INFORMATICA, 2026, Vol. 0, No. 0, 1–27

DOI: https://doi.org/10.15388/26-INFOR632

Few-Shot Training of Prototype Networks for Sign Language Recognition

Michał KALINOWSKI, Bożena KOSTEK*

Gdansk University of Technology, Faculty of Electronics, Telecommunications and Informatics, Multimedia Systems Department and Audio Acoustics Laboratory, 80-222 Gdańsk, Poland
e-mail: michal.kalinowski@pg.edu.pl, bozena.kostek@pg.edu.pl

Received: March 2026; accepted: May 2026

Abstract. Limited proficiency in sign language creates communication barriers, motivating the development of robust Automatic Sign Language Recognition (SLR) systems. We address isolated SLR in a low-resource setting using few-shot metric-based meta-learning. Sign videos are encoded with spatiotemporal convolutional backbones and classified using a prototypical network, enabling generalization to unseen classes from small support sets. We compare the SlowFast architecture with state-of-the-art video models on the LSA64 benchmark under strict class-disjoint protocols. SlowFast achieves 94.33% accuracy, outperforming competing backbones and demonstrating an effective and data-efficient approach for low-resource isolated SLR.

Key words: Isolated Sign Language Recognition, few-shot learning, SlowFast, prototype networks, UMAP.

1. Introduction – Background, Current Approach, and Challenges in Sign Language Recognition and Translation

1.1. *Background*

According to a 2021 report by the World Health Organization (WHO), more than 1.5 billion people worldwide live with some degree of hearing loss, representing approximately 20% of the global population. Among them, 430 million individuals experience moderate or greater hearing loss, defined as $\geqslant$ 35 dB hearing level (dB HL), a condition that, if left untreated and without rehabilitation, significantly affects quality of life (Rastgoo *et al.*, 2021). The World Federation of the Deaf (WFD) estimates that the global deaf community includes approximately 70 million people who use a sign language as their primary means of communication.

Sign language is therefore a fundamental communication modality for deaf and hard-of-hearing individuals worldwide. Unlike spoken languages, sign languages are not globally standardized and exhibit substantial linguistic variation across regions. Even in countries where the same spoken language predominates, sign languages may differ signifi-

*Correponding author.

cantly. For example, American Sign Language (ASL) and British Sign Language (BSL) are not mutually intelligible, despite both being used in predominantly English-speaking countries (Emmorey, 2023). It is estimated that approximately 300 distinct sign languages are currently in use worldwide, e.g. American Sign Language (ASL), Japanese Sign Language, Hong Kong Sign Language, Swiss German Sign Language, etc. Moreover, a cross-linguistic variation in the use of prosodic cues, such as, e.g. eye blinks – to mark prosodic constituents in sign languages exists (Tang *et al.*, 2010).

This linguistic diversity creates persistent communication barriers – not only between sign language users and hearing individuals, but also among signers from different regions, sometimes even within the same country. Addressing these barriers requires scalable and adaptable technological solutions (de Amorim *et al.*, 2019). To this end, recent advances in machine learning and deep learning have opened promising avenues for sign language processing. Existing approaches typically rely on RGB video data (Zhou *et al.*, 2021a; Min *et al.*, 2021; Papastratis *et al.*, 2020; Ahn *et al.*, 2023; Sari *et al.*, 2023), skeletal (pose-based) representations (Ferreira *et al.*, 2022; Alsulami *et al.*, 2024; Boháček and Hrúz, 2023), or multimodal fusion of both. Additional modalities, such as optical flow, further enrich motion modelling and remain an active area of investigation.

There are two primary sub-tasks in sign language recognition (SLR), namely isolated SLR, which focuses on recognizing individual signs, and continuous sign language recognition (CSLR), which aims to interpret sequences of signs from uninterrupted video streams (Zhou *et al.*, 2021a). A widely adopted framework combines Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs) (Min *et al.*, 2021; Papastratis *et al.*, 2020), where CNNs extract spatial features from video frames and RNNs model temporal dependencies to produce gloss sequences (Papastratis *et al.*, 2020). More recently, Transformer-based architectures have gained popularity, particularly in Sign Language Translation (SLT), leveraging RGB inputs, pose-based representations, or hybrid modalities. Given the structural nature of skeletal data, graph neural networks have also been explored, drawing inspiration from advances in action recognition (de Amorim *et al.*, 2019).

Together, these developments highlight the rapid progress of data-driven approaches to SLR, while also underscoring the ongoing challenges posed by linguistic diversity, limited annotated resources, and the need for robust spatiotemporal modelling. Despite these advancements, the field continues to face significant challenges due to the scarcity of large-scale, well-annotated sign language datasets (Ferreira *et al.*, 2022; Alsulami *et al.*, 2024). Data collection is resource-intensive, requiring controlled recordings and detailed annotation performed by domain experts, often native signers. Consequently, only a limited number of sign languages are supported by sufficiently large corpora for data-hungry deep learning models. The choice of the dataset is inherently task-dependent: CSLR and SLT require continuous sequences with frame-level gloss annotations or sentence-level transcriptions, whereas isolated SLR relies on datasets with a single labelled sign per video clip.

To mitigate the limitations imposed by small datasets, few-shot learning has emerged as a promising paradigm in sign language processing. Originally developed for image classification under limited-sample conditions, few-shot techniques have been successfully adapted to sign language tasks. Among metric-based methods, prototypical networks

have proven particularly effective, representing each class by a prototype in an embedding space and encouraging intra-class compactness and inter-class separability (Snell *et al.*, 2017). Such approaches are especially well-suited to low-resource sign language settings, where the ability to generalize to novel classes from only a handful of labelled examples is essential.

In this work, the SlowFast network (Feichtenhofer *et al.*, 2019) is employed within a prototypical learning framework, motivated by its strong performance in continuous sign language recognition (CSLR) (Ahn *et al.*, 2023). To verify whether this choice generalizes to low-resource isolated SLR, we additionally evaluate several widely used spatiotemporal feature extractors, including S3D (Xie *et al.*, 2018), I3D (Carreira and Zisserman, 2018), R(2+1)D (Tran *et al.*, 2018), and (3+2+1)D ResNet (Zhou *et al.*, 2021b). The LSA64 dataset (Ronchetti *et al.*, 2023) is used in the experimental setup, with the primary goal of assessing and comparing the effectiveness of these feature extractors for isolated SLR under conditions of limited sample availability. This enables a controlled evaluation of whether architectures successful in large-scale CSLR remain effective in few-shot isolated recognition scenarios.

We start with a review of current works and systems in SLR that justifies the choice of the models used in our approach. We also discuss the main challenges in sign language recognition and translation. Next, we briefly introduce few-shot learning, a machine learning technique that enables generalization to unseen categories with only a few examples. This approach is motivated by data sparsity. After that, we describe the proposed methodology, specifically prototypical networks, followed by an explanation of the feature extraction process and the models used. The Experiments Section covers the dataset, data processing steps, training details, evaluation metrics, and the experimental setup and results. The results are shown using several evaluation metrics, along with Uniform Manifold Approximation and Projection (UMAP) visualizations of the learned embeddings. This is followed by a discussion, including the limitations of our approach. Finally, the Conclusion and Future Research Section summarizes the main findings and suggests directions for future work.

1.2. *Current Approach*

Modern transcription systems designed to support sign language are crucial for research on multimedia accessibility for the hearing impaired. These systems use solutions in image and sound analysis, gesture recognition, and real-time spoken language processing. The development of these technologies helps break down communication barriers and prevents social isolation for deaf individuals in daily life. It should, however, be noted that one of the first systems, namely, Hamburg Notation System for Sign Languages (HamNoSys) was developed in 1985 by a research team at the University of Hamburg. It was based on the phonological structure of the signed token theory (Hanke, 2004). Two other systems are also rooted decades back. This concerns Signing Gesture Markup Language (SiGML), a language that represents HamNoSys in code form. Tools exist that convert SiGML-encoded signs into animations depicting the performance of specific gestures,

Table 1
Overview of sign language technology systems.

Category	System	Description
AI-Driven Learning Platforms	SLAIT School (2024)	Interactive platform using webcam-based AI to analyse user signing performance in real time.
	SignAll (2026)	Multi-camera (2D/3D) system with depth sensing for sign-to-text translation and advanced training.
	Lingvano (2026)	Mobile application offering structured lessons with high-quality video demonstrations (ASL, BSL).
Real-Time Translation and Avatars	Hand Talk (2026)	Application with a 3D avatar translating speech/text into sign language in real time.
	Signapse AI (2024)	AI-driven photo-realistic avatars used for public announcements (e.g. transport systems).
	DeepSign AI (2026)	LLM-based system ensuring grammatically correct sign language translation beyond literal mapping.
Transcription and Notation Tools	ELAN (Kong, 2016)	Linguistic annotation tool for multi-layered transcription of sign language video data.
	SignBank (Sutton *et al.*, 2004)	Web-based lexical database supporting multiple sign notation systems and regional variations.
Polish Sign Language (PJM) Solutions	Migam (2025)	Platform providing on-demand video interpreting services with human translators.
	Migaj.eu (Łacheta *et al.*, 2016)	Digital dictionary and educational platform for Polish Sign Language with extensive video resources.

Table 2
Comparison of data processing.

Feature	Rule-Based (HamNoSys/SiGML)	AI-Based (Deep Learning)
Logic	Predetermined symbols and codes.	Patterns learned from thousands of videos.
Accuracy	High for specific, coded signs.	High for natural, fluid conversation.
Flexibility	Rigid; hard to “improvise”.	Adaptive; understands various accents/styles.
Use Case	Formal documentation, dictionaries.	Real-time translation, mobile apps.

making SiGML the preferred language for converting recognized signs (Kennaway, 2001). Among those, one can also list SignWriting, a section of Sutton Movement Writing used for recording sign language symbols. It was created in 1974 in Copenhagen by Valerie Sutton. Due to its universality, it can be adapted to record any sign language (Sutton, 1995). Among more up-to-date systems, several representative approaches in sign language technologies are summarized in Table 1.

These approaches are rule-based methods grounded in predefined symbolic representations. In contrast, modern sign language technologies increasingly rely on data-driven methods based on deep learning. A comparison of these two paradigms is summarized in Table 2.

However, despite the success of some of the above-mentioned applications, various methodological frameworks have been explored for sign language recognition (SLR). The selection of a particular approach typically depends on the specific problem context. Among the so-called traditional methods, combinations of Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs) (Min *et al.*, 2021; Papastratis

et al., 2020) have been widely used, particularly for continuous sign language recognition (CSLR), where modelling temporal dependencies is essential. Graph-based techniques (de Amorim *et al.*, 2019) are frequently applied in isolated SLR tasks or employed as feature extractors, leveraging skeletal representations. Transformer-based methods (Camgoz *et al.*, 2020; Ferreira *et al.*, 2022; Alsulami *et al.*, 2024), owing to their strong capability to capture long-range temporal dependencies, have demonstrated versatility across both isolated and continuous SLR tasks.

Feature extraction is a vital step in SLR, determining the quality of input to recognition systems. While graph-based methods align naturally with skeletal data, RGB-based approaches remain highly effective. In addition to SlowFast architecture, other spatiotemporal networks such as S3D (Xie *et al.*, 2018), I3D (Carreira and Zisserman, 2018), R(2+1)D (Tran *et al.*, 2018), and (3+2+1)D ResNet (Zhou *et al.*, 2021b) have demonstrated strong performance in action and gesture recognition tasks, with successful applications as feature extractors in continuous sign language recognition. Evaluating these architectures in a few-shot isolated SLR allows for a systematic comparison to determine whether the effectiveness of the SlowFast architecture generalizes across limited-data scenarios.

The SlowFast network (Feichtenhofer *et al.*, 2019), originally designed for video recognition, has been successfully adapted for SLR. In Ahn *et al.* (2023), a two-pathway SlowFast framework enhances CSLR by extracting spatial features (e.g. hand shapes, facial expressions) and dynamic features (e.g. arm movements). It incorporates Bi-directional Feature Fusion for pathway interaction and Pathway Feature Enhancement to improve feature quality during training without compromising inference speed. Similarly, Hassan *et al.* (2021) demonstrates the effectiveness of the SlowFast architecture in isolated SLR (Ahn *et al.*, 2023; Hassan *et al.*, 2021), justifying its choice as the feature extractor in this work. Moreover, its performance sets a benchmark for evaluating other recent spatiotemporal feature extractors (S3D (Xie *et al.*, 2018), I3D (Carreira and Zisserman, 2018), R(2+1)D (Tran *et al.*, 2018), and (3+2+1)D ResNet (Zhou *et al.*, 2021b)) under few-shot learning conditions.

The labour-intensive process of collecting sign language data, resulting in limited annotated samples, motivates the adoption of few-shot learning techniques. Few-shot learning has emerged as a promising approach in scenarios with limited data. Various few-shot techniques have been explored, including Matching Networks, Model-Agnostic Meta-Learning, and Prototypical Networks, applied to small datasets such as electromyograms of sign language performance (Ferreira *et al.*, 2022). Recent studies have integrated Transformer architectures with few-shot frameworks; for instance, Ferreira *et al.* (2022) applies a Transformer for skeletal data using a contrastive learning strategy, Boháček and Hrúz (2023) employs a Transformer network for similar purposes, and Alsulami *et al.* (2024) leverages a Transformer encoder for skeletal data derived from RGB inputs. The latter incorporates both embedding propagation and label propagation, achieving superior performance compared to prototypical approaches on selected datasets. Additionally, Sari *et al.* (2023) applies Shufflenet_V2 in a prototypical framework for few-shot Indonesian sign language recognition. While Bilge *et al.* (2023) explores zero-shot recognition us-

ing TSM, 3D CNN + BiLSTM, and BERT, its focus differs from few-shot learning but highlights related data scarcity solutions.

Building on previous work, this study employs SlowFast architecture (Feichtenhofer *et al.*, 2019) as the primary backbone while additionally evaluating several other spatiotemporal feature extractors (S3D (Xie *et al.*, 2018), I3D (Carreira and Zisserman, 2018), R(2+1)D (Tran *et al.*, 2018), and (3+2+1)D ResNet (Zhou *et al.*, 2021b)) in a few-shot isolated SLR framework. This systematic comparison examines whether architectures successful in CSLR retain their effectiveness when applied to low-resource isolated recognition, providing insights into feature extractor selection for future applications.

1.3. *Challenges in Sign Language Recognition and Translation*

An examination of the structural framework of sign language recognition (SLR) and sign language translation (SLT) methodologies highlights several inherent challenges in mapping visual sign sequences to spoken language. As illustrated in the example sequence (see Fig. 1), the transition from visual signals to semantic meaning involves more than direct pattern matching. Instead, it requires the modelling of linguistic structure, contextual dependencies, and multimodal cues. While sign language processing encompasses a broad range of problems, including translation and continuous sequence modelling, this work focuses on isolated SLR in a low-resource setting, with particular emphasis on data scarcity and spatiotemporal modelling, which define the primary focus of the challenges considered in this work.

Table 3 summarises the principal challenges in sign language recognition and translation discussed in this section. For each challenge, the table provides a concise description, representative evidence or examples reported in the literature, and its relevance to the present work. The summary further distinguishes between challenges that directly motivate the methodological choices adopted in this study, such as few-shot learning and high-capacity spatiotemporal encoders, and those that remain part of the broader context of sign language processing but are less directly related to isolated SLR.

Fig. 1. Qualitative example of a continuous sign language sequence from the RWTH-PHOENIX-Weather 2014 dataset (Sample ID: 15December_2010_Wednesday_tagesschau_default-6). To illustrate the linguistic and physical dynamics of the utterance, keyframe sampling was employed; specifically, frames f_1 through f_{102} were selected to highlight significant changes in motion, hand shape, and transition points (temporal flow). The hierarchy displays the alignment between the DGS Glosses (semantic labels), their English counterparts, and the final Spoken Language Translations in both German and English.

Table 3
Summary of the principal challenges in sign language recognition (SLR) and translation (SLT) discussed in Section 1.3.

Challenge	Description	Example/Evidence	Relevance to this work
Data Sparsity (Koller *et al.*, 2015; Camgoz *et al.*, 2018)	Sign language datasets are limited in vocabulary, domain, and signer diversity; out-of-vocabulary signs and dialectal or regional variation further reduce generalization.	Phoenix14 and Phoenix14T cover only German weather forecasts; in Polish Sign Language, city names can vary significantly between regions.	Directly addressed: motivates the adoption of a few-shot learning framework.
Spatiotemporal Complexity and Non-Manual Markers (NMMs) (Al Abdullah *et al.*, 2024)	Eyebrow movement, facial expression, and mouth articulation carry semantic and grammatical information that disambiguates signs with similar handshapes or spatial configurations.	Differences between frames f_{12} and f_{102} in Fig. 1 are reflected mainly in mouth and eye activity rather than in the hands; NMM cues raise accuracy by approximately 10–17%.	Directly addressed: motivates the use of high-capacity spatiotemporal encoders.
Intra-Class Variability	Differences in signer style, articulation speed, and execution introduce significant variability within the same sign class, requiring representations that are both discriminative and robust.	The same sign performed by different signers, or at different speeds, yields markedly different visual sequences.	Directly addressed: shapes the choice of prototype-based metric learning for low-resource generalization.
Modality Trade-off (RGB vs. Skeleton) (Moryossef *et al.*, 2021; Lu *et al.*, 2024)	RGB preserves rich visual information but is sensitive to lighting, occlusion, and background; skeletal pose simplifies learning but may miss subtle cues, especially under occlusion or extreme viewpoints.	Pose estimation can fail to capture mouth articulation or fine handshape detail, whereas RGB models remain vulnerable to environmental variability.	Directly addressed: this work adopts RGB-based spatiotemporal representations as best suited for isolated SLR.
Sequence Modelling and Linguistic Abstraction (Desai *et al.*, 2024)	Glosses are context-dependent linguistic abstractions, often mistakenly treated as complete translations; co-articulation between consecutive signs further obscures segmentation in continuous signing.	The DGS gloss sequence *STARK SCHNEE* (Strong Snow) must be rendered as *starkem Schneefall* (heavy snowfall) in German; in continuous signing, the boundary between successive signs is inherently ambiguous.	Broader context only: pertinent to continuous SLR and SLT, but not directly applicable to the isolated SLR setting considered here.

Taken together, the challenges summarised in Table 3 illustrate that sign language recognition encompasses a diverse set of challenges, ranging from data limitations to complex temporal and linguistic dependencies. Within this landscape, the present work concentrates on data sparsity and spatiotemporal modelling in isolated SLR. By adopting a few-shot learning framework combined with high-capacity spatiotemporal encoders, the approach aims to learn transferable representations that generalize to unseen sign classes despite limited supervision, while remaining compatible with the broader challenges outlined above.

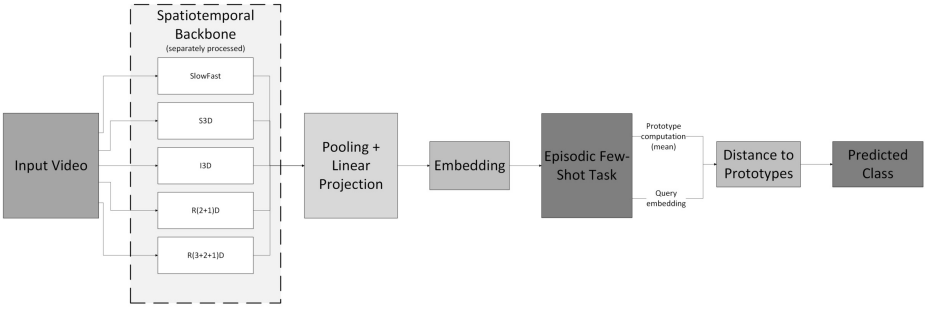

Fig. 2. Overview of the proposed few-shot sign language recognition framework. Input video sequences are processed by a spatiotemporal backbone network (e.g. SlowFast, S3D, I3D, R(2+1)D, or R(3+2+1)D) acting as a feature extractor. The resulting features are aggregated and projected into a fixed-dimensional embedding space. Training follows an episodic N-way K-shot setup, where class prototypes are computed from support embeddings, and query samples are classified based on their distance to these prototypes. All backbones are evaluated within this unified pipeline under identical conditions. The grayscale shading in the figure is used solely to visually distinguish the successive phases of the proposed pipeline – input/output stages, core processing stages, and intermediate transformation steps – and does not encode any additional information.

2. Methods

2.1. *Overview of the Proposed Approach*

The proposed framework addresses sign language recognition under limited data availability using a few-shot learning paradigm combined with prototypical classification in a shared embedding space. The overall framework is illustrated in Fig. 2 and consists of three stages: (i) feature extraction using a spatiotemporal backbone, followed by temporal pooling and linear projection into a fixed-dimensional embedding space, (ii) episodic few-shot task construction with support and query sets, where class prototypes are computed as the mean of support embeddings; and (iii) prototype-based classification of query samples using a distance measure in the embedding space.

Given a set of input sign language video samples, each video is first processed by a selected spatiotemporal backbone network (described in Section 2.4) that serves as a feature extractor. The backbone transforms the input sequence into a fixed-dimensional embedding representation capturing both spatial appearance and temporal dynamics. All backbone architectures are integrated into the same embedding and classification pipeline, differing only in the feature-extraction stage.

During training, the model follows an episodic few-shot learning strategy. Each episode is constructed from a small support set and a query set sampled from a subset of classes. The support set embeddings are used to compute class prototypes, defined as the mean vector of embeddings belonging to each class. Query samples are classified by assigning each query embedding to the class with the nearest prototype under a distance metric in the learned feature space (see Fig. 3). This formulation encourages the model to learn embeddings that preserve inter-class separability even under limited supervision.

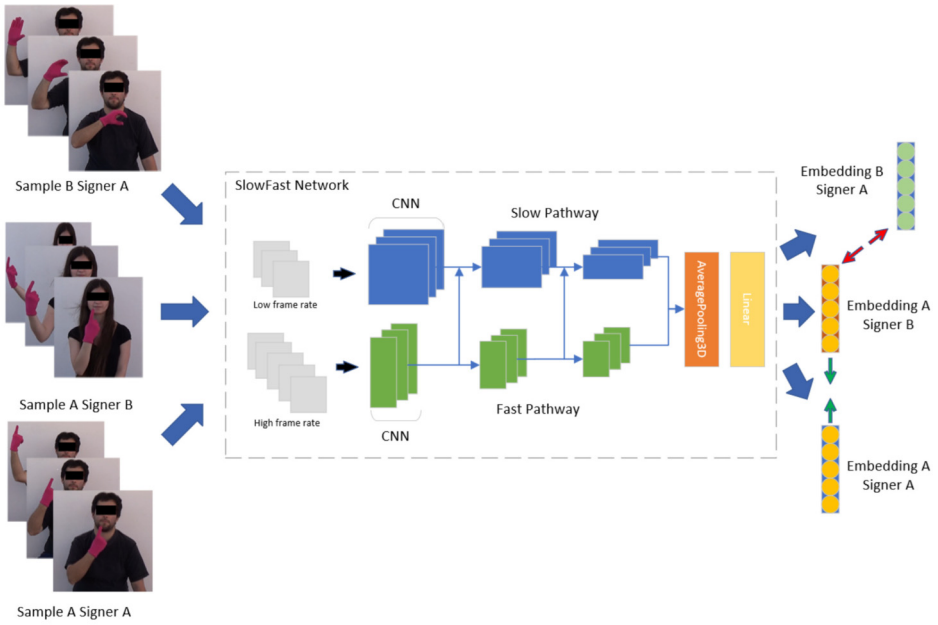

Fig. 3. Illustration of the few-shot sign language recognition pipeline using a prototypical network with the SlowFast feature embedder. The graphic shows the processing of video samples and the training process, where embeddings are learned to cluster same-class samples and separate different-class samples, enabling classification based on proximity to class prototypes. For SlowFast architecture details, see Fan *et al.* (2020), Feichtenhofer *et al.* (2019).

At inference time, unseen classes are evaluated in the same manner. Each test sample is embedded using the trained feature extractor and compared against class prototypes computed from the support set of the target episode. The predicted label corresponds to the nearest prototype in the embedding space.

This unified pipeline enables a consistent evaluation of different backbone architectures under identical few-shot learning conditions, allowing direct comparison of their ability to produce discriminative feature embeddings for sign language recognition.

2.2. *Episodic Few-Shot Learning under Data Scarcity*

Few-shot learning (FSL) is adopted in this work to address the inherent data scarcity in sign language recognition (SLR), where large-scale annotated datasets are difficult to obtain due to collection cost, privacy constraints, and the complexity of sign annotation (Snell *et al.*, 2017). In such settings, traditional deep learning approaches require extensive labelled data and often fail to generalize to unseen sign classes. To overcome this limitation, FSL reformulates the problem as episodic learning, where the model is trained on a distribution of tasks constructed from small support sets and corresponding query sets. Each episode simulates a low-data recognition scenario with novel class combinations, enforcing generalization to unseen categories.

Unlike standard classification models, which learn fixed decision boundaries over predefined classes, FSL learns a feature embedding space in which class similarity can be measured directly. This enables classification of unseen classes by comparing query samples to support set representations, improving adaptability under limited data conditions (Alsulami *et al.*, 2024).

In this work, this property is leveraged to enable robust sign recognition under constrained data availability by combining episodic training with prototype-based classification in a shared embedding space. The episodic configuration (N-way, K-shot, Q-query) and training protocol are defined in Section 3.3.

2.3. *Prototypical Classification*

The proposed framework employs Prototypical Networks (Snell *et al.*, 2017) as a metric-based classification strategy operating in the learned embedding space. In this setting, each class is represented by a prototype computed as the average embedding of its support examples.

Training follows an episodic meta-learning paradigm (Parnami and Lee, 2022), where the model is optimized over a sequence of few-shot tasks constructed from support and query sets. This encourages the model to learn representations that generalize to unseen classes by repeatedly simulating low-data classification scenarios. Optimization is based on a metric-learning objective that increases similarity between query samples and their corresponding class prototypes while reducing similarity to incorrect classes (Parnami and Lee, 2022). This improves the discriminative structure of the embedding space under limited data conditions.

During classification, a query sample is mapped into the same embedding space using the feature extractor described in Section 2.4. It is then assigned to the class whose prototype is closest according to a distance-based similarity measure, as introduced in Snell *et al.* (2017).

In this work, classification is performed directly in the embedding space produced by the backbone network, which acts as the feature extractor for all evaluated architectures. This design ensures a unified representation space across models and enables a consistent comparison under identical few-shot conditions.

2.4. *Feature Extraction Backbones*

Following our previous research (Kalinowski and Kostek, 2026b), current solutions for sign language recognition primarily rely on either RGB data or skeletal representations. While multi-modal approaches incorporating additional streams such as optical flow or depth can improve performance, they significantly increase computational complexity. In this work, we focus exclusively on RGB-based models to maintain a balance between accuracy and efficiency.

To enable a systematic comparison, we select spatiotemporal architectures that have been widely adopted in the literature and are capable of processing entire video sequences.

Specifically, we consider S3D (Xie *et al.*, 2018), I3D (Carreira and Zisserman, 2018), R(2+1)D (Tran *et al.*, 2018), (3+2+1)D ResNet (Zhou *et al.*, 2021b), and SlowFast (Feichtenhofer *et al.*, 2019). These models are integrated as interchangeable feature extractors within the proposed few-shot learning framework and evaluated under identical conditions. Additionally, the SlowFast network is included as a reference backbone due to its established effectiveness in sign language recognition benchmarks. In our previous work (Kalinowski and Kostek, 2026a), we evaluated a SlowFast-based model within a prototypical few-shot learning framework, achieving 94.33% accuracy on held-out classes. This demonstrated the effectiveness of spatiotemporal representations combined with metric learning for isolated sign recognition, motivating the extension of the evaluation to multiple backbone architectures in this study.

To ensure a fair comparison, we introduce a unified embedding representation. Features extracted by each backbone are temporally aggregated using a global pooling operation and passed through a linear projection layer to produce a fixed-dimensional embedding vector. This design enables all models to operate within the same representation space, allowing direct comparison of their effectiveness in few-shot sign language recognition.

All backbone architectures are based on their original formulations; however, due to hardware constraints, architectural adaptations were required for certain models. In particular, modifications were introduced to reduce computational and memory overhead in deeper or more parameter-intensive layers, while preserving the overall structure and spatiotemporal modelling principles of each network. As a result, the exact number of layers and layer-wise parameter configurations may differ from the original implementations. In addition, despite these architectural adjustments, it was not possible to maintain identical batch sizes across all models. The achievable batch and episodic configurations were inherently constrained by the computational complexity of each backbone, which led to model-specific limitations in the number of samples processed per batch. All training-related hyperparameters and episodic configurations are provided in Section 3.5, while all architectural implementations and modifications are documented in the accompanying source code (https://github.com/mkalinowski11/SlowFast_Prototypical_SLR.git[1]).

SlowFast. The SlowFast network (Feichtenhofer *et al.*, 2019) is a two-stream architecture designed for video understanding, consisting of a Slow pathway that captures spatial semantics at a low frame rate and a Fast pathway that models fine-grained temporal dynamics at a higher frame rate. Information from both pathways is fused through lateral connections, enabling joint spatiotemporal representation learning. Due to its strong performance in sign language recognition tasks (Ahn *et al.*, 2023; Hassan *et al.*, 2021), this model is included as a primary backbone and used as a feature extractor within the proposed framework.

S3D. The S3D architecture (Xie *et al.*, 2018) improves the efficiency of 3D convolutional networks by factorizing standard 3D convolutions into separate spatial (2D) and temporal (1D) components. This decomposition reduces computational complexity while maintaining strong representational capacity. Combined with Inception-style modules, S3D has demonstrated competitive performance in action recognition and sign language tasks (Liu

et al., 2024; Chen *et al.*, 2023b, 2023a; Chen *et al.*, 2024). In this work, it is used as a feature extractor within the proposed framework.

I3D. The I3D model (Carreira and Zisserman, 2018) extends 2D convolutional networks to the temporal domain by inflating filters into 3D, enabling joint spatiotemporal feature learning. It also supports multi-stream configurations incorporating RGB and optical flow inputs. Due to its effectiveness and widespread adoption in video understanding and sign language recognition (Cui *et al.*, 2019; Shen *et al.*, 2024), I3D is included as one of the evaluated backbones and used as a feature extractor in our framework.

R(2+1)D. The R(2+1)D architecture (Tran *et al.*, 2018) decomposes 3D convolutions into a 2D spatial convolution followed by a 1D temporal convolution, forming sequential spatial and temporal operations. This design increases the number of non-linearities and improves optimization while preserving the representational power of standard 3D convolutions. Due to its strong performance in action recognition and adoption in sign language research (Han *et al.*, 2022; Jiang *et al.*, 2021; Papadimitriou and Potamianos, 2023), it is incorporated as a feature extractor within the proposed framework.

(3+2+1)D ResNet. The (3+2+1)D ResNet model (Zhou *et al.*, 2021b) combines $(2+1)$D and 3D convolutional blocks within a single architecture. Early layers focus on separate spatial and temporal feature extraction, while deeper layers capture joint spatiotemporal relationships. This hybrid design enables efficient hierarchical feature learning and has also been applied to sign language recognition tasks (Zhou *et al.*, 2022). In this study, it is used as a feature extractor within the proposed framework.

2.5. *Evaluation Protocols*

All experiments are conducted on the test split described in Section 3.2 and follow three complementary evaluation protocols designed to assess few-shot generalization performance and the stability of the learned prototype representations.

Episodic few-shot evaluation. In this setting, each episode follows an N-way, K-shot configuration, where N classes are randomly sampled and K support samples per class are used to compute class prototypes as the mean embedding of the support examples. A fixed number of $Q = 15$ query samples are drawn from the same classes and classified by nearest-prototype matching in the embedding space using the distance metric defined in Section 3.4. To ensure statistically robust estimates, performance is averaged over 1000 independently sampled episodes, and the evaluation is performed for $N \in 5, 10$ and $K \in 1, 5, 10$, covering varying levels of data scarcity. All episodic tasks are sampled from the same fixed test split used in the full test-set evaluation, implying that individual samples may appear in multiple episodes across the 1000-task evaluation.

Full test-set prototype-based evaluation. In addition to episodic evaluation, a deterministic protocol is employed in which all available test samples are processed simultaneously. For each class, a single prototype is computed as the mean embedding of all samples belonging to that class, and each test instance is classified based on its nearest prototype across the full set of classes. This setting evaluates the global discriminative structure of the embedding space without episodic sampling variability.

Split-based evaluation. To further assess robustness, each class is partitioned into five equal subsets. For each split, four subsets (i.e. 80% of the samples) are used to compute class prototypes, while the remaining subset (20%) serves as the evaluation set. This process is repeated five times, with each subset used once for evaluation. Samples in the evaluation set are classified using nearest-prototype assignment in the embedding space. Performance is reported as the mean and standard deviation across all splits.

The SlowFast network (Feichtenhofer *et al.*, 2019) was selected as a reference backbone due to its strong expected performance; therefore, the episodic few-shot evaluation across multiple (N, K) configurations is performed in detail for this model. The remaining architectures are primarily assessed using the full test-set and split-based protocols to enable a consistent and computationally tractable comparison. The exact episodic configurations used for each model may vary and are provided in Section 3.3.

3. Experiments

3.1. *Dataset*

The LSA64 dataset comprises 3,200 videos of Argentinian Sign Language (LSA), collected to support the development of a sign lexicon and the training of an automatic sign recognition system. It features 10 non-expert participants, each producing 5 repetitions of 64 selected signs, covering both verbs and nouns that are frequently used in the Argentinian Sign Language vocabulary. The videos were recorded in front of a white background, with participants dressed in black clothing and wearing fluorescent gloves to facilitate hand segmentation and reduce the effects of skin tone variations, while still maintaining the difficulty associated with hand shape recognition (Ronchetti *et al.*, 2023).

3.2. *Data Processing*

The dataset was divided into training and validation subsets according to an 80:20 ratio, resulting in 52 classes assigned to the training set and 12 classes reserved for validation. The validation samples were strictly excluded from the training phase to prevent the model from adapting its embeddings to unseen data. A detailed description of the dataset partitioning is provided in Table 4. All parameters governing the data split, including the fixed random seed used to ensure reproducibility, are specified in the accompanying code[1].

During preprocessing, all video samples were resized to a spatial resolution of 224 × 224 pixels and normalized to the range [0, 1]. These steps ensured compatibility with

Table 4
Dataset split into train and test sets.

Split	# Samples	Sample IDs
Train	52	1, 2, 4, 5, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 19, 20, 21, 22, 23, 24, 26, 28, 29, 30, 31, 32, 33, 34, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 51, 52, 53, 55, 57, 58, 59, 60, 61, 62
Test	12	3, 6, 7, 18, 25, 27, 35, 50, 54, 56, 63, 64

the downstream model requirements, while retaining all three RGB channels throughout training.

Since the video sequences had different frame counts, shorter sequences were padded with zeros to match the length of the longest sequence in each batch. This zero-padding method ensured uniform input sizes across all training samples.

3.3. *Training Details*

Training was conducted over 64 epochs, with 100 episodes sampled per epoch. Episodes were generated using a custom sampling strategy (see source code[1]) designed to ensure a balanced representation of classes within each episode. The exact episodic configuration depended on the selected model architecture and its computational complexity. For certain architectures, increasing the number of support or query samples significantly enlarged the computational graph and gradient memory footprint, particularly in models with skip connections. As a result, larger episodic configurations exceeded the available GPU memory during backpropagation. Therefore, for memory-intensive models, the (N,K,Q) setting was reduced to ensure stable training (see Table 6).

For the SlowFast (Feichtenhofer *et al.*, 2019) architecture, a 5-way setting was used, meaning that each episode included 5 classes, with 3 support samples per class for prototype computation and 2 query samples per class for distance-based evaluation. In the case of the R(2+1)D (Tran *et al.*, 2018) model, a 5-way configuration was likewise adopted; however, due to its greater computational complexity, only 2 support samples and 1 query sample per class were feasible. To provide a fair and informative comparison, the SlowFast model was additionally evaluated under a reduced 3-way 2-shot 1-query configuration. This enables direct comparison with architectures that were restricted to lower episodic settings due to hardware limitations. The remaining architectures, including S3D (Xie *et al.*, 2018), I3D (Carreira and Zisserman, 2018), and R(3+2+1)D (Zhou *et al.*, 2021b), were trained under a 3-way 2-shot 1-query episodic configuration, corresponding to the highest configuration achievable without exceeding GPU memory limits (see Table 6).

Within each episode, both the classes and their corresponding support and query samples were selected at random. To guarantee reproducibility, the seed value was fixed at 123 for both the PyTorch and Python random libraries, and applied consistently throughout training, validation, and testing. The training objective relied on the Euclidean distance metric, following the algorithm and loss formulation described in Snell *et al.* (2017), with necessary adjustments to accommodate video inputs.

3.4. *Metrics*

The evaluation of models trained under the few-shot learning paradigm requires adequate metrics to compare models and assess their ability to embed sign language samples. We can measure how well the classification of unseen samples matches their respective class by checking if they are closer to their class centroid. We should also evaluate the quality of the centroids, ensuring they are well separated from each other to keep samples of

different signs apart, while embeddings of samples within the same class remain grouped. Although accuracy is an important factor, considering only the model's overall accuracy may not be sufficient for a complete assessment. Even if samples are close to their centroid, centroids might be too close to one another in the embedding space, leading to potential misclassification of future samples due to variations in setting, environment, signer, or angle. This is why, beyond the accuracy metric, we use the following additional metrics: Prototype–Centroid Distance Ratio, Intra / Inter-Class Distance Ratio, and Prototype Rank Consistency.

To thoroughly validate the model's performance, embeddings were generated for each sample within the test dataset. These embeddings were subsequently organized by their corresponding classes, and a prototype, representing the centroid of all embeddings within each class, was computed for each group. Prototypes were computed as the mean embedding vector for each class based on training samples. For each test embedding, the Euclidean distance to each class prototype was then calculated, and the sample was assigned to the class associated with the nearest prototype.

Prototypical Calculation

The prototype for each class k, denoted as $\mathbf{c}_k$, is computed as the mean of the embedding vectors $f_\phi(\mathbf{x}_i)$ for all samples $(\mathbf{x}_i, y_i)$ belonging to the support set S_k represented by samples of the same class:

$$\mathbf{c}_k = \frac{1}{|S_k|} \sum_{(\mathbf{x}_i, y_i) \in S_k} f_\phi(\mathbf{x}_i). \tag{1}$$

For a test embedding $f_\phi(\mathbf{x})$, the classification is performed by finding the nearest prototype based on the Euclidean distance function d:

$$\hat{y} = \arg\min_k \; d\big(f_\phi(\mathbf{x}), \mathbf{c}_k\big). \tag{2}$$

Accuracy Calculation

The overall accuracy (Acc) for the evaluation task is then computed using the indicator function $\mathbb{I}[\cdot]$ over all samples in the test dataset Q:

$$Acc = \frac{1}{|Q|} \sum_{(\mathbf{x}, y) \in Q} \mathbb{I}[y = \hat{y}], \tag{3}$$

where $|Q|$ represents the total number of test samples.

Prototype-Centroid Distance Ratio (PCDR)

To evaluate the discriminative quality of the learned embedding space, we employ the Prototype-Centroid Distance Ratio (PCDR). This metric measures how well a sample is clustered around its corresponding class prototype relative to its separation from competing class prototypes.

For a test sample $\mathbf{x}$ belonging to class y, the PCDR is defined as the ratio between the squared Euclidean distance to its assigned class prototype $\mathbf{c}_y$ and the minimum squared Euclidean distance to any other class prototype $\mathbf{c}_k$, $k \neq y$:

$$PCDR(\mathbf{x}) = \frac{\| f_\phi(\mathbf{x}) - \mathbf{c}_y \|^2}{\min_{k \neq y} \| f_\phi(\mathbf{x}) - \mathbf{c}_k \|^2}. \tag{4}$$

A lower average PCDR across the test set indicates that samples are more tightly clustered around their correct prototypes while remaining well separated from prototypes of other classes, reflecting higher feature discriminability.

Intra-/Inter-Class Distance Ratio (ICDR)

To further assess the structure of the learned embedding space, we employ the Intra-/Inter-Class Distance Ratio (ICDR), which evaluates the compactness of samples within the same class relative to the separation between different class prototypes.

The intra-class distance for a class k is defined as the average squared Euclidean distance between the embeddings of test samples belonging to that class and the corresponding class prototype $\mathbf{c}_k$:

$$D_{\text{intra}}(k) = \frac{1}{|Q_k|} \sum_{\mathbf{x} \in Q_k} \| f_\phi(\mathbf{x}) - \mathbf{c}_k \|^2, \tag{5}$$

where Q_k denotes the set of test samples belonging to class k.

The inter-class distance is defined as the average squared Euclidean distance between the prototype $\mathbf{c}_k$ and all other class prototypes:

$$D_{\text{inter}}(k) = \frac{1}{K-1} \sum_{j \neq k} \| \mathbf{c}_k - \mathbf{c}_j \|^2. \tag{6}$$

The ICDR for class k is then computed as the ratio between the intra-class and inter-class distances:

$$ICDR(k) = \frac{D_{\text{intra}}(k)}{D_{\text{inter}}(k)}. \tag{7}$$

Finally, the overall ICDR score is obtained by averaging over all classes:

$$ICDR = \frac{1}{K} \sum_{k=1}^{K} ICDR(k). \tag{8}$$

A lower ICDR value indicates tighter clustering of samples within each class and greater separation between different class prototypes, reflecting a more discriminative embedding space.

Prototype Rank Consistency (PRC)

To assess the stability of class relationships in the learned embedding space, we introduce the Prototype Rank Consistency (PRC) metric. This metric evaluates whether the relative ranking of class prototypes with respect to a given sample is consistent with the ground-truth class assignment.

For a test sample $\mathbf{x}$, the distances to all class prototypes $\mathbf{c}_k$ are computed and sorted in ascending order, producing an ordered list of class indices based on proximity in the embedding space. Let $r(\mathbf{x})$ denote the rank position of the correct class prototype $\mathbf{c}_y$ within this ordered list, where $r(\mathbf{x}) = 1$ indicates that the correct prototype is the nearest.

The PRC score for a test sample is defined as:

$$PRC(\mathbf{x}) = 1 - \frac{r(\mathbf{x}) - 1}{C - 1}, \tag{9}$$

where C denotes the total number of classes. This formulation yields a normalized score in the range [0, 1], where a value of 1 indicates that the correct prototype is ranked first (i.e. it is the closest prototype to the sample).

The overall PRC metric is then computed as the average PRC score over all test samples in the query set Q:

$$PRC = \frac{1}{|Q|} \sum_{\mathbf{x} \in Q} PRC(\mathbf{x}). \tag{10}$$

A higher PRC value indicates greater consistency in prototype ranking, reflecting a more stable and reliable embedding space for prototype-based classification.

3.5. *Experimental Setup and Results*

All experiments were conducted on the test split described in Section 3.2. As motivated in the Introduction, the SlowFast network (Feichtenhofer *et al.*, 2019) was assumed to provide the best performance on the considered datasets; therefore, the initial detailed evaluation was performed using this model. Following the episodic evaluation protocol defined in Section 2.5, experiments were conducted using 1000 episodes per configuration. Each episode used a fixed number of $Q = 15$ query samples per class, independently sampled for each episode. Performance was measured as the average episode-level accuracy over 1000 episodes. The evaluation was performed for $N \in 5, 10$ and $K \in 1, 5, 10$, and the resulting average accuracies are summarized in Table 5. It should be emphasized that the accuracies reported in Table 5 (maximum 92.0%, obtained for the 5-way 10-shot configuration) correspond to episodic few-shot evaluation, where performance is averaged over randomly sampled tasks with limited support and query sets. These results are therefore not directly comparable to the higher accuracy reported later (94.33%), which is obtained under a different evaluation protocol using the full test set and complete class prototypes. The episodic evaluation reflects performance under stochastic, low-data classification scenarios, where each task is constructed from a limited number of support and

Table 5
Average classification accuracy for different few-shot configurations obtained using the SlowFast model ($q_{\text{query}} = 15$, $n_{\text{episodes}} = 1000$; values in [%].

n_way	Accuracy (k_support)		
	1-shot	5-shot	10-shot
5-way	80.0	90.7	92.0
10-way	66.0	81.9	84.9

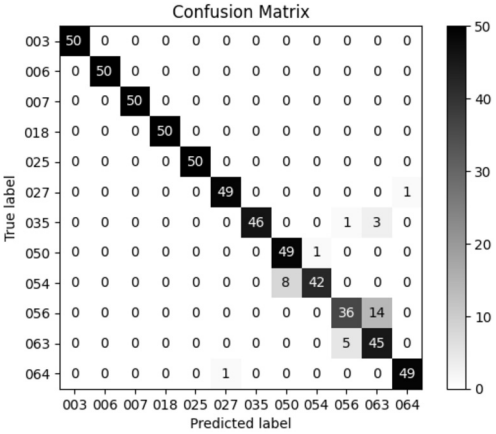

Fig. 4. Confusion matrix of test set predictions for the SlowFast model under the 5-way, 3-shot, 2-query (5-3-2) configuration. The matrix summarizes classification results over 600 test samples, achieving an overall accuracy of 94.33%, with 34 misclassified instances. Diagonal elements indicate correct predictions, while off-diagonal entries highlight class-level confusions.

query samples. In contrast, the full test-set evaluation represents a deterministic setting in which all available samples contribute to prototype estimation, resulting in a more stable and typically higher accuracy.

In the second experiment, a full test-set prototype-based evaluation was performed. In contrast to the episodic setting, where only $Q = 15$ query samples are evaluated per randomly sampled episode, the consecutive experiment utilizes the entire test set of 600 samples simultaneously in a single deterministic evaluation. Following the full test-set evaluation protocol (Section 2.5), a single prototype per class was computed using all 50 samples, and all 600 test instances were classified in a single deterministic pass. Under this setting, 34 samples were misclassified, resulting in an overall accuracy of 94.33%. The detailed classification outcomes are visualized in the confusion matrix shown in Fig. 4, which highlights both correct predictions (diagonal entries) and inter-class confusions (off-diagonal entries).

The same evaluation protocol was applied to additional spatiotemporal architectures, including S3D (Xie *et al.*, 2018), I3D (Carreira and Zisserman, 2018), R(2+1)D (Tran *et al.*, 2018), and R(3+2+1)D (Zhou *et al.*, 2021b). For each model, class prototypes were computed using all available samples per class, and classification was performed by

Table 6
Comparison of few-shot models across accuracy and prototype-based metrics. The column (N,K,Q) specifies the N-way classification setting with K-support samples per class and Q-query samples.

Model	(N, K, Q)	Accuracy (% ↑)	PCDR (↓)	ICDR (↓)	PRC (↑)
SlowFast (Feichtenhofer *et al.*, 2019)	5-3-2	**94.33**	**0.4448**	0.1433	**0.9938**
SlowFast (Feichtenhofer *et al.*, 2019)	3-2-1	87.33	0.5881	0.1424	0.9853
S3D (Xie *et al.*, 2018)	3-2-1	82.00	0.6051	0.1311	0.9798
I3D (Carreira and Zisserman, 2018)	3-2-1	82.83	0.6549	**0.0841**	0.9830
R(2+1)D (Tran *et al.*, 2018)	5-2-1	90.33	0.5195	0.1503	0.9897
R(3+2+1)D (Zhou *et al.*, 2021b)	3-2-1	82.67	0.7335	0.1281	0.9809

nearest-prototype matching across all 600 test instances. The resulting accuracies, along with prototype-based metrics such as per-class distance rate (PCDR), inter-class distance rate (ICDR), and prototype robustness coefficient (PRC), are reported in Table 6, providing a consistent comparison across architectures under the same evaluation protocol.

Finally, an additional evaluation was conducted to assess the stability of the prototype-based representation using a split-based protocol. Following the split-based evaluation protocol (Section 2.5), performance was computed as the mean and standard deviation across all five splits, as summarized in Table 7. This evaluation provides additional insight into the robustness and generalization of the learned embeddings under varying data partitions.

For completeness, Table 8 presents a comparison of the evaluated architectures in terms of the number of trainable parameters used in this study, reflecting the computational complexity of the adapted backbone models within the proposed few-shot learning framework.

3.6. *UMAP Visualization*

The following Uniform Manifold Approximation and Projection (UMAP) visualization was performed specifically for the SlowFast network (Feichtenhofer *et al.*, 2019) to illustrate the clustering behaviour and generalization of its learned features. To visualize the embeddings, we employed the UMAP algorithm (McInnes *et al.*, 2020), a dimension reduction technique that constructs a fuzzy topological model of high-dimensional data using local manifold approximations and fuzzy simplicial sets. It assumes data points are uniformly distributed on a locally connected Riemannian manifold with a constant metric, optimizing a low-dimensional representation by minimizing cross-entropy between high- and low-dimensional models. Unlike t-distributed Stochastic Neighbour Embedding (t-SNE), UMAP preserves both local neighbourhoods and global structures efficiently (McInnes *et al.*, 2020), making it ideal for few-shot learning where sign language samples of the same class must cluster closely. We applied UMAP to test embeddings to assess generalization and to compare embeddings from 12 randomly selected training classes, with results shown in Fig. 5.

Table 7
Five–split evaluation of prototype-based metrics. For each model, one split is used as the query set while the remaining splits are used to compute class prototypes. The final row for each model reports the mean and standard deviation across splits.

Model (N, K, Q)	Split	Accuracy (↑)	PCDR (↓)	ICDR (↓)	PRC (↑)
SlowFast (5-3-2)	1	0.9250	0.4822	0.1634	0.9932
	2	0.9750	0.4392	0.1604	0.9977
	3	0.9583	0.4689	0.1677	0.9947
	4	0.9083	0.4730	0.1587	0.9902
	5	0.9500	0.4680	0.1611	0.9955
	Mean ± Std	**0.9433 ± 0.0237**	**0.4663 ± 0.0148**	0.1623 ± 0.0032	**0.9943 ± 0.0026**
SlowFast (3-2-1)	1	0.8667	0.5759	0.1454	0.9856
	2	0.8333	0.6689	0.1507	0.9833
	3	0.8750	0.5685	0.1448	0.9856
	4	0.8250	0.6278	0.1531	0.9788
	5	0.9000	0.5782	0.1365	0.9894
	Mean ± Std	0.8600 ± 0.0296	0.6039 ± 0.0408	0.1461 ± 0.0060	0.9845 ± 0.0035
S3D (3-2-1)	1	0.8250	0.6633	0.1350	0.9833
	2	0.7917	0.6159	0.1383	0.9727
	3	0.8500	0.5738	0.1286	0.9856
	4	0.8250	0.5944	0.1360	0.9803
	5	0.7750	0.6652	0.1330	0.9773
	Mean ± Std	0.8133 ± 0.0265	0.6225 ± 0.0359	0.1342 ± 0.0037	0.9798 ± 0.0050
I3D (3-2-1)	1	0.8083	0.7270	0.0864	0.9803
	2	0.8250	0.6624	0.0920	0.9818
	3	0.8333	0.7700	0.0886	0.9848
	4	0.8250	0.5898	0.0825	0.9841
	5	0.8083	0.6354	0.0820	0.9811
	Mean ± Std	0.82 ± 0.01	0.6769 ± 0.0643	**0.0863 ± 0.0038**	0.9824 ± 0.0017
R(2+1)D (5-2-1)	1	0.8917	0.5194	0.1508	0.9894
	2	0.9000	0.5188	0.1581	0.9902
	3	0.9250	0.5213	0.1520	0.9924
	4	0.8167	0.6271	0.1623	0.9818
	5	0.9167	0.4812	0.1469	0.9909
	Mean ± Std	0.8900 ± 0.0389	0.5336 ± 0.0535	0.1540 ± 0.0057	0.9889 ± 0.0041
R(3+2+1)D (3-2-1)	1	0.8500	0.6963	0.1343	0.9848
	2	0.8500	0.6509	0.1456	0.9833
	3	0.8833	0.6335	0.1424	0.9879
	4	0.8000	0.6764	0.1468	0.9795
	5	0.8083	0.6813	0.1465	0.9803
	Mean ± Std	0.8383 ± 0.0305	0.6677 ± 0.0225	0.1431 ± 0.0047	0.9832 ± 0.0031

3.7. *Discussion and Study Limitation*

The proposed SlowFast meta-based network achieved an accuracy of 94.33% on the test split dataset, effectively handling unseen data. Predictions were made by calculating the distance between a sample's embedding and class prototypes, formed by averaging embeddings within each class. This accuracy of 94.33% (error rate 5.67%) demonstrates that the network effectively fulfills prototypical learning objectives, underscoring its reliability as a feature extractor for isolated SLR.

Table 8
Comparison of evaluated spatiotemporal architectures in terms of the number of trainable parameters used in this study. The values reflect the parameter counts of the adapted backbone models employed within the prototypical few-shot learning framework.

Model	Trainable parameters
SlowFast (Feichtenhofer *et al.*, 2019)	**411 506**
I3D (Carreira and Zisserman, 2018)	12 418 464
S3D (Xie *et al.*, 2018)	12 418 464
R(2+1)D (Tran *et al.*, 2018)	4 684 167
R(3+2+1)D (Zhou *et al.*, 2021b)	3 793 300

Class separability depends strongly on intra-class consistency. Signs that differ in starting position, motion speed, or hand orientation produce more dispersed embeddings and thus reduce prototype compactness; this variation is intrinsic to LSA64, where different signers sometimes modify the same sign's execution. For instance, signs "027", "056", "063" and "064" show variation in handshape, signing height, finishing position, and temporal execution speed, occasionally accompanied by non-manual cues (e.g. facial expressions) that may mislead the model. UMAP visualization (McInnes *et al.*, 2020) of test embeddings confirms this: several classes form distinct clusters, yet visually similar gestures such as "050", "054", "056" and "063" overlap (Fig. 5). These overlaps are echoed in the confusion matrix (e.g. 14 of 50 samples from "056" misclassified as "063", Fig. 4), indicating that visual/temporal inconsistencies (including subtle non-manual cues) are primary confusion sources. A similar behaviour is observed for class "054", where 8 out of 50 samples are misclassified as "050". In addition, a single sample from class "064" is incorrectly predicted as "027". These misclassifications are reflected in the UMAP projection of the test subset (Fig. 5), where the clusters corresponding to classes "054" and "050", as well as "056" and "063", are located in close proximity and partially overlap. A comparable overlap is visible between classes "027" and "064"; however, in this case only one misclassification occurs. This indicates that, for some class pairs, embeddings remain close to their class prototype, while for others (e.g. "054" vs. "050"), increased intra-class spread causes some samples to lie nearer to neighbouring clusters, resulting in higher confusion. While the 2-D UMAP projection validates qualitative cluster structure, it can distort some 128-dimensional relationships; stronger high-dimensional evaluation metrics would complement these visualizations. Finally, the few-shot episodic sampling, where only a subset of classes appears per episode, can limit prototype refinement because not all samples contribute to prototype updates. Beyond the detailed analysis of the SlowFast model, the comparative evaluation presented in Table 6 offers additional insight into the behaviour of different spatiotemporal architectures under the prototypical few-shot learning framework. Among all evaluated models, SlowFast architecture achieves the highest classification accuracy (94.33%) together with the most favorable prototype-based metrics, including the lowest Prototype–Class Distance Ratio (PCDR) and the highest Prototype Reliability Coefficient (PRC). These results suggest that the embeddings produced by the SlowFast model form both compact intra-class clusters and well-separated inter-class boundaries, which is especially beneficial for prototype-based classification.

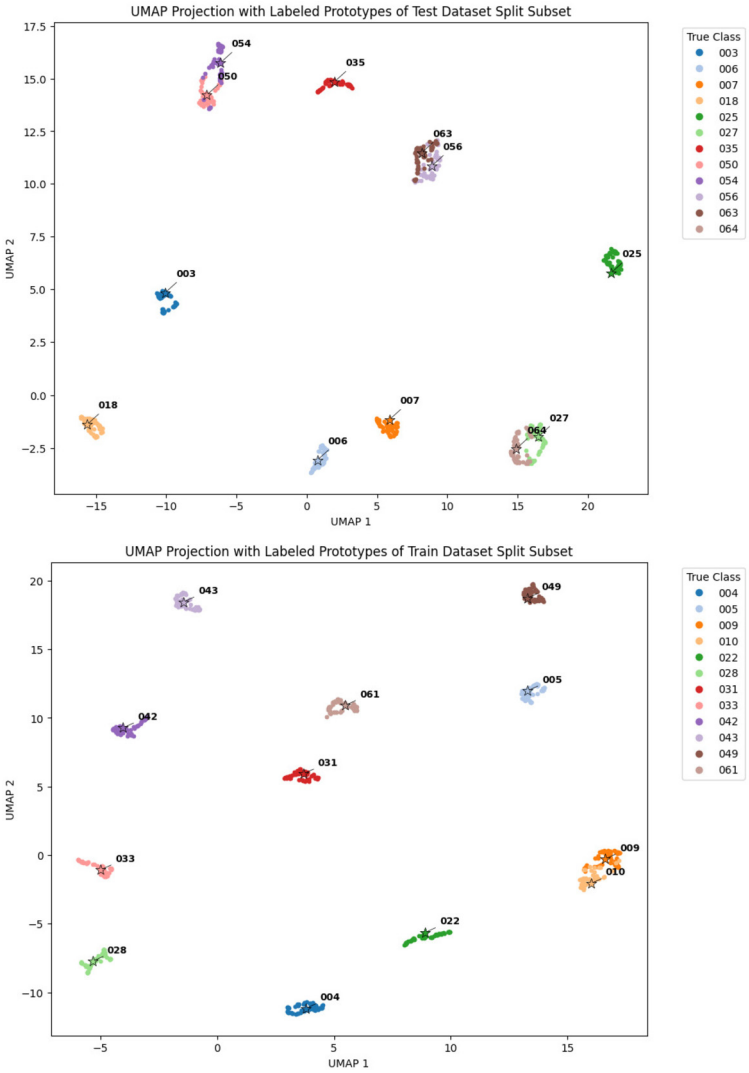

Fig. 5. UMAP visualization of 128-dimensional embeddings extracted from the SlowFast network. The plot above shows test set embeddings, while the plot below presents embeddings from 12 randomly selected training classes. The x-axis (UMAP 1) and y-axis (UMAP 2) correspond to the two components of the 128→2 UMAP projection. Class prototypes are indicated by star-shaped markers. The visualization illustrates the clustering behaviour and generalization capability of the SlowFast model in the learned embedding space.

However, an exception is observed in the Intra-Class Distance Ratio (ICDR), where the I3D model achieves the lowest value. This suggests that, although I3D produces relatively compact clusters within individual classes, the separation between different class prototypes is less pronounced compared to SlowFast architecture. Consequently, the overall classification accuracy remains lower despite favourable intra-class compactness. This shows that both intra-class consistency and inter-class separation are important when assessing embedding quality in prototype-based learning.

The comparison also reveals the influence of the episodic configuration (N, K, Q) on model performance. In few-shot learning, the number of support samples per class plays a crucial role in prototype estimation, as prototypes are constructed by averaging the embeddings of the support examples. Experiments conducted with larger support sets generally lead to more stable prototype representations and improved classification performance. This trend can be observed when comparing the SlowFast model configuration (5–3–2) with settings involving fewer samples per class, where a reduction in accuracy is evident.

Despite operating under more constrained episodic settings in some experiments, the SlowFast model behaves consistently as well as or better than architectures such as S3D, I3D, and R(3+2+1)D. This indicates that the SlowFast architecture is particularly effective at encoding discriminative spatiotemporal representations that stay reliable even when prototype estimation relies on limited data. One possible explanation for this behaviour is the design of SlowFast networks, which explicitly model temporal information at two different frame rates. The slow pathway captures semantic and structural motion patterns over longer temporal contexts, while the fast pathway focuses on rapidly changing motion cues. Such a design is particularly well suited to sign language recognition, where meaningful information is conveyed through both broader arm movements and subtle hand or finger details. The combination of these complementary temporal representations can help the SlowFast backbone generate embeddings that better reflect the spatiotemporal features of signing gestures.

Another important aspect highlighted by this study is the parameter efficiency of the evaluated architectures. As summarized in Table 8, the SlowFast model used in this research contains the smallest number of trainable parameters among the evaluated networks. Despite its lower parameter count, it consistently achieves the best performance across most evaluation metrics. This indicates that the architecture is not only effective in capturing relevant motion features but also computationally efficient, allowing more samples to be processed within the available GPU memory constraints. Consequently, the SlowFast model represents a favourable balance between representational capacity and computational cost in the context of few-shot sign language recognition.

To address these overlaps, a margin-based prototypical loss could be adopted to explicitly enlarge inter-class distances and improve discrimination among visually similar signs. We used prototypical loss for its robustness, i.e. class anchors are computed as class means. However, experimenting with margin terms and alternatives (e.g. median anchors or hybrid anchor estimators) may increase prototype stability and reduce confusion from outliers.

It is important to note that all evaluated architectures were adjusted to match the available computational resources. In particular, GPU memory limits required changes to input resolution, batch sizes, and episodic setups during training. As a result, some models might not have operated under their most optimized conditions. Although each network was trained for 64 epochs with 100 episodes per epoch, giving repeated exposure to different support–query combinations, the smaller episode sizes for certain models may have limited their ability to fully utilize their representational capacity.

However, several limitations must be recognized. First, the scope of the evaluation is constrained by the dataset setup. Although the LSA64 dataset includes 64 gesture classes,

only 12 were used in the experiments (Table 4), which may limit the ability to assess its generalizability to a broader range of vocabulary. Additionally, while the model performs well for most categories, it still struggles with overlapping or visually similar signs, indicating limited class separation in certain areas of the embedding space. Conducting broader experiments with multiple random seeds would better quantify variability, although they require more computational resources than currently available. The use of fluorescent gloves during recording (Ronchetti *et al.*, 2023) facilitated hand segmentation and improved recognition, but limits real-world applicability since users are unlikely to wear such aids. Relying only on the RGB input may also cause the model to pick up background or contextual cues that do not generalize.

Overall, the model demonstrates strong performance in few-shot sign language recognition, with effective feature extraction, although challenges persist in separating certain classes and ensuring robustness across broader datasets and more unconstrained visual conditions.

4. Conclusion and Future Research

This study showed that the SlowFast network combined with prototypical learning achieves 94.33% accuracy in isolated sign language recognition (SLR), demonstrating a strong ability to extract distinctive features and generalize to unseen classes within a few-shot learning framework. The model effectively clusters most gesture categories in the embedding space, confirming the suitability of metric-based learning for low-resource SLR scenarios.

Future research should therefore aim to validate the approach with larger, more diverse datasets, especially glove-free and naturalistic RGB recordings, to better reflect real-world conditions. Expanding the evaluation to include all gesture classes would also provide a more comprehensive analysis of scalability. Additionally, incorporating complementary modalities like skeletal landmarks may improve robustness and decrease reliance on artificial markers. Using structural motion data with graph-convolutional SlowFast architecture variants could help the model better leverage spatiotemporal relationships in sign language.

From an optimization perspective, exploring margin-based loss functions (e.g. multi-way contrastive loss) and systematically tuning hyperparameters may further enhance class separability and reduce sampling biases. These improvements could boost the discriminative power of the embedding space, especially for overlapping gesture classes.

Overall, the results support the potential of combining SlowFast architectures with prototypical learning for few-shot SLR, while highlighting several research directions necessary to develop robust, scalable, and marker-independent sign language recognition systems.

References

Ahn, J., Jang, Y., Chung, J.S. (2023). SlowFast Network for Continuous Sign Language Recognition. https://arxiv.org/abs/2309.12304.

Al Abdullah, B.A., Amoudi, G.A., Alghamdi, H.S. (2024). Advancements in sign language recognition: a comprehensive review and future prospects. *IEEE Access*, 12, 128871–128895. https://doi.org/10.1109/ACCESS.2024.3457692.

Alsulami, A., Bajbaa, K., Laradji, I., Luqman, H. (2024). Few-shot learning for sign language recognition with embedding propagation. *Nafath*, 9(27).

Bilge, Y.C., Cinbis, R.G., Ikizler-Cinbis, N. (2023). Towards zero-shot sign language recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(1), 1217–1232. https://doi.org/10.1109/tpami.2022.3143074.

Boháček, M., Hrúz, M. (2023). Learning from what is already out there: few-shot sign language recognition with online dictionaries. In: *2023 IEEE 17th International Conference on Automatic Face and Gesture Recognition*, pp. 1–6. https://api.semanticscholar.org/CorpusID:255570058.

Camgoz, N.C., Hadfield, S., Koller, O., Ney, H., Bowden, R. (2018). Neural sign language translation. In: *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7784–7793. https://doi.org/10.1109/CVPR.2018.00812.

Camgoz, N.C., Koller, O., Hadfield, S., Bowden, R. (2020). Multi-channel Transformers for Multi-articulatory Sign Language Translation. https://arxiv.org/abs/2009.00299.

Carreira, J., Zisserman, A. (2018). Quo Vadis, Action Recognition? A New Model and the Kinetics Dataset. https://arxiv.org/abs/1705.07750.

Chen, H., Wang, J., Guo, Z., Li, J., Zhou, D., Wu, B., Guan, C., Chen, G., Heng, P.-A. (2024). SignVTCL: Multi-Modal Continuous Sign Language Recognition Enhanced by Visual-Textual Contrastive Learning. https://arxiv.org/abs/2401.11847.

Chen, Y., Wei, F., Sun, X., Wu, Z., Lin, S. (2023a). A Simple Multi-Modality Transfer Learning Baseline for Sign Language Translation. https://arxiv.org/abs/2203.04287.

Chen, Y., Zuo, R., Wei, F., Wu, Y., Liu, S., Mak, B. (2023b). Two-Stream Network for Sign Language Recognition and Translation. https://arxiv.org/abs/2211.01367.

Cui, R., Liu, H., Zhang, C. (2019). A deep neural framework for continuous sign language recognition by iterative training. *IEEE Transactions on Multimedia*, 21(7), 1880–1891. https://doi.org/10.1109/TMM.2018.2889563.

de Amorim, C.C., Macêdo, D., Zanchettin, C. (2019). *Spatial-Temporal Graph Convolutional Networks for Sign Language Recognition*. Springer International Publishing, pp. 646–657. https://doi.org/10.1007/978-3-030-30493-5_59.

DeepSign AI (2026). DeepSign AI. https://deepsignai.com/. Accessed: February 2026.

Desai, A., Meulder, M.D., Hochgesang, J.A., Kocab, A., Lu, A.X. (2024). Systemic Biases in Sign Language AI Research: A Deaf-Led Call to Reevaluate Research Agendas. https://arxiv.org/abs/2403.02563.

Emmorey, K. (2023). Ten things you should know about sign languages. *Current Directions in Psychological Science*, 32(5), 387-394. https://doi.org/10.1177/09637214231173071.

Fan, H., Li, Y., Xiong, B., Lo, W.-Y., Feichtenhofer, C. (2020). PySlowFast. https://github.com/facebookresearch/slowfast.

Feichtenhofer, C., Fan, H., Malik, J., He, K. (2019). SlowFast Networks for Video Recognition. https://arxiv.org/abs/1812.03982.

Ferreira, S., Costa, E., Dahia, M., Rocha, J. (2022). A Transformer-Based Contrastive Learning Approach for Few-Shot Sign Language Recognition. https://arxiv.org/abs/2204.02803.

Han, X., Lu, F., Yin, J., Tian, G., Liu, J. (2022). Sign Language Recognition Based on R(2+1)D With Spatial–Temporal–Channel Attention. *IEEE Transactions on Human-Machine Systems*, 52(4), 687–698. https://doi.org/10.1109/THMS.2022.3144000.

Hand Talk (2026). Artificial intelligence for sign language translation. https://www.handtalk.me/en/. Accessed: February 2026.

Hanke, T. (2004). HamNoSys – Representing Sign Language Data in Language Resources and Language Processing Contexts. https://api.semanticscholar.org/CorpusID:15434469.

Hassan, A., Elgabry, A., Hemayed, E. (2021). Enhanced dynamic sign language recognition using SlowFast networks. In: *2021 17th International Computer Engineering Conference (ICENCO)*, pp. 124–128. https://doi.org/10.1109/ICENCO49852.2021.9698904.

Jiang, S., Sun, B., Wang, L., Bai, Y., Li, K., Fu, Y. (2021). Skeleton Aware Multi-Modal Sign Language Recognition. https://arxiv.org/abs/2103.08833.

Kalinowski, M., Kostek, B. (2026a). Few-shot isolated sign language recognition with spatiotemporal SlowFast prototypes. In: *Proceedings of the International Conference on Artificial Intelligence and Soft Computing (ICAISC)*, Zakopane, Poland.

Kalinowski, M., Kostek, B. (2026b). Machine learning in sign language: a comprehensive analysis and trend survey. *Computer Science Review*, 60, 100895. https://api.semanticscholar.org/CorpusID:284849780.

Kennaway, R. (2001). Synthetic animation of deaf signing gestures. In: *Gesture Workshop*. https://api.semanticscholar.org/CorpusID:8191959.

Koller, O., Forster, J., Ney, H. (2015). Continuous sign language recognition: towards large vocabulary statistical recognition systems handling multiple signers. *Computer Vision and Image Understanding*, 141, 108–125. https://doi.org/10.1016/j.cviu.2015.09.013.

Kong, A.P.H. (2016). Multi-linear Transcription and Analysis of Oral Discourse. In: *Analysis of Neurogenic Disordered Discourse Production: From Theory to Practice*. Routledge. 978-1-315-63937-6. https://doi.org/10.4324/9781315639376.

Lingvano (2026). Lingvano. https://app.lingvano.com/. Accessed: February 22, 2026.

Liu, Z., Pang, L., Qi, X. (2024). MEN: mutual enhancement networks for sign language recognition and education. *IEEE Transactions on Neural Networks and Learning Systems*, 35(1), 311–325. https://doi.org/10.1109/TNNLS.2022.3174031.

Lu, H., Salah, A.A., Poppe, R. (2024). TCNet: Continuous Sign Language Recognition from Trajectories and Correlated Regions. https://arxiv.org/abs/2403.11818.

Łacheta, J., Czajkowska-Kisil, M., Linde-Usiekniewicz, J., Rutkowski, P. (Eds.) (2016). *Korpusowy Słownik Polskiego Języka Migowego [Corpus Dictionary of Polish Sign Language]*. Wydział Polonistyki Uniwersytetu Warszawskiego [Faculty of Polish Studies, University of Warsaw], Warszawa, Poland. 978-83-64111-49-5. https://www.slownikpjm.uw.edu.pl/.

McInnes, L., Healy, J., Melville, J. (2020). UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction. https://arxiv.org/abs/1802.03426.

Migam (2025). Omnichannel communication for the Deaf. https://migam.org/. Accessed: 2026-02-22.

Min, Y., Hao, A., Chai, X., Chen, X. (2021). Visual Alignment Constraint for Continuous Sign Language Recognition. https://arxiv.org/abs/2104.02330.

Moryossef, A., Tsochantaridis, I., Dinn, J., Camgöz, N.C., Bowden, R., Jiang, T., Rios, A., Müller, M., Ebling, S. (2021). Evaluating the immediate applicability of pose estimation for sign language recognition. In: *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pp. 3429–3435. https://doi.org/10.1109/CVPRW53098.2021.00382.

Papadimitriou, K., Potamianos, G. (2023). Sign language recognition via deformable 3D convolutions and modulated graph convolutional networks. In: *ICASSP 2023 – 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1–5. https://doi.org/10.1109/ICASSP49357.2023.10096714.

Papastratis, I., Dimitropoulos, K., Konstantinidis, D., Daras, P. (2020). Continuous sign language recognition through cross-modal alignment of video and text embeddings in a joint-latent space. *IEEE Access*, 8, 91170–91180. https://doi.org/10.1109/ACCESS.2020.2993650.

Parnami, A., Lee, M. (2022). Learning from Few Examples: A Summary of Approaches to Few-Shot Learning. https://arxiv.org/abs/2203.04291.

Rastgoo, R., Kiani, K., Escalera, S. (2021). Sign language recognition: a deep survey. *Expert Systems with Applications*, 164, 113794. https://doi.org/10.1016/j.eswa.2020.113794. https://www.sciencedirect.com/science/article/pii/S095741742030614X.

Ronchetti, F., Quiroga, F.M., Estrebou, C., Lanzarini, L., Rosete, A. (2023). LSA64: An Argentinian Sign Language Dataset. https://arxiv.org/abs/2310.17429.

Sari, I.P., Mumtas, F., Fauzan Putra, Z.E.F., Sari, R.D., Zaidiah, A., Snatoni, M.M. (2023). Enhanced few-shot learning for Indonesian sign language with prototypical networks approach. In: *2023 International Conference on Informatics, Multimedia, Cyber and Informations System (ICIMCIS)*, pp. 278–283. https://doi.org/10.1109/ICIMCIS60089.2023.10349031.

Shen, X., Zheng, Z., Yang, Y. (2024). StepNet: spatial-temporal part-aware network for isolated sign language Recognition. *ACM Transactions on Multimedia Computing, Communications, and Applications*, 20(7), 1–19. https://doi.org/10.1145/3656046.

SignAll (2026). A communication bridge between d/Deaf and hearing. https://futureofinterface.org/signall/. Accessed: February 2026.

Signapse AI (2024). Generative AI for sign language announcements. https://www.signapse.ai/. Accessed: 2026-02-22.

SLAIT School (2024). SLAIT School: Real-time ASL Learning Platform. Accessed: February 22, 2026. https://slait.school/.

Snell, J., Swersky, K., Zemel, R.S. (2017). Prototypical Networks for Few-shot Learning. https://arxiv.org/abs/1703.05175.

Sutton, V. (1995). *Lessons in SignWriting: Textbook and Workbook*. The Deaf Action Committee for SignWriting and the Center for Sutton Movement Writing, Inc., La Jolla, CA.

Sutton, V., Slevinski, S., Duell, T. (2004). The SignBank Web Site: SignWriting Software. https://www.signbank.org/. Accessed: 2026-02-22. Includes SignPuddle Online (est. 2004) and SignBank FileMaker Pro databases.

Tang, G.W.L., Brentari, D., González, C., Sze, F.Y.B. (2010). Crosslinguistic variation in prosodic cues. In: *Sign Languages*. https://api.semanticscholar.org/CorpusID:61004529.

Tran, D., Wang, H., Torresani, L., Ray, J., LeCun, Y., Paluri, M. (2018). A Closer Look at Spatiotemporal Convolutions for Action Recognition. https://arxiv.org/abs/1711.11248.

Xie, S., Sun, C., Huang, J., Tu, Z., Murphy, K. (2018). Rethinking Spatiotemporal Feature Learning: Speed-Accuracy Trade-offs in Video Classification. https://arxiv.org/abs/1712.04851.

Zhou, H., Zhou, W., Qi, W., Pu, J., Li, H. (2021a). Improving Sign Language Translation with Monolingual Data by Sign Back-Translation. https://arxiv.org/abs/2105.12397.

Zhou, Z., Tam, V.W.L., Lam, E.Y. (2022). A cross-attention BERT-based framework for continuous sign language recognition. *IEEE Signal Processing Letters*, 29, 1818–1822. https://doi.org/10.1109/LSP.2022.3199665.

Zhou, Z., Lui, K.-S., Tam, V.W.L., Lam, E.Y. (2021b). Applying (3+2+1)D residual neural network with frame selection for Hong Kong sign language recognition. In: *2020 25th International Conference on Pattern Recognition (ICPR)*, pp. 4296–4302. https://doi.org/10.1109/ICPR48806.2021.9412075.

M. Kalinowski is a PhD candidate at the Faculty of Electronics, Telecommunications and Informatics, Gdańsk University of Technology (Gdansk Tech). He completed his studies in 2023 at the Faculty of Electronics, Telecommunications, and Informatics, Gdansk Tech, specializing in artificial intelligence. During his studies, he took part in a research project that was recognized with the Dean's Award. Currently, he is conducting research on sign language processing using deep learning methods. A particular area of interest is sign language recognition and representation learning using multimodal approaches. His broader interests include generative models and agentic AI systems.

B. Kostek is a professor at the Faculty of Electronics, Telecommunications and Informatics, Gdansk University of Technology, Poland. She is a full member of the Polish Academy of Sciences and a fellow of the Audio Engineering Society and the Acoustical Society of America. Her primary scientific interests include signal processing, psychoacoustics, multimedia, music information retrieval, cognitive and behavioural processing, as well as the applications of machine learning to these domains. She is the recipient of many prestigious research awards, including those of the Prime Minister of Poland (twice), the Ministry of Science, and the Polish Academy of Sciences. She was the editor-in-chief of the *Journal of the Audio Engineering Society*, as well as Associate Editor of IEEE/ACM TASLP and Guest Editor of JASA, JIIS, and JAES.