning in speech, vision and language". In: *ICML*.

Alexei Baevski, Yuhao Zhou, Abdelrahman Mohamed, and Michael Auli (2020). "wav2vec 2.0: A framework for self-supervised learning of speech representations". In: *Advances in neural information processing systems*.

Max Bain, Jaesung Huh, Tengda Han, and Andrew Zisserman (2023). *WhisperX: Time-Accurate Speech Transcription of Long-Form Audio*. Tech. rep. University of Oxford.

Loic Barrault, Yu-An Chung, Mariano Cora Meglioli, David Dale, Ning Dong, Paul-Ambroise Duquenne, Hady Elsahar, Hongyu Gong, Kevin Heffernan,

John Hoffman, et al. (2023a). “SeamlessM4T-Massively Multilingual & Multimodal Machine Translation”. In: *arXiv preprint arXiv:2308.11596*.

Loic Barrault, Yu-An Chung, Mariano Coria Meglioli, David Dale, Ning Dong, Mark Duppenthaler, Paul-Ambroise Duquenne, Brian Ellis, Hady Elsahar, Justin Haaheim, et al. (2023b). “Seamless: Multilingual Expressive and Streaming Speech Translation”. In: *arXiv preprint arXiv:2312.05187*.

Britta Bauer and Hermann Hienz (2000). “Relevant features for video-based continuous sign language recognition”. In: *Proceedings Fourth IEEE International Conference on Automatic Face and Gesture Recognition (Cat. No. PR00580)*. IEEE.

Valentin Belissen, Annelies Braffort, and Michèle Gouiffès (2020). “Experimenting the Automatic Recognition of Non-Conventionalized Units in Sign Language”. In: *Algorithms*.

Axel Berg, Mark O’Connor, and Miguel Tairum Cruz (2021). “Keyword Transformer: A Self-Attention Model for Keyword Spotting”. In: *arXiv preprint arXiv:2104.00769*.

Gedas Bertasius, Heng Wang, and Lorenzo Torresani (2021). “Is Space-Time Attention All You Need for Video Understanding?” In: *Proc. ICML*.

Fedor Borisyuk, Albert Gordo, and Viswanath Sivakumar (2018). “Rosetta: Large scale system for text detection and recognition in images”. In: *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*.

P Boyes Braem and RL Sutton-Spence (2001). *The Hands Are The Head of The Mouth. The Mouth as Articulator in Sign Languages*. English. Hamburg: Signum Press.

Danielle Bragg, Oscar Koller, Mary Bellard, Larwan Berke, Patrick Boudreault, Annelies Braffort, Naomi Caselli, Matt Huenerfauth, Hernisa Kacorri, Tessa Verhoef, Christian Vogler, and Meredith Ringel Morris (2019). “Sign Language Recognition, Generation, and Translation: An Interdisciplinary Perspective”. In: *ACM SIGACCESS*.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. (2020). “Language models are few-shot learners”. In: *Advances in neural information processing systems*.

Shyamal Buch, Cristobal Eyzaguirre, Adrien Gaidon, Jiajun Wu, Li Fei-Fei, and Juan Carlos Niebles (2022). “Revisiting the “Video” in Video-Language Understanding”. In: *CVPR*.

Patrick Buehler, Mark Everingham, Daniel P. Huttenlocher, and Andrew Zisserman (2008). “Long Term Arm and Hand Tracking for Continuous Sign Language TV Broadcasts”. In: *Proceedings of the 19th British Machine Vision Conference, Leeds*.

Patrick Buehler, Andrew Zisserman, and Mark Everingham (2009). “Learning sign language by watching TV (using weakly aligned subtitles)”. In: *2009 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE.

Hannah Bull, Triantafyllos Afouras, Gül Varol, Samuel Albanie, Liliane Momeni, and Andrew Zisserman (2021). “Aligning Subtitles in Sign Language Videos”. In: *ICCV*. IEEE.

Hannah Bull, Michèle Gouiffès, and Annelies Braffort (2020). “Automatic Segmentation of Sign Language into Subtitle-Units”. In: *ECCVW, Sign Language Recognition, Translation and Production (SLRTP)*.

Necati Camgoz, Ben Saunders, Guillaume Rochette, Marco Giovanelli, Giacomo Inches, Robin Nachtrab-Ribback, and Richard Bowden (2021). “Content4All Open Research Sign Language Translation Datasets”. In: *arXiv preprint arXiv:2105.02351*.

Necati Cihan Camgoz, Simon Hadfield, Oscar Koller, Hermann Ney, and Richard Bowden (2018). “Neural Sign Language Translation”. In: *CVPR*.

Necati Cihan Camgoz, Oscar Koller, Simon Hadfield, and Richard Bowden (2020a). “Multi-channel Transformers for Multi-articulatory Sign Language Translation”. In: *ECCVW*.

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

Dogan Can and M. Saraçlar (2011). “Lattice Indexing for Spoken Term Detection”. In: *IEEE Transactions on Audio, Speech, and Language Processing*.

Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko (2020). "End-to-End Object Detection with Transformers". In: *arXiv preprint arXiv:2005.12872*.

J. Carreira and A. Zisserman (2017). "Quo vadis, action recognition? A new model and the kinetics dataset". In: *CVPR*.

Edresson Casanova, Julian Weber, Christopher D Shulby, Arnaldo Candido Junior, Eren Gölge, and Moacir A Ponti (2022). "Yourtts: Towards zero-shot multi-speaker tts and zero-shot voice conversion for everyone". In: *International Conference on Machine Learning*. PMLR.

William Chan, Navdeep Jaitly, Quoc V. Le, and Oriol Vinyals (2016). "Listen, Attend and Spell: A Neural Network for Large Vocabulary Conversational Speech Recognition". In: *ICASSP*.

Oscar Chang, Hank Liao, Dmitriy Serdyuk, Ankit Shahy, and Olivier Siohan (2024). "Conformer is All You Need for Visual Speech Recognition". In: *ICASSP*. IEEE.

C Charayaphan and AE Marble (1992). "Image processing system for interpreting motion in American Sign Language". In: *Journal of Biomedical Engineering*.

Jingyuan Chen, Xinpeng Chen, Lin Ma, Zequn Jie, and Tat-Seng Chua (2018). "Temporally grounding natural sentence in video". In: *EMNLP*.

Xi Chen, Shouyi Yin, Dandan Song, Peng Ouyang, Leibo Liu, and Shaojun Wei (2019). "Small-footprint Keyword Spotting with Graph Convolutional Network". In: *IEEE Automatic Speech Recognition and Understanding Workshop*.

Ka Leong Cheng, Zhaoyang Yang, Qifeng Chen, and Yu-Wing Tai (2020). "Fully Convolutional Networks for Continuous Sign Language Recognition". In: *ECCV*.

Yiting Cheng, Fangyun Wei, Jianmin Bao, Dong Chen, and Wenqiang Zhang (2023). "Cico: Domain-aware sign language retrieval via cross-lingual contrastive learning". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*.

Minsu Cho, Suha Kwak, Cordelia Schmid, and Jean Ponce (2015). "Unsupervised object discovery and localization in the wild: Part-based matching with bottom-up region proposals". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*.

Joon Son Chung (2019). “Naver at ActivityNet Challenge 2019–Task B Active Speaker Detection (AVA)”. In: *arXiv preprint arXiv:1906.10555*.

Joon Son Chung, Arsha Nagrani, and Andrew Zisserman (2018). “VoxCeleb2: Deep Speaker Recognition”. In: *INTERSPEECH*.

Joon Son Chung, Andrew Senior, Oriol Vinyals, and Andrew Zisserman (2017). “Lip Reading Sentences in the Wild”. In: *CVPR*.

Joon Son Chung and Andrew Zisserman (2016a). “Lip Reading in the Wild”. In: *ACCV*.

Joon Son Chung and Andrew Zisserman (2016b). “Out of time: automated lip sync in the wild”. In: *Workshop on Multi-view Lip-reading, ACCV*.

Joon Son Chung and Andrew Zisserman (2017). “Lip Reading in Profile”. In: *BMVC*.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov (2019). “Unsupervised cross-lingual representation learning at scale”. In: *arXiv preprint arXiv:1911.02116*.

Helen Cooper and Richard Bowden (2009). “Learning signs from subtitles: A weakly supervised approach to sign language recognition”. In: *CVPR*.

Helen Cooper, Nicolas Pugeault, and Richard Bowden (2011). “Reading the signs: A video based sign dictionary”. In: *2011 IEEE international conference on computer vision workshops (ICCV Workshops)*. IEEE.

Kearsy Cormier, Adam Schembri, and Bencie Woll (2013). “Pronouns and pointing in sign languages”. In: *Lingua*.

Marta R Costa-jussà, James Cross, Onur Çelebi, Maha Elbayad, Kenneth Heafield, Kevin Heffernan, Elahe Kalbassi, Janice Lam, Daniel Licht, Jean Maillard, et al. (2022). “No language left behind: Scaling human-centered machine translation”. In: *arXiv preprint arXiv:2207.04672*.

Runpeng Cui, Hu Liu, and Changshui Zhang (2017). “Recurrent convolutional neural networks for continuous sign language recognition by staged optimization”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*.

Runpeng Cui, Hu Liu, and Changshui Zhang (2019). “A Deep Neural Framework for Continuous Sign Language Recognition by Iterative Training”. In: *IEEE Transactions on Multimedia*.

Trevor J Darrell, Irfan A Essa, and Alex P Pentland (1996). “Task-specific gesture analysis in real-time using interpolated views”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Jeffrey Davis (1990). “Linguistic Transference and Interference: Interpreting Between”. In: *Sign language research: Theoretical issues*.

Mathieu De Coster, Mieke Van Herreweghe, and Joni Dambre (2020). “Sign language recognition with transformer networks”. In: *12th international conference on language resources and evaluation*. European Language Resources Association (ELRA).

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei (2009). “Imagenet: A large-scale hierarchical image database”. In: *2009 IEEE conference on computer vision and pattern recognition*. Ieee.

Thomas Deselaers, Bogdan Alexe, and Vittorio Ferrari (2010). “Localizing objects while learning their appearance”. In: *European conference on computer vision*. Springer.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova (2018). “BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding”. In: *CoRR*.

Terrance DeVries and Graham W. Taylor (2017). “Improved Regularization of Convolutional Neural Networks with Cutout”. In: *arXiv preprint arXiv:1708.04552*.

Thomas G Dietterich, Richard H Lathrop, and Tomás Lozano-Pérez (1997). “Solving the multiple instance problem with axis-parallel rectangles”. In: *Artificial intelligence*.

Runwei Ding, Cheng Pang, and Hong Liu (2018). “Audio-Visual Keyword Spotting Based on Multidimensional Convolutional Neural Network”. In: *IEEE International Conference on Image Processing*.

Yasser Abdelaziz Dahou Djilali, Sanath Narayan, Haithem Boussaid, Ebtessam Almazrouei, and Merouane Debbah (2023). “Lip2Vec: Efficient and Robust Visual Speech Recognition via Latent-to-Latent Visual to Audio

Representation Mapping". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*.

Yasser Abdelaziz Dahou Djilali, Sanath Narayan, Eustache LeBihan, Haithem Boussaid, Ebtesam Almazrouei, and Merouane Debbah (2024). "Do VSR Models Generalize Beyond LRS3?" In: *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*.

Linhao Dong, Shuang Xu, and Bo Xu (2018). "Speech-Transformer: A No-Recurrence Sequence-to-Sequence Model for Speech Recognition". In: *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby (2021). "An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale". In: *International Conference on Learning Representations*. URL: https://openreview.net/forum?id=YicbFdNTTy.

Philippe Dreuw, Jens Forster, Thomas Deselaers, and Hermann Ney (2008). "Efficient Approximations to Model-based Joint Tracking and Recognition of Continuous Sign Language". In: *IEEE International Conference on Automatic Face and Gesture Recognition*.

Amanda Duarte, Samuel Albanie, Xavier Giró-i-Nieto, and Gül Varol (2022). "Sign Language Video Retrieval with Free-Form Textual Queries". In: *CVPR*.

Amanda Duarte, Shruti Palaskar, Lucas Ventura, Deepti Ghadiyaram, Kenneth DeHaan, Florian Metze, Jordi Torres, and Xavier Giro-i-Nieto (2021). "How2Sign: A Large-scale Multimodal Dataset for Continuous American Sign Language". In: *Conference on Computer Vision and Pattern Recognition (CVPR)*.

Abhishek Dutta and Andrew Zisserman (2019). "The VIA annotation software for images, audio and video". In: *Proceedings of the 27th ACM international conference on multimedia*.

Ariel Ephrat, Inbar Mosseri, Oran Lang, Tali Dekel, Kevin Wilson, Avinatan Hassidim, William T Freeman, and Michael Rubinstein (2018). "Looking to Listen at the Cocktail Party: A Speaker-Independent Audio-Visual Model for Speech Separation". In: *ACM Transactions on Graphics*.

Sergio Escalera, Jordi Gonzàlez, Xavier Baró, Miguel Reyes, Oscar Lopes, Isabelle Guyon, Vassilis Athitsos, and Hugo Escalante (2013). "Multi-modal gesture recognition challenge 2013: Dataset and results". In: *Proceedings of the 15th ACM on International conference on multimodal interaction*.

Mark Everingham, S. M. Ali Eslami, Luc Van Gool, Chris K. I. Williams, John Winn, and Andrew Zisserman (Jan. 2015). "The Pascal Visual Object Classes Challenge: A Retrospective". In: *IJCV*.

Mark Everingham, Luc Van Gool, Chris K. I. Williams, John Winn, and Andrew Zisserman (2010). "The PASCAL Visual Object Classes (VOC) Challenge". In: *IJCV*.

Gaolin Fang, Xiujuan Gao, Wen Gao, and Yiqiang Chen (2004). "A novel approach to automatically extracting basic units from chinese sign language". In: *Proceedings of the 17th International Conference on Pattern Recognition, 2004. ICPR 2004.* IEEE.

Ali Farhadi and David Forsyth (2006). "Aligning ASL for statistical translation using a discriminative word model". In: *CVPR*.

Pooya Fayyazsanavi, Negar Nejatishahidin, and Jana Košecká (2024). "Fingerspelling PoseNet: Enhancing Fingerspelling Translation with Pose-Based Transformer Models". In: *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*.

Ingo Feinerer and Kurt Hornik (2020). *wordnet: WordNet Interface*. R package version 0.1-15. URL: https://CRAN.R-project.org/package=wordnet.

Souheil Fenghour, Daqing Chen, Kun Guo, and Perry Xiao (2020). "Lip Reading Sentences Using Deep Learning With Only Visual Cues". In: *IEEE Access*.

Santiago Fernandez, Alex Graves, and Jurgen Schmidhuber (2007). "An application of recurrent neural networks to discriminative keyword spotting". In: *International Conference on Artificial Neural Networks*.

Michael Filhol (2020). "Elicitation and Corpus of Spontaneous Sign Language Discourse Representation Diagrams". In: *LREC*.

Holger Fillbrandt, Suat Akyol, and K-F Kraiss (2003). "Extraction of 3D hand shape and posture from image sequences for sign language recognition". In: *SOI*.

Frank Fowley and Anthony Ventresque (2021). "Sign Language Fingerspelling Recognition using Synthetic Data." In: *AICS*.

William T Freeman and Michal Roth (1995). “Orientation histograms for hand gesture recognition”. In: *International workshop on automatic face and gesture recognition*. Citeseer.

Jiyang Gao, Chen Sun, Zhenheng Yang, and Ram Nevatia (2017). “Tall: Temporal activity localization via language query”. In: *ICCV*.

Esam Ghaleb, Ilya Burenko, Marlou Rasenberg, Wim Pouw, Peter Uhrig, Judith Holler, Ivan Toni, Aslı Özyürek, and Raquel Fernández (2024). “Co-Speech Gesture Detection through Multi-phase Sequence Labeling”. In: *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*.

Soham Ghosh, Anuva Agarwal, Zarana Parekh, and Alexander Hauptmann (2019). “ExCL: Extractive Clip Localization Using Natural Language Descriptions”. In: *NAACL-HLT*.

Shiry Ginosar, Amir Bar, Gefen Kohavi, Caroline Chan, Andrew Owens, and Jitendra Malik (2019). “Learning individual styles of conversational gesture”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*.

R Girshick (2015). “Fast r-cnn”. In: *arXiv preprint arXiv:1504.08083*.

Paul Goh and Eun-Jung Holden (2006). “Dynamic fingerspelling recognition using geometric and motion features”. In: *2006 International Conference on Image Processing*. IEEE.

Ramazan Gokberk Cinbis, Jakob Verbeek, and Cordelia Schmid (2014). “Multi-fold mil training for weakly supervised object localization”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*.

Alan J Goldschen, Oscar N Garcia, and Eric Petajan (1994). “Continuous optical automatic speech recognition by lipreading”. In: *Proceedings of 1994 28th Asilomar Conference on Signals, Systems and Computers*. Ieee.

Alan J Goldschen, Oscar N Garcia, and Eric D Petajan (1997). “Continuous automatic speech recognition by lipreading”. In: *Motion-Based recognition*. Springer.

Javier Gonzalez-Dominguez, David Eustis, Ignacio Lopez-Moreno, Andrew Senior, Françoise Beaufays, and Pedro J Moreno (2014). “A real-time end-to-end multilingual speech recognition architecture”. In: *IEEE Journal of selected topics in signal processing*.

John N Gowdy, Amarnag Subramanya, Chris Bartels, and Jeff Bilmes (2004). “DBN based multi-stream models for audio-visual speech recognition”. In: *2004 IEEE International conference on acoustics, speech, and signal processing*. IEEE.

Raghav Goyal, Samira Ebrahimi Kahou, Vincent Michalski, Joanna Materzynska, Susanne Westphal, Heuna Kim, Valentin Haenel, Ingo Fruend, Peter Yianilos, Moritz Mueller-Freitag, et al. (2017). “The" something something" video database for learning and evaluating visual common sense”. In: *Proceedings of the IEEE international conference on computer vision*.

Alex Graves, Santiago Fernández, Faustino Gomez, and Jürgen Schmidhuber (2006). “Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks”. In: *Proceedings of the International Conference on Machine Learning*. ACM.

Anmol Gulati, James Qin, Chung-Cheng Chiu, Niki Parmar, Yu Zhang, Jiahui Yu, Wei Han, Shibo Wang, Zhengdong Zhang, Yonghui Wu, and Ruoming Pang (2020). “Conformer: Convolution-augmented Transformer for Speech Recognition”. In: *INTERSPEECH*.

Zhengsheng Guo, Zhiwei He, Wenxiang Jiao, Xing Wang, Rui Wang, Kehai Chen, Zhaopeng Tu, Yong Xu, and Min Zhang (2024). “Unsupervised Sign Language Translation and Generation”. In: *Proceedings of the International Conference on Learning Representations*.

Anup Kumar Gupta, Puneet Gupta, and Esa Rahtu (2021). “FATALRead-Fooling visual speech recognition models”. In: *Applied Intelligence*.

Sylvain Guy, Stéphane Lathuilière, Pablo Mesejo, and Radu Horaud (2021). “Learning Visual Voice Activity Detection with an Automatically Annotated Dataset”. In: *2020 25th International Conference on Pattern Recognition (ICPR)*. IEEE.

Alexandros Haliassos, Pingchuan Ma, Rodrigo Mira, Stavros Petridis, and Maja Pantic (2022). “Jointly Learning Visual and Auditory Speech Representations from Raw Data”. In: *arXiv preprint arXiv:2212.06246*.

Thomas Hanke (2004). “HamNoSys - representing sign language data in language resources and language processing contexts”. In: *LREC Workshop proceedings: Representation and processing of sign languages*.

Mingfeng Hao, Mutallip Mamut, Nurbiya Yadikar, Alimjan Aysa, and Kurban Ubul (2020). “A survey of research on lipreading technology”. In: *IEEE Access*.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun (2016). “Deep residual learning for image recognition”. In: *CVPR*.

Yanzhang He, Rohit Prabhavalkar, Kanishka Rao, Wei Li, Anton Bakhtin, and Ian McGraw (2017). “Streaming small-footprint keyword spotting using sequence-to-sequence models”. In: *IEEE Automatic Speech Recognition and Understanding Workshop*.

Yanzhang He, Tara N Sainath, Rohit Prabhavalkar, Ian McGraw, Raziel Alvarez, Ding Zhao, David Rybach, Anjuli Kannan, Yonghui Wu, Ruoming Pang, et al. (2019). “Streaming end-to-end speech recognition for mobile devices”. In: *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE.

Sindhu Hegde, Rudrabha Mukhopadhyay, CV Jawahar, and Vinay Namboodiri (2023). “Towards Accurate Lip-to-Speech Synthesis in-the-Wild”. In: *Proceedings of the 31st ACM International Conference on Multimedia*.

Sindhu B Hegde and Andrew Zisserman (2023). “GestSync: Determining who is speaking without a talking head”. In: *BMVC*.

Sindhu B. Hegde, K.R. Prajwal, Rudrabha Mukhopadhyay, Vinay P. Namboodiri, and C.V. Jawahar (2021). “Visual Speech Enhancement Without a Real Visual Stream”. In: *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*.

Lisa Anne Hendricks, O. Wang, E. Shechtman, Josef Sivic, Trevor Darrell, and Bryan C. Russell (2017). “Localizing Moments in Video with Natural Language”. In: *ICCV*.

Sepp Hochreiter, Yoshua Bengio, Paolo Frasconi, Jürgen Schmidhuber, et al. (2001). *Gradient flow in recurrent nets: the difficulty of learning long-term dependencies*.

Yukiya Hono, Koh Mitsuda, Tianyu Zhao, Kentaro Mitsui, Toshiaki Wakatsuki, and Kei Sawada (2024). “Integrating pre-trained speech and language models for end-to-end speech recognition”. In: *Findings of the Association for Computational Linguistics ACL 2024*.

Wei-Ning Hsu et al. (2021). “Hubert: Self-supervised speech representation learning by masked prediction of hidden units”. In: *IEEE/ACM Transactions on Audio, Speech, and Language Processing*.

Wei-Ning Hsu and Bowen Shi (2022). “u-HuBERT: Unified Mixed-Modal Speech Pretraining And Zero-Shot Transfer to Unlabeled Modality”. In: *Advances in Neural Information Processing Systems*.

Hezhen Hu, Weichao Zhao, Wengang Zhou, Yuechen Wang, and Houqiang Li (2021a). “SignBERT: Pre-Training of Hand-Model-Aware Representation for Sign Language Recognition”. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*.

Hezhen Hu, Wengang Zhou, Junfu Pu, and Houqiang Li (2021b). “Global-local enhancement network for NMF-aware sign language recognition”. In: *ACM transactions on multimedia computing, communications, and applications (TOMM)*.

Jie Huang, Wengang Zhou, Houqiang Li, and Weiping Li (2018a). “Attention-based 3D-CNNs for large-vocabulary sign language recognition”. In: *IEEE Transactions on Circuits and Systems for Video Technology*.

Jie Huang, Wengang Zhou, Qilin Zhang, Houqiang Li, and Weiping Li (2018b). “Video-based Sign Language Recognition without Temporal Segmentation”. In: *AAAI*.

Kyuyeon Hwang, Minjae Lee, and Wonyong Sung (2015). “Online Keyword Spotting with a Character-Level Recurrent Neural Network”. In: *arXiv:1512.08903*.

Fumitada Itakura (1990). “Minimum Prediction Residual Principle Applied to Speech Recognition”. In: *Readings in Speech Recognition*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc.

Mahir Jethanandani and Derek Tang (2020). “Adversarial Attacks Against LipNet: End-to-End Sentence Level Lipreading”. In: *2020 IEEE Security and Privacy Workshops (SPW)*. IEEE.

Saumya Jetley, Nicholas A Lord, Namhoon Lee, and Philip HS Torr (2018). “Learn to pay attention”. In: *Proceedings of the International Conference on Learning Representations*.

Abhishek Jha, Vinay Namboodiri, and C. Jawahar (Mar. 2019). “Spotting words in silent speech videos: a retrieval-based approach”. In: *Machine Vision and Applications*.

Abhishek Jha, Vinay P. Namboodiri, and C. V. Jawahar (2018). “Word spotting in silent lip videos”. In: *IEEE Winter Conference on Applications of Computer Vision*.

Tao Jiang, Necati Cihan Camgoz, and Richard Bowden (2021). “Looking for the Signs: Identifying Isolated Sign Instances in Continuous Video Footage”. In: *IEEE International Conferene on Automatic Face and Gesture Recognition*.

Peiqi Jiao, Yuecong Min, Yanan Li, Xiaotao Wang, Lei Lei, and Xilin Chen (2023). “CoSign: Exploring Co-occurrence Signals in Skeleton-based Continuous Sign Language Recognition”. In: *ICCV*.

Tao Jin and Zhou Zhao (2021). “Contrastive Disentangled Meta-Learning for Signer-Independent Sign Language Translation”. In: *Proceedings of the 29th ACM International Conference on Multimedia*.

Benjamin Johnston and Philip Chazal (Sept. 2018). “A review of image-based automatic facial landmark identification techniques”. In: *EURASIP Journal on Image and Video Processing*.

Trevor Johnston (2012). “Lexical frequency in sign languages”. In: *Journal of deaf studies and deaf education*.

Armand Joulin, Francis Bach, and Jean Ponce (2010). “Discriminative clustering for image co-segmentation”. In: *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. IEEE.

Armand Joulin, Kevin Tang, and Li Fei-Fei (2014). “Efficient image and video co-localization with frank-wolfe algorithm”. In: *European Conference on Computer Vision*. Springer.

Hamid Reza Vaezi Joze and Oscar Koller (2019). “MS-ASL: A Large-Scale Data Set and Benchmark for Understanding American Sign Language”. In: *BMVC*.

Biing-Hwang Juang and Lawrence R Rabiner (2005). “Automatic speech recognition–a brief history of the technology development”. In: *Georgia Institute of Technology. Atlanta Rutgers University and the University of California. Santa Barbara*.

Anjuli Kannan, Yonghui Wu, Patrick Nguyen, Tara N. Sainath, Zhifeng Chen, and Rohit Prabhavalkar (2018). “An analysis of incorporating an external language

model into a sequence-to-sequence model". In: *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*.

Shigeki Karita, Nelson Yalta, Shinji Watanabe, M. Delcroix, A. Ogawa, and T. Nakatani (2019). "Improving Transformer-Based End-to-End Speech Recognition with Connectionist Temporal Classification and Language Model Integration". In: *INTERSPEECH*.

Angelos Katharopoulos, Apoorv Vyas, Nikolaos Pappas, and François Fleuret (2020). "Transformers are rnns: Fast autoregressive transformers with linear attention". In: *International Conference on Machine Learning*. PMLR.

Will Kay, Joao Carreira, Karen Simonyan, Brian Zhang, Chloe Hillier, Sudheendra Vijayanarasimhan, Fabio Viola, Tim Green, Trevor Back, Paul Natsev, et al. (2017). "The kinetics human action video dataset". In: *arXiv preprint arXiv:1705.06950*.

Daniel Kelly, John Mc Donald, and Charles Markham (2010). "Weakly supervised training of a sign language recognition system using multiple instance learning density matrices". In: *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*.

Adam Kendon (2004). *Gesture: Visible action as utterance*. Cambridge University Press.

Sanshzar Kettebekov, Mohammed Yeasin, and Rajeev Sharma (2003). "Improving continuous gesture recognition with spoken prosody". In: *Computer Vision and Pattern Recognition, 2003. Proceedings. 2003 IEEE Computer Society Conference on*. IEEE.

Gunhee Kim, Eric P Xing, Li Fei-Fei, and Takeo Kanade (2011). "Distributed cosegmentation via submodular optimization on anisotropic diffusion". In: *2011 international conference on computer vision*. IEEE.

Minsu Kim, Chae Won Kim, and Yong Man Ro (2023a). "Deep visual forced alignment: learning to align transcription with talking face video". In: *Proceedings of the AAAI Conference on Artificial Intelligence*.

Minsu Kim, Jeong Hun Yeo, Jeongsoo Choi, and Yong Man Ro (2023b). "Lip reading for low-resource languages by learning and combining general speech knowledge

and language-specific knowledge". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*.

Taehwan Kim, Jonathan Keane, Weiran Wang, Hao Tang, Jason Riggle, Gregory Shakhnarovich, Diane Brentari, and Karen Livescu (2017). "Lexicon-free fingerspelling recognition from video: Data, models, and signer adaptation". In: *Computer Speech & Language*.

Taejun Kim and Juhan Nam (2019). "Temporal Feedback Convolutional Recurrent Neural Networks for Keyword Spotting". In: *arXiv:1911.01803*.

Diederik P Kingma and Jimmy Ba (2015). "ADAM: A method for stochastic optimization". In: *Proceedings of the International Conference on Learning Representations*.

Diederik P. Kingma and Max Welling (2014). "Auto-Encoding Variational Bayes". In: *Proceedings of the International Conference on Learning Representations*.

Sang-Ki Ko, Chang Jo Kim, Hyedong Jung, and Choongsang Cho (2019). "Neural Sign Language Translation based on Human Keypoint Estimation". In: *Appl. Sci.*

Oscar Koller (2020). "Quantitative survey of the state of the art in sign language recognition". In: *arXiv preprint arXiv:2008.09918*.

Oscar Koller, Jens Forster, and Hermann Ney (2015a). "Continuous sign language recognition: Towards large vocabulary statistical recognition systems handling multiple signers". In: *Computer Vision and Image Understanding*.

Oscar Koller, Hermann Ney, and Richard Bowden (2015b). "Deep Learning of Mouth Shapes for Sign Language". In: *2015 IEEE International Conference on Computer Vision Workshop (ICCVW)*.

Oscar Koller, O Zargaran, Hermann Ney, and Richard Bowden (2016). "Deep sign: Hybrid CNN-HMM for continuous sign language recognition". In: *BMVC*.

Oscar Koller, Sepehr Zargaran, and Hermann Ney (2017). "Re-Sign: Re-Aligned End-to-End Sequence Modelling with Deep Recurrent CNN-HMMs". In: *CVPR*.

Alexandros Koumparoulis, Gerasimos Potamianos, Youssef Mroueh, and Steven J. Rennie (2017). "Exploring ROI size in deep learning based lipreading". In: *Proc. The 14th International Conference on Auditory-Visual Speech Processing*. URL: http://dx.doi.org/10.21437/AVSP.2017-13.

Hendrik Laux, Emil Mededovic, Ahmed Hallawa, Lukas Martin, Arne Peine, and Anke Schmeink (2023). “LiteVSR: Efficient Visual Speech Recognition by Learning from Speech Representations of Unlabeled Data”. In: *arXiv preprint arXiv:2312.09727*.

Matthew Le, Apoorv Vyas, Bowen Shi, Brian Karrer, Leda Sari, Rashel Moritz, Mary Williamson, Vimal Manohar, Yossi Adi, Jay Mahadeokar, et al. (2023). “Voicebox: Text-Guided Multilingual Universal Speech Generation at Scale”. In: *arXiv preprint arXiv:2306.15687*.

Dong-Hyun Lee et al. (2013). “Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks”. In: *Workshop on challenges in representation learning, ICML*.

Sangho Lee, Jiwan Chung, Youngjae Yu, Gunhee Kim, Thomas Breuel, Gal Chechik, and Yale Song (2021). “ACAV100M: Automatic Curation of Large-Scale Datasets for Audio-Visual Video Representation Learning”. In: *ICCV*.

Yong-Hyeok Lee and Namhyun Cho (2023). “Phonmatchnet: phoneme-guided zero-shot keyword spotting for user-defined keywords”. In: *arXiv preprint arXiv:2308.16511*.

Chris Lengerich and Awni Hannun (2016). “An End-to-End Architecture for Keyword Spotting and Voice Activity Detection”. In: *NIPS 2016 End-to-End Learning for Speech and Audio Processing Workshop*.

Juan León-Alcázar, Fabian Caba Heilbron, Ali Thabet, and Bernard Ghanem (2021). “MAAS: Multi-modal Assignation for Active Speaker Detection”. In: *Proceedings of the International Conference on Computer Vision*.

Dongxu Li, Cristian Rodriguez Opazo, Xin Yu, and Hongdong Li (2019). “Word-level Deep Sign Language Recognition from Video: A New Large-scale Dataset and Methods Comparison”. In: *WACV*.

Dongxu Li, Chenchen Xu, Liu Liu, Yiran Zhong, Rong Wang, Lars Petersson, and Hongdong Li (2021). “Transcribing Natural Languages for The Deaf via Neural Editing Programs”. In: *arXiv preprint arXiv:2112.09600*.

Dongxu Li, Chenchen Xu, Xin Yu, Kaihao Zhang, Ben Swift, Hanna Suominen, and Hongdong Li (2020a). “Tspnet: Hierarchical feature learning via temporal semantic pyramid for sign language translation”. In: *arXiv preprint arXiv:2010.05468*.

Dongxu Li, Xin Yu, Chenchen Xu, Lars Petersson, and Hongdong Li (2020b). "Transferring cross-domain knowledge for video sign language recognition". In: *CVPR*.

Hui Li, Peng Wang, and Chunhua Shen (2017). "Towards end-to-end text spotting with convolutional recurrent neural networks". In: *Proceedings of the IEEE international conference on computer vision*.

Jinyu Li et al. (2022). "Recent advances in end-to-end automatic speech recognition". In: *APSIPA Transactions on Signal and Information Processing*.

Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi (2023). *BLIP-2: Bootstrapping Language-Image Pre-training with Frozen Image Encoders and Large Language Models*. arXiv: 2301.12597 [cs.CV].

Junnan Li, Dongxu Li, Caiming Xiong, and Steven C. H. Hoi (2022). "BLIP: Bootstrapping Language-Image Pre-training for Unified Vision-Language Understanding and Generation". In: *ICML*.

Wei Li, Sicheng Wang, Ming Lei, Sabato Marco Siniscalchi, and Chin-Hui Lee (2019). "Improving Audio-visual Speech Recognition Performance with Cross-modal Student-teacher Training". In: *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE.

Jiachen Lian, Alexei Baevski, Wei-Ning Hsu, and Michael Auli (2023). "AV-data2vec: Self-supervised Learning of Audio-Visual Speech Representations with Contextualized Target Representations". In: *arXiv preprint arXiv:2302.06419*.

Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick (2014). "Microsoft coco: Common objects in context". In: *ECCV*.

Meng Liu, Xiang Wang, Liqiang Nie, Xiangnan He, Baoquan Chen, and Tat-Seng Chua (2018). "Attentive moment retrieval in videos". In: *ACM SIGIR*.

Peng Liu and Zuoying Wang (2004). "Voice activity detection using visual information". In: *2004 IEEE International Conference on Acoustics, Speech, and Signal Processing*.

Qingju Liu, Andrew J. Aubrey, and Wenwu Wang (2014). "Interference Reduction in Reverberant Speech Separation With Visual Voice Activity Detection". In: *IEEE Transactions on Multimedia*.

Qingju Liu, Wenwu Wang, and Philip Jackson (2011). “A visual voice activity detection method with adaboosting”. In: *Sensor Signal Processing for Defence (SSPD 2011)*.

Xubo Liu, Egor Lakomkin, Konstantinos Vougioukas, Pingchuan Ma, Honglie Chen, Ruiming Xie, Morrie Doulaty, Niko Moritz, Jachym Kolar, Stavros Petridis, Maja Pantic, and Christian Fuegen (2023). “SynthVSR: Scaling Up Visual Speech Recognition With Synthetic Supervision”. In: *CVPR*.

Ze Liu, Jia Ning, Yue Cao, Yixuan Wei, Zheng Zhang, Stephen Lin, and Han Hu (2022). “Video Swin Transformer”. In: *CVPR*.

Karen Livescu, Ozgur Cetin, Mark Hasegawa-Johnson, Simon King, Chris Bartels, Nash Borges, Arthur Kantor, Partha Lal, Lisa Yung, Ari Bezman, et al. (2007). “Articulatory feature-based methods for acoustic and audio-visual speech recognition: Summary from the 2006 JHU summer workshop”. In: *2007 IEEE International Conference on Acoustics, Speech and Signal Processing-ICASSP'07*. IEEE.

Stephan Liwicki and Mark Everingham (2009). “Automatic recognition of fingerspelled words in british sign language”. In: *2009 IEEE computer society conference on computer vision and pattern recognition workshops*. IEEE.

Ilya Loshchilov and Frank Hutter (2017). “Decoupled weight decay regularization”. In: *arXiv preprint arXiv:1711.05101*.

Huaishao Luo, Lei Ji, Ming Zhong, Yang Chen, Wen Lei, Nan Duan, and Tianrui Li (2022). “CLIP4Clip: An empirical study of CLIP for end to end video clip retrieval and captioning”. In: *Neurocomputing*.

Pingchuan Ma, Alexandros Haliassos, Adriana Fernandez-Lopez, Honglie Chen, Stavros Petridis, and Maja Pantic (2023). “Auto-AVSR: Audio-Visual Speech Recognition with Automatic Labels”. In: *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*.

Pingchuan Ma, Rodrigo Mira, Stavros Petridis, Björn Schuller, and Maja Pantic (2021a). “LiRA: Learning Visual Speech Representations from Audio through Self-supervision”. In: *Interspeech*.

Pingchuan Ma, Stavros Petridis, and Maja Pantic (2021b). “Detecting Adversarial Attacks on Audiovisual Speech Recognition”. In: *ICASSP 2021-2021 IEEE*

International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE.

Pingchuan Ma, Stavros Petridis, and Maja Pantic (2021c). "End-to-end audio-visual speech recognition with conformers". In: *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE.

Pingchuan Ma, Stavros Petridis, and Maja Pantic (2022). "Visual speech recognition for multiple languages in the wild". In: *Nature Machine Intelligence*.

Andrew L. Maas, Ziang Xie, Dan Jurafsky, and Andrew Y. Ng (2015). "Lexicon-Free Conversational Speech Recognition with Neural Networks". In: *Proceedings the North American Chapter of the Association for Computational Linguistics*.

Takaki Makino, Hank Liao, Yannis Assael, Brendan Shillingford, Basilio Garcia, Otavio Braga, and Olivier Siohan (2019). "Recurrent Neural Network Transducer for Audio-Visual Speech Recognition". In: *Interspeech*.

Nguyen Manh Tien Anh and Thach Ho Sy (2024). "Improving Speech Recognition with Prompt-based Contextualized ASR and LLM-based Re-predictor". In: *Interspeech*.

Cynthia Matuszek, Liefeng Bo, Luke Zettlemoyer, and Dieter Fox (2014). "Learning from unscripted deictic gesture and language for human-robot interactions". In: *Proceedings of the AAAI Conference on Artificial Intelligence*.

Michael McAuliffe, Michaela Socolof, Sarah Mihuc, Michael Wagner, and Morgan Sonderegger (2017). "Montreal forced aligner: Trainable text-speech alignment using kaldi." In: *Interspeech*.

David McNeill (1992). *Hand and mind: What gestures reveal about thought*. University of Chicago press.

Yuecong Min, Aiming Hao, Xiujuan Chai, and Xilin Chen (2021). "Visual alignment constraint for continuous sign language recognition". In: *Proceedings of the IEEE/CVF international conference on computer vision*.

Abdelrahman Mohamed, Dmytro Okhonko, and Luke Zettlemoyer (2019). "Transformers with convolutional context for ASR". In: *arXiv preprint arXiv:1904.11660*.

Pavlo Molchanov, Xiaodong Yang, Shalini Gupta, Kihwan Kim, Stephen Tyree, and Jan Kautz (2016). "Online detection and classification of dynamic hand gestures

with recurrent 3d convolutional neural network". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*.

Liliane Momeni, Triantafyllos Afouras, Themos Stafylakis, Samuel Albanie, and Andrew Zisserman (2020a). "Seeing wake words: Audio-Visual Keyword Spotting". In: *BMVC*.

Liliane Momeni, Hannah Bull, K R Prajwal, Samuel Albanie, Gül Varol, and Andrew Zisserman (2022). "Automatic dense annotation of large-vocabulary sign language videos". In: *ECCV*.

Liliane Momeni, Gül Varol, Samuel Albanie, Triantafyllos Afouras, and Andrew Zisserman (2020b). "Watch, Read and Lookup: Learning to Spot Signs from Multiple Supervisors". In: *ACCV*.

Louis-Philippe Morency, Ariadna Quattoni, and Trevor Darrell (2007). "Latent-dynamic discriminative models for continuous gesture recognition". In: *CVPR*.

Amit Moryossef, Kayo Yin, Graham Neubig, and Yoav Goldberg (2021). "Data Augmentation for Sign Language Gloss Translation". In: *MTSUMMIT*.

Jacob L Newman and Stephen J Cox (2012). "Language identification using visual features". In: *IEEE Transactions on audio, speech, and language processing*.

Minh Hoai Nguyen, Lorenzo Torresani, Fernando De La Torre, and Carsten Rother (2009). "Weakly supervised discriminative localization and classification: a joint learning process". In: *2009 IEEE 12th International Conference on Computer Vision*. IEEE.

Kuniaki Noda, Yuki Yamaguchi, Kazuhiro Nakadai, Hiroshi G Okuno, Tetsuya Ogata, et al. (2014). "Lipreading using convolutional neural network." In: *Interspeech*.

E. Ong, H. Cooper, N. Pugeault, and R. Bowden (2012). "Sign Language Recognition using Sequential Pattern Trees". In: *CVPR*.

Eng-Jon Ong and Richard Bowden (2011). "Learning temporal signatures for Lip Reading". In: *2011 IEEE International Conference on Computer Vision Workshops (ICCV Workshops)*.

Eng-Jon Ong, Oscar Koller, Nicolas Pugeault, and Richard Bowden (2014). "Sign spotting using hierarchical sequential patterns with temporal intervals". In: *CVPR*.

Aaron van den Oord, Yazhe Li, and Oriol Vinyals (2018). “Representation learning with contrastive predictive coding”. In: *arXiv preprint arXiv:1807.03748*.

Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. (2023). “Dinov2: Learning robust visual features without supervision”. In: *arXiv preprint arXiv:2304.07193*.

Carol Padden and Darline Clark Gunsauls (2003). “How the Alphabet Came to Be Used in a Sign Language”. In: *Sign Language Studies*.

Dimitri Palaz, Gabriel Synnaeve, and Ronan Collobert (2016). “Jointly learning to locate and classify words using convolutional networks”. In: *INTERSPEECH*.

Vassil Panayotov, Guoguo Chen, Daniel Povey, and Sanjeev Khudanpur (2015). “Librispeech: an ASR corpus based on public domain audio books”. In: *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE.

George Papandreou, Athanassios Katsamanis, Vassilis Pitsikalis, and Petros Maragos (2009). “Adaptive multimodal fusion by uncertainty compensation with application to audiovisual speech recognition”. In: *Audio, Speech, and Language Processing, IEEE Transactions on*.

Foteini Patrona, Alexandros Iosifidis, Anastasios Tefas, Nikolaos Nikolaidis, and Ioannis Pitas (2016). “Visual Voice Activity Detection in the Wild”. In: *IEEE Transactions on Multimedia*.

Jeffrey Pennington, Richard Socher, and Christopher Manning (2014). “GloVe: Global Vectors for Word Representation”. In: *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Eric David Petajan (1984). *Automatic lipreading to enhance speech recognition (speech reading)*. University of Illinois at Urbana-Champaign.

S. Petridis and M. Pantic (2016). “Deep Complementary Bottleneck Features for Visual Speech Recognition”. In: *ICASSP*.

Stavros Petridis, Themos Stafylakis, Pingchuan Ma, Feipeng Cai, Georgios Tzimiropoulos, and Maja Pantic (2018a). “End-to-end Audiovisual Speech Recognition”. In: *CoRR*. arXiv: 1802.06424.

Stavros Petridis, Themos Stafylakis, Pingchuan Ma, Georgios Tzimiropoulos, and Maja Pantic (2018b). "Audio-Visual Speech Recognition with a Hybrid CTC/Attention Architecture". In: *IEEE Spoken Language Technology Workshop*.

Tomas Pfister, James Charles, and Andrew Zisserman (2013). "Large-scale Learning of Sign Language by Watching TV (Using Co-occurrences)." In: *BMVC*.

Tomas Pfister, James Charles, and Andrew Zisserman (2014). "Domain-adaptive Discriminative One-shot Learning of Gestures". In: *ECCV*.

G. Potamianos, C. Neti, G. Gravier, A. Garg, and A.W. Senior (2003). "Recent advances in the automatic recognition of audiovisual speech". In: *Proceedings of the IEEE*.

Gerasimos Potamianos, C. Neti, Juergen Luettin, and Iain Matthews (2004). "Audio-Visual Automatic Speech Recognition: An Overview". In: *Issues in audio-visual speech processing*.

Rohit Prabhavalkar, Takaaki Hori, Tara N Sainath, Ralf Schlüter, and Shinji Watanabe (2023). "End-to-end speech recognition: A survey". In: *IEEE/ACM Transactions on Audio, Speech, and Language Processing*.

K R Prajwal, Triantafyllos Afouras, and Andrew Zisserman (2022a). "Sub-word Level Lip Reading With Visual Attention". In: *CVPR*.

K R Prajwal, Triantafyllos Afouras, and Andrew Zisserman (2024). "Speech Recognition Models are Strong Lip-readers". In: *INTERSPEECH*.

K R Prajwal, Hannah Bull, Liliane Momeni, Samuel Albanie, Gül Varol, and Andrew Zisserman (2022b). "Weakly-supervised Fingerspelling Recognition in British Sign Language Videos". In: *BMVC*.

K R Prajwal, Liliane Momeni, Triantafyllos Afouras, and Andrew Zisserman (2021). "Visual Keyword Spotting with Attention". In: *BMVC*.

Nicolas Pugeault and Richard Bowden (2011). "Spelling it out: Real-time ASL fingerspelling recognition". In: *2011 IEEE International conference on computer vision workshops (ICCV workshops)*. IEEE.

Filip Radenovic, Abhimanyu Dubey, Abhishek Kadian, Todor Mihaylov, Simon Vandenhende, Yash Patel, Yi Wen, Vignesh Ramanathan, and Dhruv Mahajan (2023). "Filtering, Distillation, and Hard Negatives for Vision-Language Pre-Training". In: *arXiv*.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever (2021). "Learning Transferable Visual Models From Natural Language Supervision". In: *ICML*.

Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine McLeavey, and Ilya Sutskever (2022). "Robust speech recognition via large-scale weak supervision". In: *arXiv preprint arXiv:2212.04356*.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. (2019). "Language models are unsupervised multitask learners". In: *OpenAI blog*.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu (2020). "Exploring the limits of transfer learning with a unified text-to-text transformer". In: *The Journal of Machine Learning Research*.

Akam Rahimi, Triantafyllos Afouras, and Andrew Zisserman (2022). "Reading to Listen at the Cocktail Party: Multi-Modal Speech Separation". In: *CVPR*.

Javier Ramirez, Juan Manuel Górriz, and José Carlos Segura (2007). "Voice activity detection. fundamentals and speech recognition system robustness". In: *Robust speech recognition and understanding*.

Charles Raude, K R Prajwal, Liliane Momeni, Hannah Bull, Samuel Albanie, Andrew Zisserman, and Gül Varol (2024). "A Tale of Two Languages: Large-Vocabulary Continuous Sign Language Recognition from Spoken Language Supervision". In: *arXiv*.

Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi (2016). "You only look once: Unified, real-time object detection". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*.

Katrin Renz, Nicolaj Stache, Samuel Albanie, and Gül Varol (2021a). "Sign Segmentation with Temporal Convolutional Networks". In: *International Conference on Acoustics, Speech, and Signal Processing*.

Katrin Renz, Nicolaj Stache, Neil Fox, Gül Varol, and Samuel Albanie (2021b). "Sign Segmentation with Changepoint-Modulated Pseudo-Labelling". In: *Workshop on ChaLearn Looking at People Sign Language Recognition in the Wild, CVPR*. IEEE.

Rami Al-Rfou, Dokook Choe, Noah Constant, Mandy Guo, and Llion Jones (2019). "Character-level language modeling with deeper self-attention". In: *Proceedings of the AAAI Conference on Artificial Intelligence*.

Susanna Ricco and Carlo Tomasi (2009). "Fingerspelling recognition through classification of letter-to-letter transitions". In: *Asian conference on computer vision*. Springer.

Fred Richardson, Douglas Reynolds, and Najim Dehak (2015). "Deep neural network approaches to speaker and language recognition". In: *IEEE signal processing letters*.

Richard C Rose and Douglas B Paul (1990). "A hidden Markov model based keyword recognition system". In: *International Conference on Acoustics, Speech, and Signal Processing*. IEEE.

Andrew Rosenberg, Kartik Audhkhasi, Abhinav Sethy, Bhuvana Ramabhadran, and Michael Picheny (2017). "End-to-end speech recognition and keyword search on low-resource languages". In: *ICASSP*.

Joseph Roth, Sourish Chaudhuri, Ondrej Klejch, Radhika Marvin, Andrew Gallagher, Liat Kaver, Sharadh Ramaswamy, Arkadiusz Stopczynski, Cordelia Schmid, Zhonghua Xi, et al. (2020). "Ava active speaker: An audio-visual dataset for active speaker detection". In: *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE.

Carsten Rother, Tom Minka, Andrew Blake, and Vladimir Kolmogorov (2006). "Cosegmentation of image pairs by histogram matching-incorporating a global constraint into mrfs". In: *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*. IEEE.

Michael Rubinstein, Armand Joulin, Johannes Kopf, and Ce Liu (2013). "Unsupervised joint object discovery and segmentation in internet images". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*.

Tara N. Sainath and Carolina Parada (2015). "Convolutional neural networks for small-footprint keyword spotting". In: *INTERSPEECH*.

Hiroaki Sakoe and Seibi Chiba (1978). "Dynamic programming algorithm optimization for spoken word recognition". In: *IEEE TRANSACTIONS ON ACOUSTICS, SPEECH, AND SIGNAL PROCESSING*.

Elizabeth Salesky, Matthew Wiesner, Jacob Bremerman, Roldano Cattoni, Matteo Negri, Marco Turchi, Douglas W. Oard, and Matt Post (2021). "Multilingual TEDx Corpus for Speech Recognition and Translation". In: *INTERSPEECH*.

Ben Saunders, Necati Cihan Camgoz, and Richard Bowden (2020). "Progressive transformers for end-to-end sign language production". In: *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XI 16*. Springer.

Ben Saunders, Necati Cihan Camgoz, and Richard Bowden (2022). "Signing at scale: Learning to co-articulate signs for large-scale photo-realistic sign language production". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*.

Adam Schembri, Jordan Fenlon, Ramas Rentelis, and Kearsy Cormier (2017). *British Sign Language Corpus Project: A corpus of digital video data and annotations of British Sign Language 2008-2017 (Third Edition)*. URL: http://www.bslcorpusproject.org.

Adam Schembri and Trevor Johnston (2007). "Sociolinguistic variation in the use of fingerspelling in Australian Sign Language: A pilot study". In: *Sign Language Studies*.

Steffen Schneider, Alexei Baevski, Ronan Collobert, and Michael Auli (2019). "wav2vec: Unsupervised pre-training for speech recognition". In: *arXiv preprint arXiv:1904.05862*.

Dmitriy Serdyuk, Otavio Braga, and Olivier Siohan (2022). "Transformer-based video front-ends for audio-visual speech recognition". In: *arXiv preprint arXiv:2201.10439*.

Laura Sevilla-Lara, Shengxin Zha, Zhicheng Yan, Vedanuj Goswami, Matt Feiszli, and Lorenzo Torresani (2021). "Only Time Can Tell: Discovering Temporal Data for Temporal Modeling". In: *WACV*.

Rahul Sharma, Krishna Somandepalli, and Shrikanth Narayanan (2019). "Toward Visual Voice Activity Detection for Unconstrained Videos". In: *2019 IEEE International Conference on Image Processing (ICIP)*.

Rajeev Sharma, Jiongyu Cai, Srivat Chakravarthy, Indrajit Poddar, and Yogesh Sethi (2000). "Exploiting speech/gesture co-occurrence for improving continuous gesture recognition in weather narration". In: *Proceedings Fourth IEEE International Conference on Automatic Face and Gesture Recognition (Cat. No. PR00580)*. IEEE.

Xiaolong Shen and Zhedong Zheng anc Yi Yang (2022). "StepNet: Spatial-temporal Part-aware Network for Isolated Sign Language Recognition". In.

Bowen Shi, Diane Brentari, Greg Shakhnarovich, and Karen Livescu (2021). "Fingerspelling Detection in American Sign Language". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*.

Bowen Shi, Diane Brentari, Greg Shakhnarovich, and Karen Livescu (2022a). "Open-Domain Sign Language Translation Learned from Online Video". In: *EMNLP*.

Bowen Shi, Diane Brentari, Greg Shakhnarovich, and Karen Livescu (2022b). "Searching for fingerspelled content in American Sign Language". In: *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*.

Bowen Shi, Aurora Martinez Del Rio, Jonathan Keane, Jonathan Michaux, Diane Brentari, Greg Shakhnarovich, and Karen Livescu (2018). "American sign language fingerspelling recognition in the wild". In: *2018 IEEE Spoken Language Technology Workshop (SLT)*. IEEE.

Bowen Shi, Wei-Ning Hsu, Kushal Lakhotia, and Abdelrahman Mohamed (2022c). "Learning audio-visual speech representation by masked multimodal cluster prediction". In: *arXiv preprint arXiv:2201.02184*.

Bowen Shi and Karen Livescu (2017). "Multitask training with unlabeled data for end-to-end sign language fingerspelling recognition". In: *2017 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*. IEEE.

Bowen Shi, Aurora Martinez Del Rio, Jonathan Keane, Diane Brentari, Greg Shakhnarovich, and Karen Livescu (2019). "Fingerspelling recognition in the wild with iterative visual attention". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*.

Zhiyuan Shi, Timothy M Hospedales, and Tao Xiang (2013). "Bayesian joint topic modelling for weakly supervised object localisation". In: *Proceedings of the IEEE International Conference on Computer Vision*.

Brendan Shillingford, Yannis Assael, Matthew W. Hoffman, Thomas Paine, Cían Hughes, Utsav Prabhu, Hank Liao, Hasim Sak, Kanishka Rao, Lorrayne Bennett, Marie Mulville, Ben Coppin, Ben Laurie, Andrew Senior, and Nando de Freitas (2019). "Large-Scale Visual Speech Recognition". In: *INTERSPEECH*.

Spyridon Siatras, Nikos Nikolaidis, Michail Krinidis, and Ioannis Pitas (2009). "Visual Lip Activity Detection and Speaker Detection Using Mouth Region Intensities". In: *IEEE Transactions on Circuits and Systems for Video Technology*.

Ozge Mercanoglu Sincan, Necati Cihan Camgoz, and Richard Bowden (2023). "Is context all you need? Scaling Neural Sign Language Translation to Large Domains of Discourse". In: *ICCV, ACVR workshop*.

Ozge Mercanoglu Sincan, Necati Cihan Camgoz, and Richard Bowden (2024). "Using an LLM to Turn Sign Spottings into Spoken Language Sentences". In: *arXiv preprint arXiv:2403.10434*.

D. Sodoyer, B. Rivet, L. Girin, J.-L. Schwartz, and C. Jutten (2006). "An Analysis of Visual Speech Information Applied to Voice Activity Detection". In: *2006 IEEE International Conference on Acoustics Speech and Signal Processing Proceedings*.

David Sodoyer, Bertrand Rivet, Laurent Girin, Christophe Savariaux, jean-luc Schwartz, and Christian Jutten (Mar. 2009). "A study of lip movements during spontaneous dialog and its application to voice activity detection". In: *The Journal of the Acoustical Society of America*.

Kaitao Song, Xu Tan, Tao Qin, Jianfeng Lu, and Tie-Yan Liu (2020). "MPNet: Masked and permuted pre-training for language understanding". In: *NeurIPS*.

Speech Group at Carnegie Mellon University (2014). *CMU Pronouncing Dictionary*. http://www.speech.cs.cmu.edu/cgi-bin/cmudict.

Themos Stafylakis and Georgios Tzimiropoulos (2017). "Combining Residual Networks with LSTMs for Lipreading". In: *Interspeech*.

Themos Stafylakis and Georgios Tzimiropoulos (2018). “Zero-shot keyword spotting for visual speech recognition in-the-wild”. In: *Proceedings of the European Conference on Computer Vision (ECCV)*.

Thad Starner (1995). “Visual Recognition of American Sign Language Using Hidden Markov Models”. MA thesis. Massachusetts Institute of Technology.

Stephanie Stoll, Necati Cihan Camgoz, Simon Hadfield, and Richard Bowden (2020). “Text2Sign: towards sign language production using neural machine translation and generative adversarial networks”. In: *International Journal of Computer Vision*.

William H Sumby and Irwin Pollack (1954). “Visual contribution to speech intelligibility in noise”. In: *The journal of the acoustical society of america*.

Chen Sun, Austin Myers, Carl Vondrick, Kevin Murphy, and Cordelia Schmid (2019). “Videobert: A joint model for video and language representation learning”. In: *Proceedings of the IEEE/CVF international conference on computer vision*.

Ming Sun, Anirudh Raju, George Tucker, Sankaran Panchapagesan, Gengshen Fu, Arindam Mandal, Spyridon Matsoukas, Nikko Strom, and Shiv Vitaladevuni (2016). “Max-pooling loss training of long short-term memory networks for small-footprint keyword spotting”. In: *2016 IEEE Spoken Language Technology Workshop (SLT)*.

Ilya Sutskever, Oriol Vinyals, and Quoc V. Le (2014). “Sequence to sequence learning with neural networks”. In: *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2*. MIT Press.

Valerie Sutton (1990). *Lessons in sign writing*. SignWriting.

Rachel Sutton-Spence (2007). “Mouthings and Simultaneity in British Sign Language”. In: *Simultaneity in Signed Languages: Form and Function*. John Benjamins.

Rachel Sutton-Spence and Bencie Woll (1999a). *The Linguistics of British Sign Language: An Introduction*. Cambridge University Press.

Rachel Sutton-Spence and Bencie Woll (1999b). *The linguistics of British Sign Language: an introduction*. Cambridge University Press.

Shinichi Tamura and Shingo Kawasaki (1988). “Recognition of sign language motion images”. In: *Pattern recognition*.

Kevin Tang, Armand Joulin, Li-Jia Li, and Li Fei-Fei (2014). “Co-localization in real-world images”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*.

Ruijie Tao, Zexu Pan, Rohan Kumar Das, Xinyuan Qian, Mike Zheng Shou, and Haizhou Li (2021). “Is Someone Speaking? Exploring Long-term Temporal Features for Audio-visual Active Speaker Detection”. In: *Proceedings of the 29th ACM International Conference on Multimedia*.

Laia Tarrés, Gerard I Gállego, Amanda Duarte, Jordi Torres, and Xavier Giró-i-Nieto (2023). “Sign language translation from instructional videos”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*.

Andros Tjandra, Nayan Singhal, David Zhang, Ozlem Kalinli, Abdelrahman Mohamed, Duc Le, and Michael L. Seltzer (2023). “Massively Multilingual ASR on 70 Languages: Tokenization, Architecture, and Generalization Capabilities”. In: *ICASSP*.

Zhan Tong, Yibing Song, Jue Wang, and Limin Wang (2022). “Videomae: Masked autoencoders are data-efficient learners for self-supervised video pre-training”. In: *Advances in neural information processing systems*.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. (2023). “Llama: Open and efficient foundation language models”. In: *arXiv preprint arXiv:2302.13971*.

Maria Tsimpoukelli, Jacob L Menick, Serkan Cabi, SM Eslami, Oriol Vinyals, and Felix Hill (2021). “Multimodal few-shot learning with frozen language models”. In: *NeurIPS*.

Aäron van den Oord, Yazhe Li, and Oriol Vinyals (2018). “Representation Learning with Contrastive Predictive Coding”. In: *arXiv preprint arXiv:1807.03748*.

Gül Varol, Liliane Momeni, Samuel Albanie, Triantafyllos Afouras, and Andrew Zisserman (2021). “Read and Attend: Temporal Localisation in Sign Language Videos”. In: *CVPR*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin (2017). “Attention Is All You Need”. In: *Advances in Neural Information Processing Systems*.

Ville Viitaniemi, Tommi Jantunen, Leena Savolainen, Matti Karppa, and Jorma Laaksonen (2014). “S-pot – a benchmark in spotting signs within continuous signing”. In: *LREC*.

Van Huy Vo, Elena Sizikova, Cordelia Schmid, Patrick Pérez, and Jean Ponce (2021). “Large-scale unsupervised object discovery”. In: *Advances in Neural Information Processing Systems*.

Christian Vogler and Dimitris Metaxas (1997). “Adapting hidden Markov models for ASL recognition by using three-dimensional computer vision methods”. In: *1997 IEEE International Conference on Systems, Man, and Cybernetics. Computational Cybernetics and Simulation*. IEEE.

Christian Vogler and Dimitris Metaxas (1998). “ASL recognition based on a coupling between HMMs and 3D motion analysis”. In: *Sixth International Conference on Computer Vision (IEEE Cat. No. 98CH36271)*. IEEE.

Christian Vogler and Dimitris Metaxas (2001). “A framework for recognizing the simultaneous aspects of american sign language”. In: *Computer Vision and Image Understanding*.

Christian Vogler and Dimitris Metaxas (2003). “Handshapes and movements: Multiple-channel american sign language recognition”. In: *International Gesture Workshop*. Springer.

U. von Agris, M. Knorr, and K. Kraiss (2008). “The significance of facial features for automatic sign language recognition”. In: *8th IEEE International Conference on Automatic Face Gesture Recognition*.

Petra Wagner, Zofia Malisz, and Stefan Kopp (2014). *Gesture and speech in interaction: An overview*.

Harry Walsh, Ozge Mercanoglu Sincan, Ben Saunders, and Richard Bowden (2023). “Gloss Alignment using Word Embeddings”. In: *2023 IEEE International Conference on Acoustics, Speech, and Signal Processing Workshops (ICASSPW)*. IEEE.

J. Wan, S. Z. Li, Y. Zhao, S. Zhou, I. Guyon, and S. Escalera (2016). “ChaLearn Looking at People RGB-D Isolated and Continuous Datasets for Gesture Recognition”. In: *IEEE Conference on Computer Vision and Pattern Recognition Workshops*.

Michael Wand, Jan Koutník, and Jürgen Schmidhuber (2016). "Lipreading with long short-term memory". In: *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE.

Changhan Wang, Morgane Riviere, Ann Lee, Anne Wu, Chaitanya Talnikar, Daniel Haziza, Mary Williamson, Juan Pino, and Emmanuel Dupoux (Aug. 2021). "VoxPopuli: A Large-Scale Multilingual Speech Corpus for Representation Learning, Semi-Supervised Learning and Interpretation". In: *ACL*.

Chengyi Wang, Sanyuan Chen, Yu Wu, Ziqiang Zhang, Long Zhou, Shujie Liu, Zhuo Chen, Yanqing Liu, Huaming Wang, Jinyu Li, et al. (2023). "Neural codec language models are zero-shot text to speech synthesizers". In: *arXiv preprint arXiv:2301.02111*.

Chong Wang, Weiqiang Ren, Kaiqi Huang, and Tieniu Tan (2014). "Weakly supervised object localization with latent category learning". In: *European Conference on Computer Vision*. Springer.

Limin Wang, Bingkun Huang, Zhiyu Zhao, Zhan Tong, Yinan He, Yi Wang, Yali Wang, and Yu Qiao (2023). "Videomae v2: Scaling video masked autoencoders with dual masking". In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*.

Mengmeng Wang, Jiazheng Xing, and Yong Liu (2021). "ActionCLIP: A New Paradigm for Video Action Recognition". In: *arXiv:2109.08472*.

Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He (2018). "Non-local Neural Networks". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

Yi Wang, Kunchang Li, Yizhuo Li, Yinan He, Bingkun Huang, Zhiyu Zhao, Hongjie Zhang, Jilan Xu, Yi Liu, Zun Wang, Sen Xing, Guo Chen, Junting Pan, Jiashuo Yu, Yali Wang, Limin Wang, and Yu Qiao (2022). "InternVideo: General Video Foundation Models via Generative and Discriminative Learning". In: *arXiv preprint arXiv:2212.03191*.

Yuxuan Wang, Pascal Getreuer, Thad Hughes, Richard Lyon, and Rif Saurous (2017). "Trainable frontend for robust and far-field keyword spotting". In: *ICASSP*.

Fangyun Wei and Yutong Chen (2023). "Improving Continuous Sign Language Recognition with Cross-Lingual Signs". In: *ICCV*.

R. B. Wilbur and A. C. Kak (2006). “Purdue RVL-SLLL American Sign Language Database”. In: *Technical Report*.

Ronnie B Wilbur (2013). “Phonological and prosodic layering of nonmanuals in American Sign Language”. In: *The signs of language revisited*. Psychology Press.

Jay Wilpon, Chin-Hui Lee, and Lawrence Rabiner (June 1989). “Application of hidden Markov models for recognition of a limited set of words in unconstrained speech”. In: *International Conference on Acoustics, Speech, and Signal Processing,*

Bencie Woll (2001). “he sign that dares to speak its name: echo phonology in British Sign Language (BSL)”. In: *P. Boyes Braem & R. Sutton-Spence (eds.)*

Ryan Wong, Necati Cihan Camgoz, and Richard Bowden (2023). “Learnt Contrastive Concept Embeddings for Sign Recognition”. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*.

World Federation of the Deaf (n.d.). Available at http://wfdeaf.org/our-work/, accessed: 30-09-2020.

Bichen Wu, Chenfeng Xu, Xiaoliang Dai, Alvin Wan, Peizhao Zhang, Masayoshi Tomizuka, Kurt Keutzer, and Peter Vajda (2020). “Visual Transformers: Token-based Image Representation and Processing for Computer Vision”. In: *CoRR*.

Pingping Wu, Hong Liu, Xiaofei Li, Ting Fan, and Xuewu Zhang (2016). “A Novel Lip Descriptor for Audio-Visual Keyword Spotting Based on Adaptive Decision Fusion”. In: *IEEE Transactions on Multimedia*.

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V. Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, Jeff Klingner, Apurva Shah, Melvin Johnson, Xiaobing Liu, Lukasz Kaiser, Stephan Gouws, Yoshikiyo Kato, Taku Kudo, Hideto Kazawa, Keith Stevens, George Kurian, Nishant Patil, Wei Wang, Cliff Young, Jason Smith, Jason Riesa, Alex Rudnick, Oriol Vinyals, Greg Corrado, Macduff Hughes, and Jeffrey Dean (2016). “Google’s Neural Machine Translation System: Bridging the Gap between Human and Machine Translation”. In: *CoRR*.

Bo Xu, Cheng Lu, Yandong Guo, and Jacob Wang (2020). “Discriminative Multi-Modality Speech Recognition”. In: *CVPR*.

Huijuan Xu, Kun He, Bryan A Plummer, Leonid Sigal, Stan Sclaroff, and Kate Saenko (2019). "Multilevel language and vision integration for text-to-clip retrieval". In: *AAAI*.

Runhua Xu, Nathalie Baracaldo, and James Joshi (2021). "Privacy-preserving machine learning: Methods, challenges and directions". In: *arXiv preprint arXiv:2108.04417*.

Hee-Deok Yang, Stan Sclaroff, and Seong-Whan Lee (2008). "Sign language spotting with a threshold model based on conditional random fields". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Ming-Hsuan Yang, Narendra Ahuja, and Mark Tabb (2002). "Extraction of 2d motion trajectories and its application to hand gesture recognition". In: *IEEE Transactions on pattern analysis and machine intelligence*.

Polina Yanovich, Carol Neidle, and Dimitris Metaxas (2016). "Detection of major ASL sign types in continuous signing for ASL recognition". In: *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*.

Yue Yao, Tianyu Wang, Heming Du, Liang Zheng, and Tom Gedeon (2019). "Spotting Visual Keywords from Temporal Sliding Windows". In: *Mandarin Audio-Visual Speech Recognition Challenge*.

Zhuyu Yao, Jiangbo Ai, Boxun Li, and Chi Zhang (2021). "Efficient DETR: Improving End-to-End Object Detector with Dense Prior". In: *CoRR*.

David Yarowsky (1995). "Unsupervised word sense disambiguation rivaling supervised methods". In: *33rd annual meeting of the association for computational linguistics*.

Jeong Hun Yeo, Minsu Kim, and Yong Man Ro (2023). "Multi-temporal lip-audio memory for visual speech recognition". In: *ICASSP*. IEEE.

Jeong Hun Yeo, Minsu Kim, Shinji Watanabe, and Yong Man Ro (2024). "Visual Speech Recognition for Languages with Limited Labeled Data Using Automatic Labels from Whisper". In: *ICASSP*.

Aoxiong Yin, Tianyun Zhong, Li Tang, Weike Jin, Tao Jin, and Zhou Zhao (2023). "Gloss attention for gloss-free sign language translation". In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*.

Kayo Yin and Jesse Read (2020). "Better Sign Language Translation with STMC-Transformer". In: *COLING*.

Youngwoo Yoon, Bok Cha, Joo-Haeng Lee, Minsu Jang, Jaeyeon Lee, Jaehong Kim, and Geehyuk Lee (2020). “Speech gesture generation from the trimodal context of text, audio, and speaker identity”. In: *ACM Transactions on Graphics (TOG)*.

Jianwei Yu, Shi-Xiong Zhang, Jian Wu, Shahram Ghorbani, Bo Wu, Shiyin Kang, Shansong Liu, Xunying Liu, Helen Meng, and Dong Yu (May 2020). “Audio-Visual Recognition of Overlapped Speech for the LRS2 Dataset”. In: *ICASSP*.

Yitian Yuan, Tao Mei, and Wenwu Zhu (2019). “To find where you talk: Temporal sentence localization in video with attention based location regression”. In: *AAAI*.

Yuhui Yuan and Jingdong Wang (2018). “Ocnet: Object context network for scene parsing”. In: *arXiv preprint arXiv:1809.00916*.

Runhao Zeng, H. Xu, W. Huang, Peihao Chen, Mingkui Tan, and Chuang Gan (2020). “Dense Regression Network for Video Grounding”. In: *CVPR*.

Haitong Zhang, Junbo Zhang, and Yujun Wang (2018). “Sequence-to-sequence Models for Small-Footprint Keyword Spotting”. In: *arXiv:1811.00348*.

Hang Zhang, Xin Li, and Lidong Bing (2023). “Video-LLaMA: An Instruction-tuned Audio-Visual Language Model for Video Understanding”. In: *arXiv preprint arXiv:2306.02858*. URL: https://arxiv.org/abs/2306.02858.

Huaiwen Zhang, Zihang Guo, Yang Yang, Xin Liu, and De Hu (2023). “C2ST: Cross-Modal Contextualized Sequence Transduction for Continuous Sign Language Recognition”. In: *ICCV*.

Jihai Zhang, Wengang Zhou, and Houqiang Li (2014). “A threshold-based hmm-dtw approach for continuous sign language recognition”. In: *Proceedings of International Conference on Internet Multimedia Computing and Service*.

Kaiwen Zhang, Kunchen Zhao, and Yunong Tian (2024). “Temporal–Semantic Aligning and Reasoning Transformer for Audio-Visual Zero-Shot Learning”. In: *Mathematics*.

Shifeng Zhang, Xiangyu Zhu, Zhen Lei, Hailin Shi, Xiaobo Wang, and Stan Z. Li (2017). “S3FD: Single Shot Scale-Invariant Face Detector”. In: *IEEE International Conference on Computer Vision (ICCV)*.

Xingxuan Zhang, Feng Cheng, and Shilin Wang (2019). “Spatio-temporal fusion based convolutional sequence learning for lip reading”. In: *Proceedings of the IEEE/CVF International conference on Computer Vision (ICCV)*.

Yaodong Zhang and James Glass (Jan. 2010). “Unsupervised spoken keyword spotting via segmental DTW on Gaussian Posteriorgrams”. In: *Proceedings of the 2009 IEEE Workshop on Automatic Speech Recognition and Understanding*.

Yuanhang Zhang, Shuang Yang, Jingyun Xiao, Shiguang Shan, and Xilin Chen (2020). “Can We Read Speech Beyond the Lips? Rethinking RoI Selection for Deep Visual Speech Recognition”. In: *arXiv preprint arXiv:2003.03206*.

Yundong Zhang, Naveen Suda, Liangzhen Lai, and Vikas Chandra (2017). “Hello Edge: Keyword Spotting on Microcontrollers”. In: *CoRR*.

Ya Zhao, Rui Xu, Xinchao Wang, Peng Hou, Haihong Tang, and Mingli Song (2020). “Hearing lips: Improving lip reading by distilling speech recognizers”. In: *Proceedings of the AAAI Conference on Artificial Intelligence*.

Yihao Zhi, Xiaodong Cun, Xuelin Chen, Xi Shen, Wen Guo, Shaoli Huang, and Shenghua Gao (2023). “Livelyspeaker: Towards semantic-aware co-speech gesture generation”. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*.

Benjia Zhou, Zhigang Chen, Albert Clapés, Jun Wan, Yanyan Liang, Sergio Escalera, Zhen Lei, and Du Zhang (2023). “Gloss-free Sign Language Translation: Improving from Visual-Language Pretraining”. In: *ICCV*.

Hao Zhou, Wen-gang Zhou, Weizhen Qi, Junfu Pu, and Houqiang Li (2020a). “Improving Sign Language Translation with Monolingual Data by Sign Back-Translation”. In: *CVPR*.

Hao Zhou, Wengang Zhou, Yun Zhou, and Houqiang Li (2020b). “Spatial-temporal multi-cue network for continuous sign language recognition”. In: *Proceedings of the AAAI Conference on Artificial Intelligence*.

Zhenxing Zhou, Vincent WL Tam, and Edmund Y Lam (2021). “SignBERT: A BERT-Based Deep Learning Framework for Continuous Sign Language Recognition”. In: *IEEE Access*.

Ziheng Zhou, Guoying Zhao, Xiaopeng Hong, and Matti Pietikäinen (2014). “A review of recent advances in visual speech decoding”. In: *Image and vision computing*.

Bin Zhu, Bin Lin, Munan Ning, Yang Yan, Jiaxi Cui, HongFa Wang, Yatian Pang, Wenhao Jiang, Junwu Zhang, Zongwei Li, et al. (2023). “Languagebind: Extending

video-language pretraining to n-modality by language-based semantic alignment". In: *arXiv preprint arXiv:2310.01852*.

Yimeng Zhuang, Xuankai Chang, Yanmin Qian, and Kai Yu (2016). "Unrestricted Vocabulary Keyword Spotting Using LSTM-CTC". In: *INTERSPEECH*.

Ronglai Zuo and Brian Mak (2022). "C2slr: Consistency-enhanced continuous sign language recognition". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*.

Chapter 11

Statement of Authorship

A statement of authorship is provided for each multi-authored paper included in this thesis. The statements describe the candidate's and co-authors' independent research contributions in the thesis publications. For each publication, there exists a complete statement that is filled out and signed by the candidate and supervisor.

Statement of Authorship for the paper: 'Sub-word Level Lip Reading With Visual Attention' in Chapter 2.

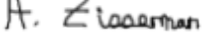
Paper title	Sub-word Level Lip Reading With Visual Attention
Authors	**K R Prajwal**, Triantafyllos Afouras, Andrew Zisserman.
Publication status	Published
Publication details	Computer Vision and Pattern Recognition (CVPR), 2022

Student confirmation

Student name	K R Prajwal
Contribution to the paper	• Joint conception of the idea • Research of prior work • Design and implementation of models • Running of all experiments • A portion of writing and presentation of the paper
Signature and Date	 October 10th 2024

Supervisor confirmation

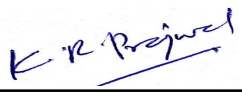
By signing the Statement of Authorship, the supervisor is certifying that the candidate made a substantial contribution to the publication, and that the description above is accurate.

Supervisor name	Prof. Andrew Zisserman
Supervisor comments	
Signature and Date	 10/10/2024

Statement of Authorship for the paper: 'Speech Recognition Models are Strong Lip-readers' in Chapter 3

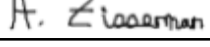
Paper title	Speech Recognition Models are Strong Lip-readers
Authors	**K R Prajwal**, Triantafyllos Afouras, Andrew Zisserman.
Publication status	Published
Publication details	INTERSPEECH, 2024

Student confirmation

Student name	K R Prajwal
Contribution to the paper	• Joint conception of the idea • Research of prior work • Design of models and evaluations • Running part of the experiments and sign language data processing • Writing and presentation of the paper
Signature and Date	 October 10th 2024

Supervisor confirmation

By signing the Statement of Authorship, the supervisor is certifying that the candidate made a substantial contribution to the publication, and that the description above is accurate.

Supervisor name	Prof. Andrew Zisserman
Supervisor comments	
Signature and Date	 10/10/2024

Statement of Authorship for the paper: 'Scaling Multilingual Visual Speech Recognition' in Chapter 4

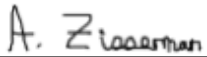
Paper title	Scaling Multilingual Visual Speech Recognition
Authors	**K R Prajwal***, Sindhu Hegde*, Andrew Zisserman.
Publication status	Published
Publication details	ICASSP 2025

Student confirmation

Student name	K R Prajwal
Contribution to the paper	• Joint conception of the idea • Research of prior work • Design and implementation of our new approach • A portion of writing and presentation of the paper
Signature and Date	 October 10th 2024

Supervisor confirmation

By signing the Statement of Authorship, the supervisor is certifying that the candidate made a substantial contribution to the publication, and that the description above is accurate.

Supervisor name	Prof. Andrew Zisserman
Supervisor comments	
Signature and Date	 10/10/2024

Statement of Authorship for the paper: 'Visual Keyword Spotting with Attention' in Chapter 5

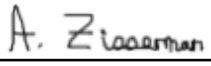
Paper title	Visual Keyword Spotting with Attention
Authors	**K R Prajwal***, Liliane Momeni*, Triantafyllos Afouras, Andrew Zisserman.
Publication status	Published
Publication details	British Machine Vision Conference, 2021.

Student confirmation

Student name	K R Prajwal
Contribution to the paper	• Joint conception of the idea • Research of prior work • Design and implementation of models • Running of experiments • A part of writing and presentation of the paper
Signature and Date	 October 10th 2024

Supervisor confirmation

By signing the Statement of Authorship, the supervisor is certifying that the candidate made a substantial contribution to the publication, and that the description above is accurate.

Supervisor name	Prof. Andrew Zisserman
Supervisor comments	
Signature and Date	 10/10/2024

Statement of Authorship for the paper: 'Read and Attend: Temporal Localisation in Sign Language Videos' in Chapter 6.

Paper title	Read and Attend: Temporal Localisation in Sign Language Videos
Authors	Gül Varol*, **Liliane Momeni***, Samuel Albanie*, Triantafyllos Afouras*, Andrew Zisserman.
Publication status	Published
Publication details	Conference on Computer Vision and Pattern Recognition, 2021.

Student confirmation

Student name	Liliane Momeni
Contribution to the paper	• Joint conception of the idea • Research of prior work • Design and implementation of models • Running of experiments • Writing and presentation of the paper
Signature and Date	April 17th 2024

Supervisor confirmation

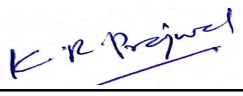
By signing the Statement of Authorship, the supervisor is certifying that the candidate made a substantial contribution to the publication, and that the description above is accurate.

Supervisor name	Prof. Andrew Zisserman
Supervisor comments	
Signature and Date	A. Zisserman 17/4/2024

Statement of Authorship for the paper: 'Automatic dense annotation of large-vocabulary sign language videos' in Chapter 6.

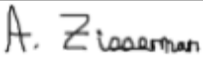
Paper title	Automatic dense annotation of large-vocabulary sign language videos
Authors	Liliane Momeni*, Hannah Bull*, **K R Prajwal***, Samuel Albanie, Gül Varol, Andrew Zisserman.
Publication status	Published
Publication details	European Conference on Computer Vision, 2022.

Student confirmation

Student name	K R Prajwal
Contribution to the paper	• Joint conception of the idea • Research of prior work • Design and implementation of mouthing-based spotting • Experiments with improving keyword spotting model • A part of writing and presentation of the paper
Signature and Date	 October 10th 2024

Supervisor confirmation

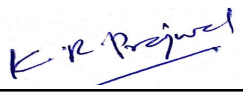
By signing the Statement of Authorship, the supervisor is certifying that the candidate made a substantial contribution to the publication, and that the description above is accurate.

Supervisor name	Prof. Andrew Zisserman
Supervisor comments	
Signature and Date	 10/10/2024

Statement of Authorship for the paper: 'Weakly-supervised Fingerspelling Recognition in British Sign Language' in Chapter 7.

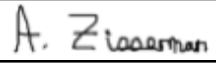
Paper title	Weakly-supervised Fingerspelling Recognition in British Sign Language
Authors	**K R Prajwal***, Hannah Bull*, Liliane Momeni*, Samuel Albanie, Gül Varol, Andrew Zisserman.
Publication status	Published
Publication details	British Machine Vision Conference, 2022.

Student confirmation

Student name	K R Prajwal
Contribution to the paper	• Joint conception of the idea • Research of prior work • Design and implementation of models • All training experiments and evaluation • Writing and presentation of the paper
Signature and Date	 October 10th 2024

Supervisor confirmation

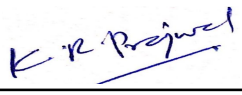
By signing the Statement of Authorship, the supervisor is certifying that the candidate made a substantial contribution to the publication, and that the description above is accurate.

Supervisor name	Prof. Andrew Zisserman
Supervisor comments	
Signature and Date	 10/10/2024

Statement of Authorship for the paper: 'A Large-Vocabulary Continuous Sign Language Recognition Benchmark' in Chapter 8

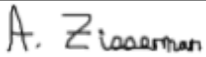
Paper title	A Large-Vocabulary Continuous Sign Language Recognition Benchmark
Authors	Charles Raude*, **K R Prajwal***, Liliane Momeni*, Hannah Bull, Samuel Albanie, Andrew Zisserman, Gül Varol.
Publication status	Unpublished manuscript
Publication details	

Student confirmation

Student name	K R Prajwal
Contribution to the paper	• Joint conception of the idea • Research of prior work • Design of models and the baseline approaches • Training better video features and implementing evaluation • Writing and presentation of the paper
Signature and Date	 October 10th 2024

Supervisor confirmation

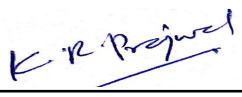
By signing the Statement of Authorship, the supervisor is certifying that the candidate made a substantial contribution to the publication, and that the description above is accurate.

Supervisor name	Prof. Andrew Zisserman
Supervisor comments	
Signature and Date	 10/10/2024

Statement of Authorship for the paper: 'Understanding Co-speech Gestures in-the-wild' in Chapter 9

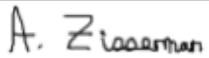
Paper title	Understanding Co-speech Gestures in-the-wild
Authors	Sindhu Hegde, **K R Prajwal**, Taein Kwon, Andrew Zisserman.
Publication status	Published
Publication details	International Conference on Computer Vision (ICCV), 2025

Student confirmation

Student name	K R Prajwal
Contribution to the paper	• Joint conception of the idea • Research of prior work • Evaluation and baselines • Writing and presentation of the paper
Signature and Date	 October 10th 2024

Supervisor confirmation

By signing the Statement of Authorship, the supervisor is certifying that the candidate made a substantial contribution to the publication, and that the description above is accurate.

Supervisor name	Prof. Andrew Zisserman
Supervisor comments	
Signature and Date	 10/10/2024