Multimedia Systems (2025) 31:327
https://doi.org/10.1007/s00530-025-01896-1

REGULAR PAPER

Teaching authentic sign language through multiple representation learning

Qiaoyun Zhang[1] · Chih-Yung Chang[2] · Christopher Chuang[2] · Wen-Hwa Liao[3] · Diptendu Sinha Roy[4]

Received: 10 August 2024 / Accepted: 3 June 2025

Abstract

Sign language recognition plays a crucial role in bridging the communication gap between hearing-impaired and hearing individuals. However, traditional teaching systems often rely on expert systems or rule-based approaches for recognition, which struggle to meet the needs of learners at different levels. This paper introduces a novel Sign Language Teaching and Scoring System (SLTS) based on multi-model collaboration, aimed at improving learning efficiency and accuracy for diverse learners. The proposed SLTS employs teaching strategies suitable for both beginners and advanced learners, offering a comprehensive solution for sign language education through multiple representation learning. Specifically, for beginners, it uses an improved Siamese Long Short-Term Memory (LSTM) module to facilitate passive learning. This approach analyzes individual gestures by comparing them to conventional sign language, allowing novices to focus on mimicking movements and establishing a solid foundation in sign language norms. For advanced learners, the proposed SLTS implements an active learning approach using an enhanced Convolutional LSTM (ConvLSTM) module to handle more complex sign language vocabulary. The system captures both spatial and temporal features of gestures, enhancing learners' fluency and expressiveness in real communication scenarios. The experimental results in real-world environments demonstrate that the proposed SLTS significantly outperforms existing methods in recognition accuracy, proving its effectiveness and advanced nature.

Keywords Behavior recognition · Human–computer interaction · Multiple representation learning

1 Introduction

Sign language serves as a vital communication medium for the deaf and hard-of-hearing communities, facilitating interaction with the hearing world. However, the complexities of sign gestures, characterized by complex spatial and temporal patterns, present significant challenges for computational recognition systems. Accurately capturing and analyzing these gestures is crucial for seamless interaction between hearing and non-hearing individuals [1, 2]. Traditional recognition systems [3] primarily focus on directly identifying sign language gestures, often neglecting the varied learning needs of individuals at different proficiency levels. This gap highlights the need for more adaptive approaches that meet the diverse needs of learners, ranging from beginners to advanced signers.

One of the key innovations of this study is the development of targeted strategies for both beginner and advanced learners of sign language. By recognizing that learners at different proficiency levels require distinct support, the proposed SLTS offers a two-tiered approach: beginners focus on

Communicated by Bing-kun Bao.

✉ Chih-Yung Chang
cychang@mail.tku.edu.tw

Qiaoyun Zhang
zqyun@chzu.edu.cn

Christopher Chuang
chrischuang@gmail.com

Wen-Hwa Liao
whliao@ntub.edu.tw

Diptendu Sinha Roy
diptendu.sr@nitm.ac.in

[1] School of Artificial Intelligence, Chuzhou University, Chuzhou 239000, China

[2] Department of Computer Science and Information Engineering, Tamkang University, New Taipei 25137, Taiwan

[3] Institute of Information and Decision Sciences, National Taipei University of Business, Taipei City 100, Taiwan

[4] Department of Computer Science and Engineering, National Institute of Technology Meghalaya, Shillong 793003, India

Published online: 21 August 2025

learning basic hand gestures, while advanced learners refine their skills through complex gesture recognition and analysis. This design allows the proposed SLTS to better cater to the specific needs of each group, enabling deaf students to gradually enhance their signing accuracy and fluency in Taiwanese Sign Language according to their skill levels.

In sign language learning, students often encounter difficulties not in understanding the meaning of gestures, but in executing them with the required precision. Even for commonly used sign gestures whose meanings are already familiar, subtle variations in hand positioning, motion trajectory, and timing can significantly impact their clarity and correctness. The so-called familiarity comes from classroom instruction, where teachers introduce these gestures. However, teaching and learning are different processes, and there is often a gap between learning and practical application. Students practice at home without their teachers present, yet they still need feedback to refine their gestures.

Traditional teaching methods often rely on self-practice or instructor feedback, which may be inconsistent or insufficient for precise correction. Although sign language learners can mimic gestures by watching videos, the lack of targeted feedback mechanisms makes it difficult for them to identify and correct subtle hand gesture errors. For example, similar gestures may be misunderstood due to differences in speed, hand shape, or angle deviations. To address this issue, this study employs Siamese LSTM to compute the similarity between learners' gestures and standard gestures, thereby providing personalized feedback that helps learners precisely adjust their gestures and improve learning efficiency. To further bridge this gap, the proposed SLTS shifts the focus from semantic interpretation to performance accuracy. By leveraging advanced recognition techniques, SLTS provides real-time, data-driven feedback on key gesture parameters, enabling students to refine their execution systematically. This approach ensures that learners not only understand the signs but can also reproduce them accurately, facilitating more effective communication in real-world scenarios.

Current systems that leverage traditional machine learning models demand extensive feature engineering and domain expertise [4], often resulting in models tailored to specific datasets but lacking adaptability for broader applications. To address these limitations, researchers have turned to deep learning methods [5–9], which can automatically learn feature representations from raw data. For example, Yirtici et al. [6] and Lee et al. [7] have noted that while convolutional neural networks (CNNs) and recurrent neural networks (RNNs) can capture either spatial or temporal aspects of sign language, they often struggle to integrate these dimensions simultaneously. This lack of integration can result in an incomplete understanding of gestures, especially when discerning gestures with similar visual features but different meanings. Moreover, these systems do not differentiate between learners' proficiency levels, providing uniform solutions that fail to meet the diverse needs of learners at different stages.

To address these challenges, this paper introduces a novel SLTS specially designed for Taiwanese Sign Language education. The proposed SLTS offers tailored approaches for different learner levels, recognizing the need for specialized educational strategies to improve both learning efficiency and accuracy. For beginners, the system selects specific signs for learners to imitate, emphasizing similarity to a standard reference. Conventional sign language refers to a structured and standardized system of gestures commonly used in formal education and textbooks, providing a consistent and reproducible framework for teaching. In contrast, natural sign language is the dynamic and context-dependent language used by deaf individuals in daily communication, which may vary regionally and evolve over time. Given these inherent differences, distinguishing between Conventional and natural signs allows the system to assess learners' proficiency more accurately, thereby enhancing the effectiveness and applicability of the recognition model in instructional settings. To facilitate this, the proposed SLTS employs a Siamese LSTM network [10, 11] for passive learning, enabling learners to compare their gestures with standard references and refine their movements accordingly. For advanced learners, who can freely practice a variety of signs, the system adopts an active learning strategy using a ConvLSTM network [12–14]. This model first extracts spatial features through convolutional layers and then analyzes both spatial and temporal characteristics using LSTM layers to classify the sign and provide feedback. The main contributions of this paper are summarized as follows:

(1) **Utilizing a siamese LSTM model to reduce computational overhead:** The proposed SLTS utilizes a Siamese LSTM to efficiently recognize and compare basic gestures, reducing computational demands and helping beginners develop foundational skills.
(2) **Applying ConvLSTM to capture spatial–temporal features:** The proposed SLTS applies ConvLSTM to effectively capture complex spatial–temporal features, allowing for precise recognition of advanced sign language gestures.
(3) **Adopting tailored learning approaches for diverse learners:** The proposed SLTS provides customized learning paths for both beginners and advanced learners by leveraging Siamese LSTM and ConvLSTM networks. This targeted strategy ensures that each learner receives appropriate support according to their proficiency level.

The rest of the paper is organized as follows: Sect. 2 presents a review of related work to establish the context of the study. Section 3 describes the problem statement and research objectives. In Sect. 4, the proposed SLTS mechanism is thoroughly detailed. Section 5 assesses the performance of the proposed SLTS. Lastly, Sect. 6 summarizes the main conclusions and offers suggestions for future work.

2 Related work

This section describes the relevant literature on sign language identification, which is integral to machine learning and deep learning approaches.

2.1 Machine learning approaches for sign language recognition

Sign language recognition has made significant strides with the application of machine learning techniques. For instance, Talukdar and Bhuyan [15] proposed a vision-based continuous sign language spotting system using a Gaussian Hidden Markov Model (HMM), which handled the temporal dynamics in continuous sign language sequences. However, HMM requires a large amount of training data to accurately model variations in sign language, especially when dealing with continuous sign language sequences. Xie et al. [16] explored federated learning, specifically applying spike neural networks for traffic sign recognition. However, federated learning may lead to inconsistent model performance due to variations in data distribution and device resources across different local models.

Some researchers [17–19] integrated elements of both traditional machine learning and deep learning to improve recognition performance. For example, Katoch et al. [17] developed an Indian sign language recognition system by combining speeded-up robust features with support vector machines for feature extraction and classification, respectively. They also incorporated a CNN to create a hybrid approach. However, CNN may have been limited to preprocessing extracted features, potentially underutilizing the CNN's capacity for automatic feature extraction.

To address the above issues, the proposed SLTS utilizes LSTM to effectively handle temporal dependencies and extract relevant features in advance. Additionally, the use of LSTM enhances the model's capability to extract and utilize features earlier in the recognition process, leading to improved overall performance in sign language recognition tasks.

2.2 Deep learning-based sign language recognition

The application of deep learning in sign language recognition has significantly advanced the field by enabling models to capture complex patterns within sign language data. Wadhawan and Kumar [20] developed a CNN-based system for static signs, focusing on automatic gesture classification. However, their approach struggled with dynamic sequences, which were critical for effective continuous sign language recognition. To tackle the dynamic aspects of sign language, Sharma et al. [21] employed deep transfer learning for continuous sign language recognition. However, their reliance on isolated sign data restricted the system's adaptability across different signers and environments. Zuo and Mak [22] aimed to improve robustness by introducing consistency constraints and signer removal techniques. However, their approach was still limited by the complexity of continuous sequences and variations in signer characteristics.

Rastgoo et al. [23] enhanced the recognition process by integrating multi-view hand skeleton data with deep learning techniques, offering a more comprehensive capture of spatial information. Despite this, their model was constrained by its dependency on skeletal data, which may not fully consider the nuances of hand gestures. In a different vein, Bilge et al. [24] explored zero-shot learning for sign language recognition, targeting the recognition of unseen signs. While innovative, their model's effectiveness was limited by the difficulties associated with learning from minimal or non-existent data samples. Abdullahi et al. [25] and Han et al. [26] addressed the spatiotemporal complexities in sign language recognition. Abdullahi et al. [25] introduced an end-to-end Fourier network for 3D sign language recognition, focusing on spatial–temporal features. Han et al. [26] advanced the field by using R(2 + 1)D networks with spatial–temporal-channel attention to enhance feature focus across dimensions. However, these systems did not differentiate between learners' proficiency levels, providing uniform solutions that failed to meet the diverse needs of learners at different stages.

Recent work has explored Transformer models for temporal-spatial feature extraction in various tasks, such as the approach presented by Pareek et al. [27], which modeled human activity recognition through a Transformer architecture with attention mechanisms to capture intricate spatial–temporal dependencies. While Transformer models have demonstrated high performance across many domains, their computation and memory demands make them better suited for large-scale datasets and substantial processing resources, often challenging for real-time applications. Conversely, the proposed SLTS employs ConvLSTM, which integrates convolutional and recurrent layers to effectively capture spatial and temporal features simultaneously. This architecture aligns well with the sequential nature of complex

sign language gestures and offers efficient processing for smaller datasets while ensuring rapid inference. Given the real-time requirements of sign language recognition, ConvLSTM offers a well-balanced trade-off between computational efficiency and temporal-spatial modeling performance, making it a practical and effective solution.

In comparison with the related studies, the proposed SLTS overcomes these challenges by employing a Siamese LSTM structure for beginners to efficiently compare designated specific signs. For advanced learners, it incorporates a ConvLSTM architecture that processes each frame sequentially, capturing both spatial and temporal features simultaneously. Additionally, this multimodal integration effectively differentiates learners at different proficiency levels, providing tailored feedback and instruction.

3 Notations, assumptions and problem descriptions

This section provides an overview of the notations, assumptions, problem descriptions, and objectives of this paper.

3.1 Notations and assumptions

There is a Taiwanese Sign Language video, consisting of m continuous frames, denoted as $\mathcal{F} = (f_1, f_2, ..., f_m)$, where f_i $(1 \le i \le m)$ represents the i-th frame captured at the specific time t_i. Each frame f_i is thus a data point comprising both the temporal information, denoted as (f_i, t_i) and spatial information of the gesture, denoted as $\mathcal{S}'_i = \{s'_{i,1}, s'_{i,2}, \dots, s'_{i,k}, \dots, s'_{i,s}\}$, where $s'_{i,k} \in \mathcal{S}'_i$ and $\int$ denote the coordinates of k-th skeletal key point and quantity of skeletal key points, respectively.

This representation acknowledges that each frame is not only a static image but also a part of a temporal sequence, capturing the dynamic nature of the gestures as they evolve. These frames capture a series of dynamic gestures that are elements of sign language, a rich visual means of communication used primarily by the deaf and hard-of-hearing community. Sign language communication involves hand shapes, orientations, movements, and facial expressions to convey meaning. Each frame f_i within the video $\mathcal{F}$ might contain a distinct gesture or a segment of a broader gesture sequence, demanding precise identification and interpretation.

3.2 Problem descriptions

Consider a sign language identification mechanism $\mathcal{M}$. There are n sigh language words, denoted by $\mathcal{C} = \{c_1, c_2, \dots, c_n\}$. Let $\mathcal{U} = [\nu_{a,b}]_{n\times n}$ denote the number of samples that are of the category $c_a \in \mathcal{C}$ and predicted to be the category $c_b \in \mathcal{C}$. For each frame $f_i \in \mathcal{F}$, if the actual category of the sample f_i is c_a and the predicted category is c_b, the proposed SLTS updates the corresponding value of $\nu_{a,b}$ in matrix $\mathcal{U}$. That is

$$\nu_{a,b} = \nu_{a,b} + 1. \tag{1}$$

Let $TP_k^{\mathcal{M}}$ denote true positive for c_k using mechanism $\mathcal{M}$. The value of $TP_k^{\mathcal{M}}$ can be calculated by Eq. (2).

$$TP_k^{\mathcal{M}} = \nu_{k,k}. \tag{2}$$

Let $FP_k^{\mathcal{M}}$, $FN_k^{\mathcal{M}}$ and $TN_k^{\mathcal{M}}$ denote false positive, false negative and true negative for c_k, respectively. The value of $FP_k^{\mathcal{M}}$, $FN_k^{\mathcal{M}}$ and $TN_k^{\mathcal{M}}$ can be computed using Eqs. (3), (4), and (5), respectively.

$$FP_k^{\mathcal{M}} = \sum_{x=1, x\neq k}^{n} \nu_{x,k}, \tag{3}$$

$$FN_k^{\mathcal{M}} = \sum\nolimits_{y=1, y\neq k}^{n} \nu_{k,y}, \tag{4}$$

$$TN_k^{\mathcal{M}} = \sum\nolimits_{x=1, x\neq k}^{n} \sum_{y=1, y\neq k}^{n} \nu_{x,y}. \tag{5}$$

Let $TP^{\mathcal{M}}$, $FP^{\mathcal{M}}$, $FN^{\mathcal{M}}$ and $TN^{\mathcal{M}}$ denote true positive, false positive, false negative, and true negative of all $\mathcal{C}$, respectively. The value of $TP^{\mathcal{M}}$, $FP^{\mathcal{M}}$, $FN^{\mathcal{M}}$ and $TN^{\mathcal{M}}$ can be measured by applying Eqs. (6)–(9), respectively.

$$TP^{\mathcal{M}} = \sum_{k=1}^{n} TP_k^{\mathcal{M}}. \tag{6}$$

$$FP^{\mathcal{M}} = \sum_{k=1}^{n} FP_k^{\mathcal{M}}. \tag{7}$$

$$FN^{\mathcal{M}} = \sum_{k=1}^{n} FN_k^{\mathcal{M}}. \tag{8}$$

$$TN^{\mathcal{M}} = \sum_{k=1}^{n} TN_k^{\mathcal{M}}. \tag{9}$$

Let $\mathcal{P}^{\mathcal{M}}$, $\mathcal{R}^{\mathcal{M}}$ and $\mathcal{A}^{\mathcal{M}}$ denote the precision, recall, and accuracy of $\mathcal{U}$, respectively. The values of $\mathcal{P}^{\mathcal{M}}$, $\mathcal{R}^{\mathcal{M}}$ and $\mathcal{A}^{\mathcal{M}}$ can be evaluated by applying Eqs. (10)–(12), respectively.

$$\mathcal{P}^{\mathcal{M}} = \frac{TP^{\mathcal{M}}}{TP^{\mathcal{M}} + FP^{\mathcal{M}}}. \tag{10}$$

$$\mathcal{R}^{\mathcal{M}} = \frac{TP^{\mathcal{M}}}{TP^{\mathcal{M}} + FN^{\mathcal{M}}}. \tag{11}$$

$$\mathcal{A}^{\mathcal{M}} = \frac{TP^{\mathcal{M}} + TN^{\mathcal{M}}}{TP^{\mathcal{M}} + TN^{\mathcal{M}} + FP^{\mathcal{M}} + FN^{\mathcal{M}}}. \tag{12}$$

Let $\mathcal{F}1^{\mathcal{M}}$ denote the F1-score of $\mathcal{U}$ obtained by the prediction of mechanism $\mathcal{M}$. The value of $\mathcal{F}1^{\mathcal{M}}$ can be calculated by Eq. (13).

$$\mathcal{F}1^{\mathcal{M}} = \frac{2 * \mathcal{P}^{\mathcal{M}} * \mathcal{R}^{\mathcal{M}}}{\mathcal{P}^{\mathcal{M}} + \mathcal{R}^{\mathcal{M}}}. \tag{13}$$

3.2.1 Objective

Let ψ is the set of all possible mechanisms employed to identify the sign language video $\mathcal{F}$. The main goal of this paper is to find the most effective mechanism $\mathbb{M} \in \psi$, which satisfies Eq. (14).

F1-score objective:

$$\mathbb{M} = arg\underset{\mathcal{M}\in\psi}{max}\mathcal{F}1^{\mathcal{M}}. \tag{14}$$

While achieving the best solution, some constraints should be satisfied. Let $G_{\mathbb{N}}(t_i)$ represent the skeleton changes of a gesture at a normal speed at a time slot t_i. This function should include the positions of all key points of the gesture, capturing the ideal state of the gesture at normal speed. Consider a time scaling factor λ_1, where $\lambda_1 > 0$. Time scaling suggests that the execution speed of the gesture can be either faster or slower than that of the normal gesture. Consequently, when the speed of the gesture changes, the function of the gesture can be represented as $G_{\mathbb{U}}(\lambda_1 t_i)$.

Let $D_{speed}(G_{\mathbb{U}}(\lambda_1 t_i), G_{\mathbb{N}}(t_i))$ denote the difference between two gestures $G_{\mathbb{U}}(\lambda_1 t_i)$ and $G_{\mathbb{N}}(t_i)$ which are performed over different periods $\lambda_1 t_i$ andt_i, respectively. The value of $D_{speed}(G_{\mathbb{U}}(\lambda_1 t_i), G_{\mathbb{N}}(t_i))$ is defined by Eq. (15).

$$D_{speed}\left(G_{\mathbb{U}}\left(\lambda_1 t_i\right), G_{\mathbb{N}}\left(t_i\right)\right) = \sum\nolimits_{k=1}^{\jmath} \frac{d\left(s_{i,k}^{\mathbb{U}}, s_{i,k}^{\mathbb{N}}\right)}{\omega_i}, \tag{15}$$

where $\jmath$ denotes the quantity of skeletal key points, and $d\left(s_{i,k}^{\mathbb{U}}, s_{i,k}^{\mathbb{N}}\right)$ is the Euclidean distance between the k-th skeletal key points of $s_{i,k}^{\mathbb{U}}$ and $s_{i,k}^{\mathbb{N}}$. The weight value of ω_i is calculated based on the time difference, which can be designed as a time-sensitive function, reflecting that frames closer in time should contribute more significantly to the overall difference. The value of ω_i is calculated by applying Eq. (16).

$$\omega_i = e^{-|\lambda_1 t_i - t_i|}. \tag{16}$$

The first constraint is the Speed-Variant Similarity. This means that a gesture is considered correctly executed if its similarity to the conventional gesture remains within a predefined threshold at any time slot t_i, regardless of the speed at which the gesture is performed.

(1) Speed-variant difference constraint

$$D_{speed}\left(G_{\mathbb{U}}\left(\lambda_1 t_i\right), G_{\mathbb{N}}\left(t_i\right)\right) \le \epsilon_1, for 1 \le i \le m, \tag{17}$$

where ϵ_1 denotes a difference threshold within which the gesture is considered correct, and m denotes the total number of frames. This constraint ensures that regardless of the execution speeds of the gestures if the similarity between the gestures $G_{\mathbb{U}}(\lambda_1 t_i)$ and $G_{\mathbb{N}}(t_i)$, remains the allowed difference ϵ_1, the gesture $G_{\mathbb{U}}(\lambda_1 t_i)$ is accepted as correct, as compared with the conventional gesture $G_{\mathbb{N}}(t_i)$.

Let $G_x(t_i)$ and $G_y(t_j)$ represent functions describing the skeletal variations of two distinct sign language gestures x and y at t_i and t_j, respectively. Let $D_{distinct}(G_x(t_i), G_y(t_j))$ denote the distinctiveness between $G_x(t_i)$ and $G_y(t_j)$. The value of $D_{distinct}(G_x(t_i), G_y(t_j))$ can be calculated using a suitable metric that quantifies the dissimilarity between their skeletal structures, potentially based on a comprehensive assessment of their key points' positions. That is

$$D_{distinct}\left(G_x\left(t_i\right), G_y\left(t_j\right)\right) = \sum\nolimits_{k=1}^{\jmath} d\left(s_{i,k}, s_{j,k}\right), \tag{18}$$

where $d\left(s_{i,k}, s_{j,k}\right)$ is the Euclidean distance between the k-th skeletal key points of $s_{i,k}$ and $s_{j,k}$.

The second constraint is Sign Language Gesture Distinctiveness, which is used to ensure that different sign language gestures are sufficiently distinctive. This distinction is important to accurately differentiate between various gestures.

(2) *Sign language gesture distinctiveness constraint*

$$\forall t_i, t_j (x \ne y), D_{distinct}\left(G_x\left(t_i\right), G_y\left(t_j\right)\right) \ge \epsilon_2, \tag{19}$$

where ϵ_2 denotes a threshold indicating the minimum difference in similarity for two sign language gestures to be considered significantly distinct. This threshold should be large enough to ensure a clear distinction between gestures. This means that for any two different sign language gestures x and y, their similarity should at least reach ϵ_2, ensuring that they have sufficient visual or functional distinctiveness.

4 The proposed SLTS mechanism

This paper proposes a novel sign language identification mechanism, called SLTS, specifically designed for accurate recognition of sign gestures. This design allows the proposed SLTS to better cater to the specific needs of each group, enabling deaf students to gradually enhance their signing accuracy and fluency based on their skill level using Taiwanese Sign Language. As shown in Fig. 1, the proposed SLTS comprises three phases: *Data Preprocessing Phase,*

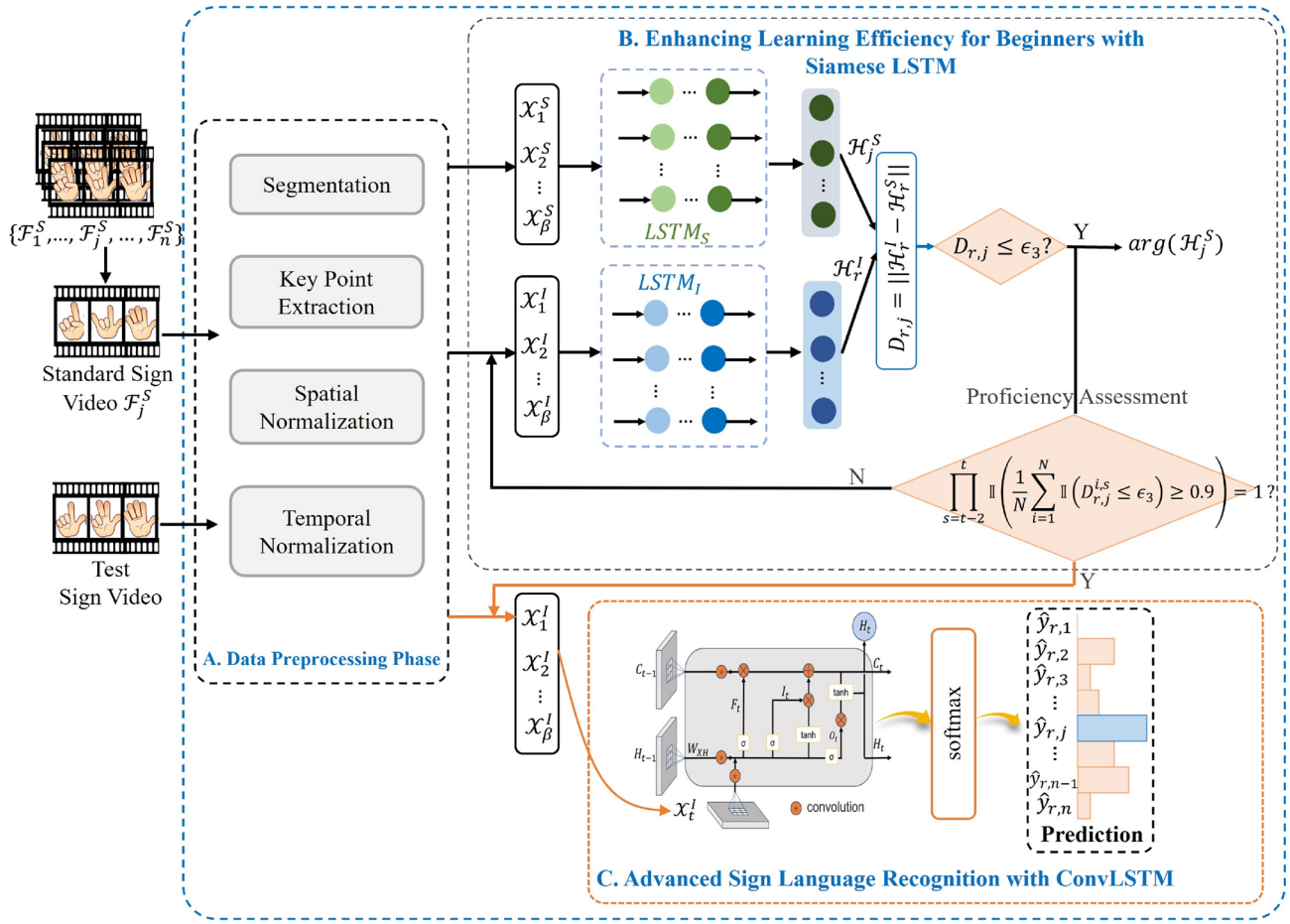


Fig. 1 The architecture of the proposed SLTS mechanism

Enhancing Learning Efficiency for Beginners with Siamese LSTM, and Advanced Sign Language Recognition with ConvLSTM. The first phase focuses on data preprocessing, which is further divided into three tasks, including Input Video Segmentation Task, Key Point Extraction Task as well as Temporal and Spatial Normalization Task. The second phase utilizes a Siamese LSTM network to efficiently compare sign language words, enabling rapid identification by specifically targeting signs for beginners. In the third phase, the proposed SLTS employs a ConvLSTM model to extract both temporal and spatial features for advanced learners. This model thoroughly analyzes each frame of the sign language video to accurately identify specific words, thereby enhancing the overall precision and reliability of the recognition process.

The proposed SLTS adopts a dual learning strategy to support deaf students with different levels of sign language proficiency. For beginners, the system uses a Siamese LSTM model to facilitate passive learning, allowing learners to imitate and compare their gestures with conventional sign language references. While conventional sign language primarily consists of individual lexical signs rather than syntactically structured sentences, each sign gesture still exhibits sequential motion dependencies, including trajectory, velocity, and transitional movements between hand configurations. The Siamese LSTM captures these temporal dynamics, allowing for more precise differentiation between correct and incorrect executions of a given sign. For advanced learners, the proposed SLTS integrates a ConvLSTM model, which actively processes complex gesture sequences to capture both spatial and temporal patterns in signing. This approach enhances learners' ability to refine subtle hand movements and improve gesture fluency, effectively supporting advanced signers in mastering nuanced expressions. By combining these tailored learning strategies, the proposed SLTS provides a comprehensive educational tool that adapts to learners' varying levels, thus meeting the unique requirements of sign language education. The following presents the details of these phases.

4.1 Data preprocessing phase

This phase aims to clean and standardize the input sign language videos, ensuring a consistent and reliable dataset for the following learning phases. It contains three tasks: Input Video Segmentation, Key Point Extraction, as well as Spatial and Temporal Normalization.

4.1.1 Input video segmentation task

To address length limitations, each sign language video is segmented into multiple fixed-length clips. This segmentation ensures that the model can handle sign language content without being affected by excessive video length.

Suppose a sign video consists of m continuous frames, denoted as $\mathcal{F} = (f_1, f_2, ..., f_m)$. As shown in Fig. 2, to facilitate this segmentation process, let ζ denote taking one frame every ζ frame from $\mathcal{F}$, and let β represent the predetermined duration for each segment, both measured in frames. Utilizing this parameter, the proposed SLTS systematically divides the video $\mathcal{F}$ frame by frame with each segment spanning β frames, generating numerous segments. This division strategy generates a series of segments, ensuring that the video content is efficiently processed and analyzed.

It is important to note that in scenarios where the number of frames within a segment falls short of β frames, padding techniques are employed to bridge the gap, ensuring uniformity across segments. This adaptive approach allows for seamless processing of videos of varying lengths, enhancing the robustness and applicability of the model. Let ε denote the number of partitioned segments. The value of ε can be denoted by Eq. (20).

$$\varepsilon = \lceil \frac{m}{\zeta \times \beta} \rceil. \tag{20}$$

Once segmented, the videos are represented as $\mathcal{F} = (\mathcal{F}_1, \mathcal{F}_2, ..., \mathcal{F}_\varepsilon)$ with each segment $\mathcal{F}_r \in \mathcal{F}$ comprising precisely β frames. Within each segment, the frames are indexed from a starting point, denoted as $\mathcal{F}_r^{start} = f_{x+1}$, and continue up to an endpoint, denoted as $\mathcal{F}_r^{end} = f_{x+\beta}$. The value of each frame f_{x+j}, within a segment can be derived using Eq. (21).

$$f_{x+j} = f_{(r-1)\times\beta+j}. \tag{21}$$

After this segmentation task, the input video is transformed into multiple fixed-length clips, which serve as the foundational input for the keypoint extraction task.

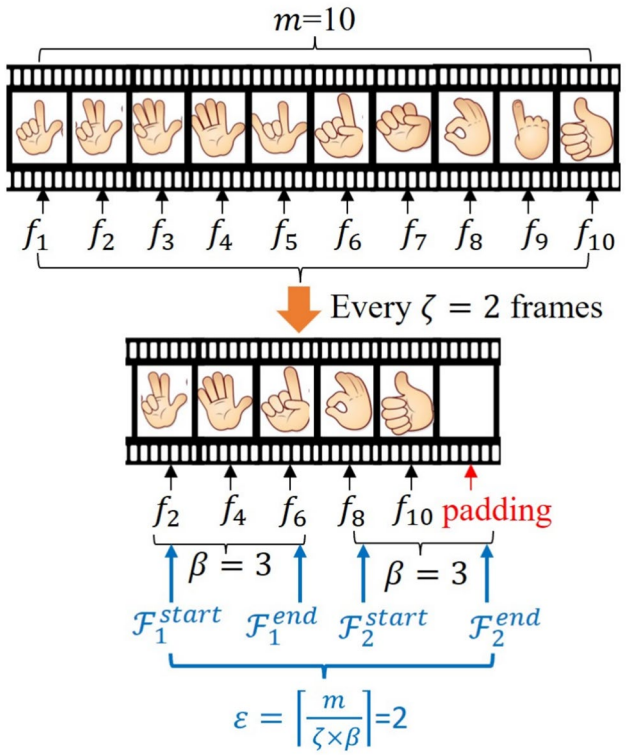


Fig. 2 The process of the input video segmentation

4.1.2 Key point extraction task

Using the segmented clips, MediaPipe is employed to extract skeletal key points from each frame, capturing subtle hand movement changes in sign language. The MediaPipe framework identifies skeletal points in each frame, creating a dataset that provides essential features for recognizing sign language. For frame f_i, the set of skeletal key points using MediaPipe is denoted as $\mathcal{S}_i' = \{s_{i,1}', s_{i,2}', \ldots, s_{i,s}'\}$, where $s_{i,k}' \in \mathcal{S}_i'$ and s denote the coordinates of k-th skeletal key point and quantity of skeletal key points, respectively. Each key point $s_{i,k}' \in \mathcal{S}_i'$ can be precisely defined by Eq. (22).

$$s_{i,k}' = \left(x_{i,k}', y_{i,k}', z_{i,k}'\right), \tag{22}$$

where $x_{i,k}'$, $y_{i,k}'$, and $z_{i,k}'$ denote the values of $s_{i,k}'$ along the x, y, and z axes, respectively.

These coordinates represent the spatial positioning of each skeletal key point within the frame, enabling a comprehensive understanding of hand and body postures. The precise determination of these coordinates facilitates accurate representation and analysis of sign language gestures, enhancing the model's ability to discern subtle nuances in expression and movement.

The utilization of MediaPipe not only enriches the informational content of video data but also furnishes the sign language recognition model with more accurate and comprehensive inputs. This meticulous processing step ensures that the model attains higher levels of accuracy and reliability in recognizing sign language gestures.

4.1.3 Spatial and temporal normalization task

This task is crucial in ensuring the precision and consistency of sign language recognition within the proposed SLTS framework. This task is divided into two steps: Spatial Normalization and Temporal Normalization.

The Spatial Normalization aims to standardize the spatial displacement values of each key skeletal point, ensuring that the spatial characteristics of different gestures are comparable scales. To achieve this, the proposed SLTS utilizes Min–Max normalization, a technique that adjusts the spatial displacement values within a specified range, typically between 0 and 1. This process involves subtracting the minimum value of the skeletal key point and then dividing it by the range of the skeletal key point. Let $x'_{i,k}$, $y'_{i,k}$, and $z'_{i,k}$ denote the values of $s'_{i,k}$ along the x, y, and z axes, respectively. Let $x_{i,k}$, $y_{i,k}$, and $z_{i,k}$ denote the normalized value of $x'_{i,k}$, $y'_{i,k}$, and $z'_{i,k}$, respectively. The transformation of $x'_{i,k}$, $y'_{i,k}$, and $z'_{i,k}$ into $x_{i,k}$, $y_{i,k}$, and $z_{i,k}$ can be computed from Eqs. (23)– (25), respectively.

$$x_{i,k} = \frac{x'_{i,k} - \min\limits_{1\le i\le m} x'_{i,k}}{\max\limits_{1\le i\le m} x'_{i,k} - \min\limits_{1\le i\le m} x'_{i,k}}, \tag{23}$$

$$y_{i,k} = \frac{y'_{i,k} - \min\limits_{1\le i\le m} y'_{i,k}}{\max\limits_{1\le i\le m} y'_{i,k} - \min\limits_{1\le i\le m} y'_{i,k}}, \tag{24}$$

$$z_{i,k} = \frac{z'_{i,k} - \min\limits_{1\le i\le m} z'_{i,k}}{\max\limits_{1\le i\le m} z'_{i,k} - \min\limits_{1\le i\le m} z'_{i,k}}. \tag{25}$$

Let $\mathcal{S}_i = \{s_{i,1}, s_{i,2}, \ldots, s_{i,s}\}$ denote the set of skeletal key points from the frame f_i after spatial normalization, where $s_{i,k} \in \mathcal{S}_i$ denotes the spatial normalized coordinates of the k-th skeletal key point. The value of $s_{i,k}$ can be defined as Eq. (26).

$$s_{i,k} = \left(x_{i,k}, y_{i,k}, z_{i,k}\right). \tag{26}$$

By applying these transformations, the proposed SLTS ensures that all gestures have a uniform spatial representation, facilitating accurate comparison and analysis across various sign gestures.

The Temporal Normalization aims to adjust each time window from each segment $\mathcal{F}_r \in \mathcal{F}$ by leveraging the spatial displacement between consecutive frames f_i and f_{i+1}. This analysis is further refined using a sliding window approach. Let $\Delta_t(i, i+1)$ denote the spatial displacement between f_i and f_{i+1}. The value of $\Delta_t(i, i+1)$ can be derived by Eq. (27).

$$\Delta_t(i, i+1) = \sum_{k=1}^{s} \sqrt{\left(\left(x_{i,k} - x_{i+1,k}\right)^2 + \left(y_{i,k} - y_{i+1,k}\right)^2 + \left(z_{i,k} - z_{i+1,k}\right)^2\right)}, \tag{27}$$

where $\Delta_t(i, i+1)$ denotes the change in skeletal key points between frames. A higher value of $\Delta_t(i, i+1)$ signifies a more pronounced average increase in movement from the start of the event, while a lower value indicates a greater average decrease in movement starting from that time step.

To accurately identify the optimal start and end times of sign language gestures, let I_{max} and I_{min} denote the maximal and minimal spatial displacement between $f_i \in \mathcal{F}_r$ and $f_{i+1} \in \mathcal{F}_r$, respectively. That is

$$I_{max} = \underset{\forall i}{argmax} \Delta_t(i, i+1), \tag{28}$$

$$I_{min} = \underset{\forall i}{argmin} \Delta_t(i, i+1). \tag{29}$$

The proposed SLTS utilizes these values to accurately identify the starting and ending of sign language gestures, subsequently removing video frames outside these bounds. This careful process results in a refined set of shortened videos, making it easier to analyze and understand. Let $\mathcal{F}'_r$ denote the video proposed by temporal normalization. The value of $\mathcal{F}'_r$ can be calculated by Eq. (30).

$$\mathcal{F}'_r = \left\{f_i | I_{max} \le i \le I_{max} + \beta\right\}, \tag{30}$$

where β represents the predetermined duration for each segment $\mathcal{F}'_r$, measured in frames. By implementing this spatial and temporal normalization process, the proposed SLTS ensures enhanced precision and consistency in recognizing and analyzing sign language gestures.

The preprocessing phase prepares data for the next phase by ensuring consistent format and timing, making it easier for the model to compare different hand sign sequences accurately. The output from this phase feeds directly into the next phase, supporting the system in capturing the clearer gesture features.

4.2 Enhancing learning efficiency for beginners with siamese LSTM

In this phase, the proposed SLTS employs a passive learning approach for beginners using the Siamese LSTM to process the normalized key point data prepared during the data preprocessing phase. The Siamese LSTM is characterized by its dual-network architecture with shared weights, enabling it to simultaneously process two video sequences: conventional sign videos and videos from beginners. This model is primarily designed to determine whether the two sign language videos convey the same

sign word by calculating the distance between their key point features. If this distance falls below a predefined threshold, the input videos are considered to represent the same sign language with designates specific signs. This can allow beginners to focus on imitating movements and building foundational skills.

Let $\mathcal{F}_r^I = \{f_1^I, f_2^I, \ldots, f_\beta^I\}$ and $\mathcal{F}_j^S = \{f_1^S, f_2^S, \ldots, f_\beta^S\}$ are the videos from beginners and conventional video sequences that are taken as the two inputs of a Siamese LSTM. In these sequences, the video frames $f_i^I \in \mathcal{F}_r^I$ and $f_i^S \in \mathcal{F}_j^S$, each containing multiple key point skeleton information, are denoted by $f_i^I = \{s_{i,1}^I, s_{i,2}^I, \ldots, s_{i,s}^I\}$ and $f_i^S = \{s_{i,1}^S, s_{i,2}^S, \ldots, s_{i,s}^S\}$, respectively.

Let $\mathcal{X}_i^I = \{\mathrm{x}_{i,1}^I, \mathrm{x}_{i,2}^I, \ldots, \mathrm{x}_{i,s}^I\}$ and $\mathcal{X}_i^S = \{\mathrm{x}_{i,1}^S, \mathrm{x}_{i,2}^S, \ldots, \mathrm{x}_{i,s}^S\}$ denote the feature vectors of frames f_i^I and f_i^S, respectively. These feature vectors capture the key skeleton information within the frames. As shown in Fig. 3, let $\mathbb{X}_r^I$ and $\mathbb{X}_j^S$ denote the input feature vectors of $\mathcal{F}_r^I$ and $\mathcal{F}_j^S$, respectively. The values of $\mathbb{X}_r^I$ and $\mathbb{X}_j^S$ can be denoted by Eqs. (31) and (32), respectively.

$$\mathbb{X}_r^I = \begin{pmatrix} \mathcal{X}_1^I \\ \mathcal{X}_2^I \\ \vdots \\ \mathcal{X}_\beta^I \end{pmatrix} = \begin{pmatrix} \mathrm{x}_{1,1}^I & \cdots & \mathrm{x}_{1,s}^I \\ \vdots & \ddots & \vdots \\ \mathrm{x}_{\beta,1}^I & \cdots & \mathrm{x}_{\beta,s}^I \end{pmatrix}, \tag{31}$$

$$\mathbb{X}_j^S = \begin{pmatrix} \mathcal{X}_1^S \\ \mathcal{X}_2^S \\ \vdots \\ \mathcal{X}_\beta^S \end{pmatrix} = \begin{pmatrix} \mathrm{x}_{1,1}^S & \cdots & \mathrm{x}_{1,s}^S \\ \vdots & \ddots & \vdots \\ \mathrm{x}_{\beta,1}^S & \cdots & \mathrm{x}_{\beta,s}^S \end{pmatrix}. \tag{32}$$

The feature vectors $\mathbb{X}_r^I$ and $\mathbb{X}_j^S$ are then taken as the two inputs of the two parallel LSTM networks, $LSTM_I$ and $LSTM_S$, respectively. These two LSTM networks have identical structures and share weights and biases to ensure consistent behavior while processing sequence data. Each network independently processes its input sequences, gradually integrating the time-dependent features within the frames and generating a composite feature vector representing the entire video sequence.

Let $\mathcal{H}_r^I$ and $\mathcal{H}_j^S$ denote the final outputs of both sequences. They are then used to compute the distance using the Euclidean. Let $D_{r,j}$ denote the distance between $\mathcal{H}_r^I$ and $\mathcal{H}_j^S$. The value of $D_{r,j}$ can be derived using Eq. (33).

$$D_{r,j} = \left|\left|\mathcal{H}_r^I - \mathcal{H}_j^S\right|\right| \tag{33}$$

In the Siamese LSTM model, the contrastive loss function is used to enable the model to learn whether the two input samples are similar. Let L_s denote the loss function of the Siamese LSTM model. The value of L_s can be calculated using Eq. (34).

$$L_s = \frac{1}{2\varepsilon}\sum_{r=1}^{\varepsilon}((1 - y_{r,j}) \cdot D_{r,j}^2 + y_{r,j} \cdot max(0, \mu - D_{r,j})^2), \tag{34}$$

where ε denotes the batch size, μ is a hyperparameter defining the minimum distance that should be achieved between samples of different categories, and $y_{r,j}$ denotes the label. When both input videos display the same sign language, $y_{r,j} = 0$, the objective is to minimize $D_{r,j}$, reducing the distance between the feature vectors of videos that represent the same sign language, bringing them closer together. Conversely, when the videos display different sign languages, $y_{r,j} = 1$, the objective is to maximize $D_{r,j}$, ensuring the

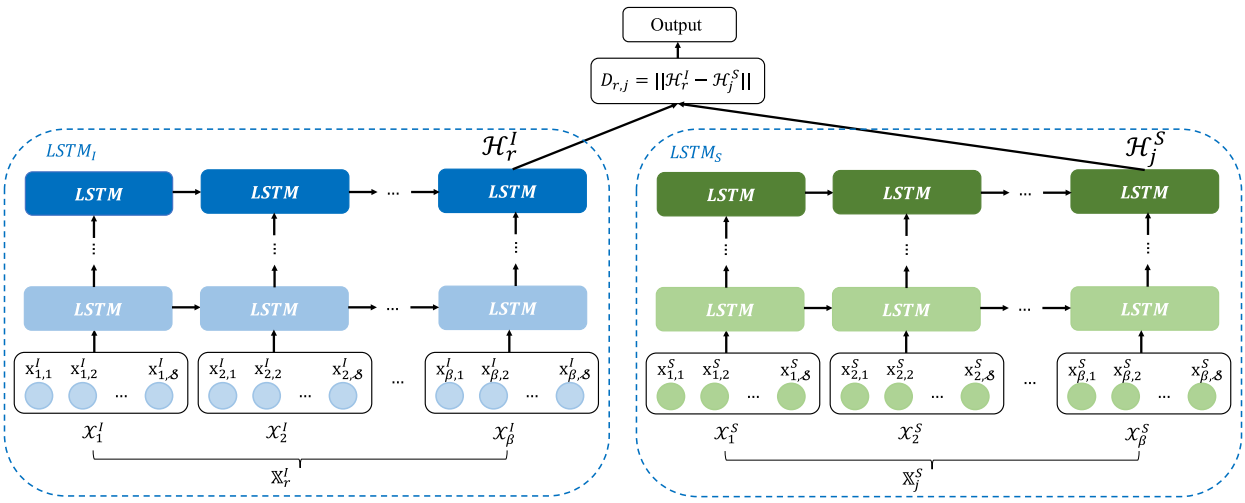


Fig. 3 The process of the Siamese LSTM

distance is at least μ. If $D_{r,j}$ is less than μ, the loss is forcibly increased by $(max(0, \mu - D_{r,j})^2)$ to encourage the model to distance the feature vectors of different sign languages further apart. In this way, the contrastive loss encourages the model to cluster samples of the same category closer together in the feature space while pushing samples of different categories apart, thereby enhancing the model's performance in sign language video recognition tasks.

During the inference phase, let ϵ_3 denote the threshold. If $D_{r,j}$ is less than or equal to ϵ_3, the video from beginners is classified as representing the same sign as the corresponding conventional sign language. Conversely, if the distance $D_{r,j}$ exceeds ϵ_3, the input video is likely to represent a different sign from the corresponding conventional sign language. This threshold-based decision helps streamline the process by quickly identifying clear matches and minimizing computational overhead on more complex recognition tasks.

The Siamese LSTM, with its dual-network shared-weight design, can simultaneously process both conventional sign language videos and beginner imitation videos, calculating the distance between key point features to determine whether the same sign language action is conveyed. This similarity-based learning mechanism effectively reduces the complexity of repeatedly comparing conventional sign language actions, thereby improving learning efficiency. Compared to traditional classification models, the Siamese LSTM is more sensitive to action similarity, providing more targeted feedback to help beginners quickly identify and refine their movements.

Beginners receive real-time feedback on gesture similarity, highlighting positional deviations relative to the reference. This immediate feedback allows learners to make necessary corrections during the imitation process, ensuring more accurate and fluid sign execution.

Additionally, by using a dynamic threshold for similarity judgment, the model allows beginners to focus on imitating movements without the need to exactly match the conventional sign language, reducing learning difficulty and making the foundational skill-building phase more adaptable and flexible. This phase establishes a basic understanding of gesture similarity between learners and conventional sign language sequences, providing valuable feedback that flows into the next phase.

The transition from beginner to advanced learning is automated based on a proficiency score calculated from the Siamese LSTM's similarity metric. Learners who satisfy the following condition Eq. (35) for three consecutive sessions are advanced to the ConvLSTM phase, ensuring they are adequately prepared to handle more complex vocabulary and signing dynamics:

$$\prod_{s=t-2}^{t} \mathbb{I}\Big(\frac{1}{N}\sum_{i=1}^{N} \mathbb{I}\Big(D_{r,j}^{i,s} \leqq \epsilon_3\Big) \geq 0.9\Big) = 1, \tag{35}$$

where N represents the total number of gestures per session, $D_{r,j}^{i,s}$ represents the similarity distance for the i-th gesture in session s. $\mathbb{I}(\bullet)$ is an indicator function that equals 1 if the condition inside holds and 0 otherwise.

4.3 Advanced sign language recognition with ConvLSTM

This phase offers more sophisticated feedback to learners who already have a basic command of the movements, guiding them toward mastering subtle nuances in signing. The proposed SLTS utilizes an active learning approach for advanced learners using the ConvLSTM network to tackle more complex sign language vocabulary. By capturing both the spatial and temporal features of gestures, the ConvLSTM network enhances fluency and expressive abilities in real-world communication scenarios, enabling learners to master the intricate nuances of sign language.

Figure 4 illustrates the structure of the ConvLSTM. The ConvLSTM layer maintains a recurrent network architecture, where each ConvLSTM cell incorporates a memory cell along with three gates, including input, output, and forget gates, to model long-term dependencies effectively. The test video feature vector is represented as $\mathbb{X}_r^I \in \mathbb{R}^{\beta \times \omega}$, which consists of a sequence of β-length across spatial dimensions, denoted by $\{\big(\mathcal{X}_1^I, \ldots, \mathcal{X}_t^I, \ldots, \mathcal{X}_\beta^I\big) | \mathcal{X}_t^I \in \mathbb{R}^{1 \times \omega}, 1 \leq t \leq \beta\}$. Each element $\mathcal{X}_t^I \in \mathbb{X}_r^I$ denotes a specific frame's feature vector in the sequence. These elements are fed into the ConvLSTM cells in the first layer one by one to capture and integrate both the spatial features of individual frames and their temporal evolution across the sequence. This step-by-step processing allows the ConvLSTM to discern subtle gestures and complex motion patterns in the sign language videos, enhancing the model's ability to recognize subtle variations in sign language expressions accurately.

Let F_t denote the forget gate. The value of F_t can be calculated using the Eq. (36).

$$F_t = \sigma\big(W_{\mathcal{X}F} * \mathcal{X}_t^I + W_{HF} * H_{t-1} + b_F\big), \tag{36}$$

where σ is the sigmoid activation function, $W_{\mathcal{X}F}$ and W_{HF} denote the weights, b_F is the bias, and H_{t-1} denotes the hidden state at time step $(t-1)$. Let I_t denote the input gate. The value of I_t can be calculated using the Eq. (37).

$$I_t = \sigma\big(W_{\mathcal{X}I} * \mathcal{X}_t^I + W_{HI} * H_{t-1} + b_I\big), \tag{37}$$

where $W_{\mathcal{X}I}$ and W_{HI} denote the weights, and b_I is the bias. Let O_t denote output gate. The value of O_t can be calculated using the Eq. (38).

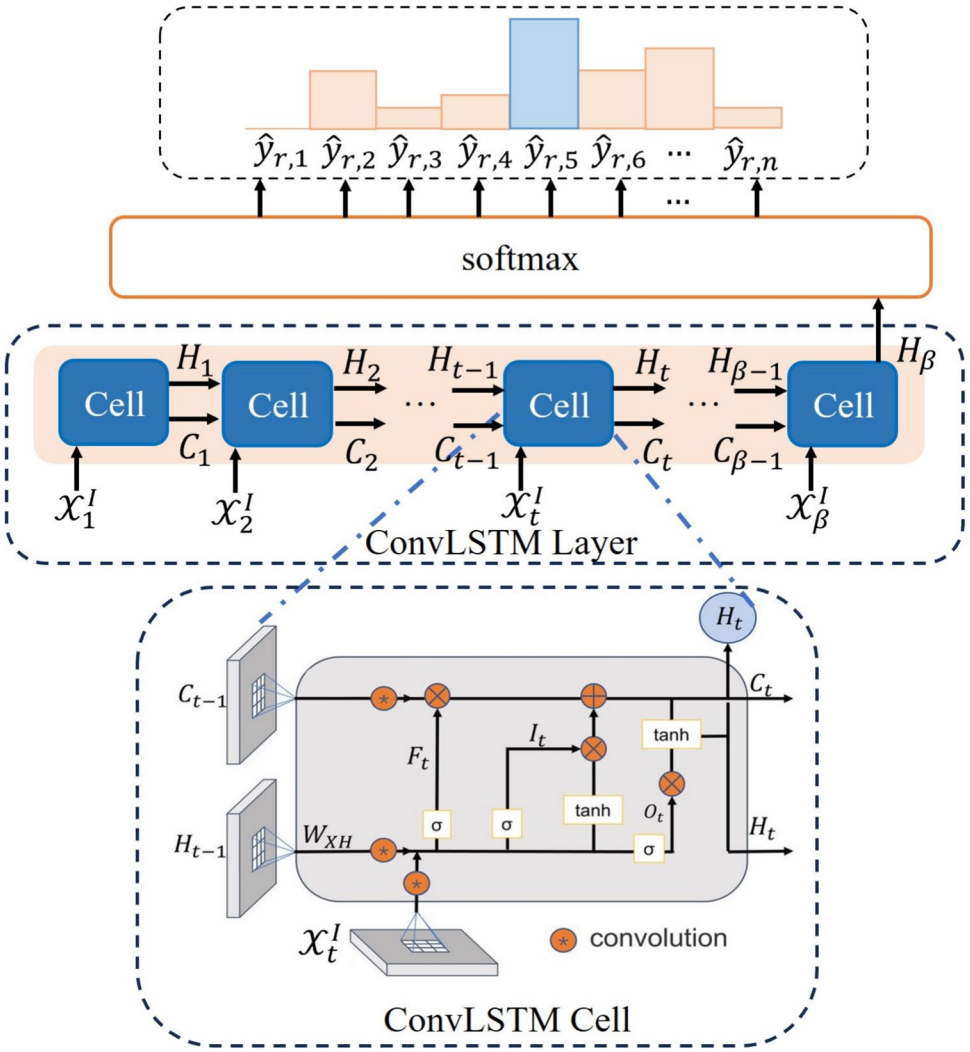


Fig. 4 The structure of the ConvLSTM

$$O_t = \sigma\left(W_{\mathcal{X}o} * \mathcal{X}_t^I + W_{Ho} * H_{t-1} + b_o\right), \tag{38}$$

where $W_{\mathcal{X}o}$ and W_{Ho} denote the weights, and b_o is the bias. Let C_t denote the memory cell. The value of C_t can be derived from Eq. (39).

$$C_t = F_t \odot C_{t-1} + I_t \odot \left(tanh\left(W_{\mathcal{X}g} * \mathcal{X}_t^I + W_{Hg} * H_{t-1} + b_g\right)\right), \tag{39}$$

where $W_{\mathcal{X}g}$ and W_{Hg} denote the weights, b_g denote the bias, *tanh* denotes the tan hyperbolic function and $\odot$ represents the Hadamard product.

Let H_t denote the output. H_t captures both spatial and temporal information at time slot t. It can be derived from Eq. (40).

$$H_t = O_t \odot tanh\left(C_t\right). \tag{40}$$

Then, the output of the ConvLSTM, denoted by H_β is fed into the softmax layer to convert it into a probability distribution. This distribution represents the likelihood of each class being the correct classification for the input $\mathbb{X}_r^I$.

Let $y_{r,j}$ denote an indicator variable, which indicates whether $\mathbb{X}_r^I$ belongs to category c_j. That is

$$y_{r,j} = \begin{cases} 1, if\ \mathbb{X}_r^I\ belongs\ to\ category\ c_j, \\ 0, Otherwise. \end{cases} \tag{41}$$

Let $\hat{y}_{r,j}$ denote the predicted probability by the model that $\mathbb{X}_r^I$ belongs to category c_j. This is the probability for each category as output by the SoftMax layer after processing the input.

Let L_C denote the loss function of the ConvLSTM model. The value of L_C can be calculated using Eq. (42).

$$L_C\left(y_{r,j}, \hat{y}_{r,j}\right) = -\sum_{r=1}^{\varepsilon}\sum_{j=1}^{n} y_{r,j}\log\left(\hat{y}_{r,j}\right), \tag{42}$$

where ε and n denote the number of samples in a batch and the number of categories, respectively. This formulation is crucial for training the ConvLSTM model efficiently, as it minimizes the discrepancy between the actual categories of the inputs and the predictions made by the model.

Through these models, the proposed SLTS mechanism can effectively process and recognize sign language videos. It utilizes the Siamese LSTM model to determine the similarity between the beginner's video and the conventional sign language video, followed by employing the ConvLSTM for advanced learners. Advanced learners receive granular feedback on spatial–temporal patterns, such as motion smoothness and transitional consistency, to refine expressive fluency. This targeted feedback helps learners develop more natural and fluid signing, bridging the gap between structured learning and real-world sign language communication. This processing approach significantly enhances the accuracy and robustness of sign language video recognition, providing tailored learning experiences for sign language learners at different levels.

4.4 Computational complexity analysis

The proposed SLTS consists of Siamese LSTM and ConvLSTM, both contributing to its overall computational complexity. The following provides a detailed analysis.

4.4.1 Siamese LSTM complexity

Each LSTM unit typically has a computational complexity of $O(d^2)$ per time step, where d represents the hidden state size. Given $\jmath$ key points per frame and β frames, the computational complexity should be $O(\beta \jmath d^2)$ for a single LSTM. Since Siamese LSTM consists of two parallel LSTM networks, the total complexity should be $O\left(2\beta \jmath d^2\right)$. That is

$$O\left(2\beta \jmath d^2\right) \approx O\left(\beta \jmath d^2\right), \tag{43}$$

4.4.2 ConvLSTM complexity

The ConvLSTM is responsible for capturing spatial–temporal features. At each time step, convolution operations are applied over a feature map of size $M \times N$ using a kernel size $K \times K$ with C input channels. The computational complexity of a single convolution operation is $O\left(K^2 CMNd\right)$. Since ConvLSTM operates over β frames, the total complexity of ConvLSTM is $O\left(\beta K^2 CMNd\right)$.

4.4.3 Overall complexity of SLTS

The total computational complexity of the SLTS is the sum of the complexities of Siamese LSTM and ConvLSTM:

$$O\left(\beta \jmath d^2\right) + O\left(\beta K^2 CMNd\right).$$

5 Performance evaluation

This section evaluates the performance improvement of the proposed SLTS against the existing related studies, especially the Temporal Attention Module (TAM) [8] and Multi-Layer Convolutional Neural Networks (ML-CNN) [9]. The TAM was designed for skeleton-based action recognition by selecting the most informative skeletons of an action at the early layers of the network. It further incorporated a lightweight graph convolutional network to reduce computational costs. However, TAM does not effectively extract spatial features, which limits its ability to capture the full complexity of sign language gestures. On the other hand, the ML-CNN focused solely on spatial features without considering temporal dynamics, which could be crucial for understanding the sequential nature of sign language. The proposed SLTS addresses these limitations by utilizing both Siamese LSTM and ConvLSTM architectures to comprehensively consider both temporal and spatial features. This dual approach significantly enhances the accuracy of sign language recognition by effectively capturing the complex patterns present in sign language gestures.

5.1 Parameter settings and dataset

Table 1 summarizes the parameter settings used in the experiments. The duration of each segment (β) is set to 90 frames, with a frame sampling interval (ζ) ranging from 1 to 10. To ensure optimal performance, the proposed SLTS was trained using the Adam optimizer with a learning rate of 1e-4 for 50 epochs, and a batch size of 32. To mitigate overfitting, early stopping with a patience of 10 epochs, and monitoring validation loss. The validation strategy followed a simple 80%-20% train-test split, ensuring a balanced evaluation of the model's generalization ability. In the Siamese LSTM model, the threshold controlling the distance between $\mathcal{H}_r^I$

Table 1 Parameter settings

Parameter	Value
Duration for each segment (β)	90 frames
Frame sampling interval (ζ)	1 ~ 10 frames
Batch size	32
Learning rate	1e-4
Threshold of distance	0.4 ~ 0.8
Validation strategy	80%–20% train-test split

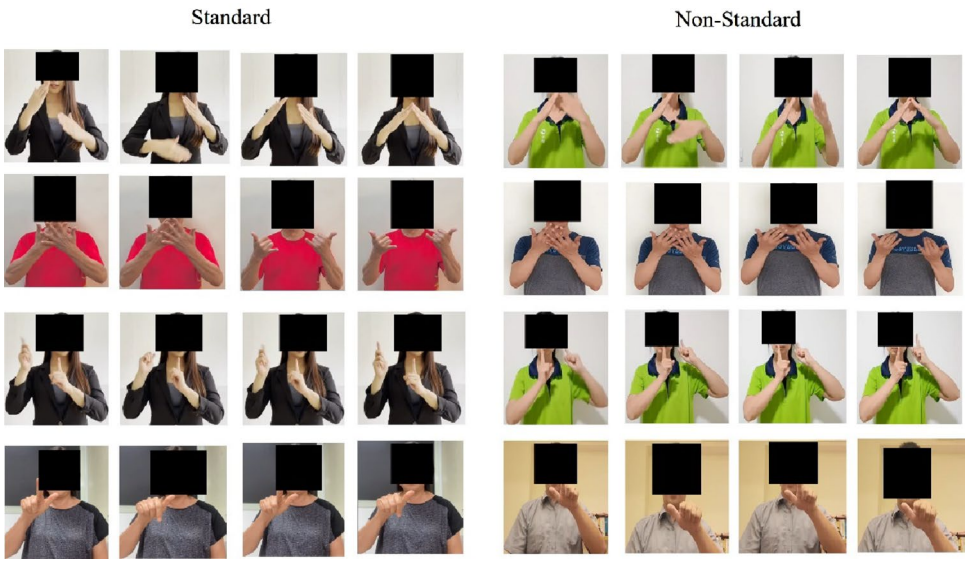


Fig. 5 Sample videos from the dataset

and $\mathcal{H}_j^S$ was adjusted within the range of 0.4 to 0.8, optimizing classification performance.

This paper leverages a publicly available dataset, PHOENIX-2014 [28], which consists of 190 sign language samples containing 965,940 frames and a total of 1558 words combined into 6861 consecutive utterances. Additionally, a custom-built dataset of Taiwanese Sign Language is employed, primarily sourced from the dataset provided by the Sign Language Translation Association of Taiwan.

Figure 5 showcases sample videos from this dataset. The dataset comprises 630 video clips, which contain various sign gestures, including 216 conventional sign language videos and 414 natural sign language videos. To ensure a standardized learning framework, conventional sign language was chosen as the primary focus of this study, as it provides a structured and reproducible reference for training and automatic scoring. In contrast, natural sign language, which varies in fluency and expressiveness among signers, was included to enhance model adaptability and robustness.

The dataset features a diverse group of 35 signers (18 male, 17 female) ranging in age from 15 to 50, with varying levels of signing proficiency. Among them, 22 are natural signers who acquired sign language from an early age, while 13 are conventional learners who learned sign language through formal education. To account for natural variations in signing styles, the dataset includes recordings of both formal classroom signing and real-world communication scenarios, capturing differences in signing speed, hand movements, and facial expressions. Videos were recorded in a controlled indoor environment with consistent lighting and a fixed camera setup to ensure high data quality and minimize external variability.

To ensure a robust evaluation, the dataset has been split into training and testing sets, with an allocation of 80% for training and 20% for testing purposes. This division ensures robust model training while allowing for an adequate evaluation of the model's performance on unseen data.

5.1.1 Simulation results

Figure 6 illustrates the relationship between training and testing of the proposed SLTS in terms of accuracy and average classification time across different sampling frame intervals. The sampling frame interval ranges from 1 to 10. The left y-axis represents accuracy, scaled from 0 to 1, while the right y-axis denotes the average classification time.

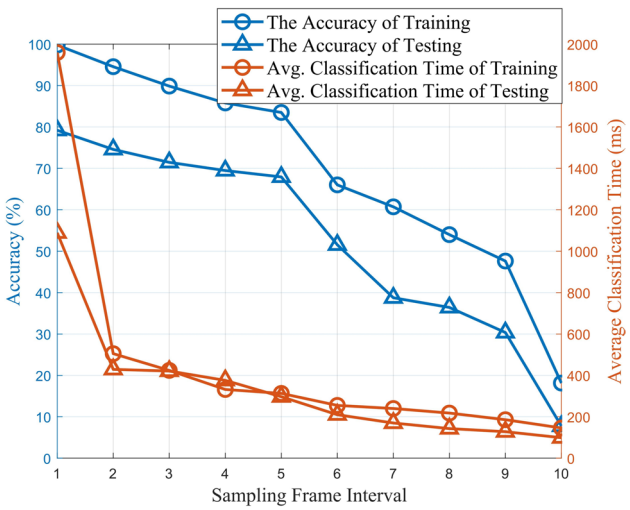


Fig. 6 Impact of sampling frame interval on accuracy and classification time

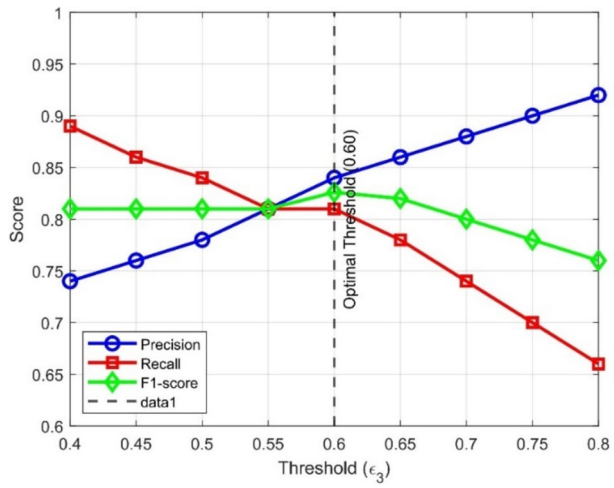


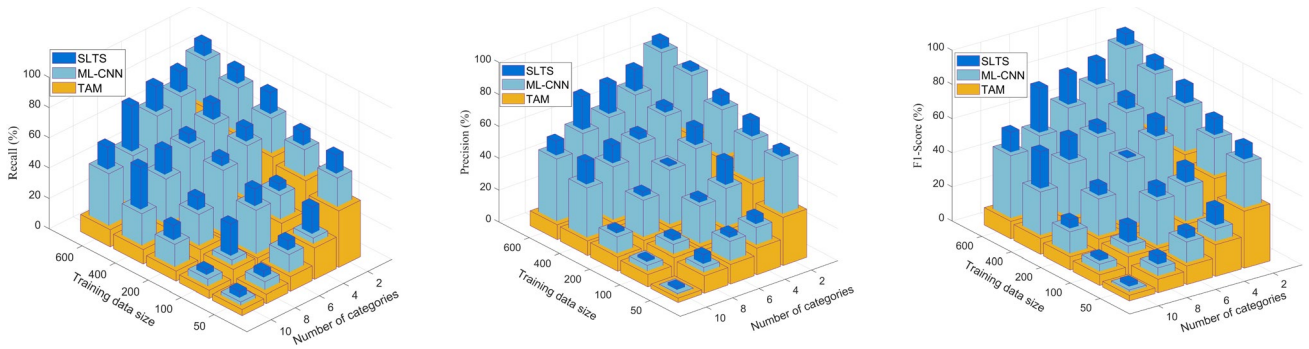
Fig. 7 Impact of threshold (ϵ_3) on precision, recall, and F1-score in Siamese LSTM

As shown in Fig. 6, it is evident that both training and test accuracy decrease with the sampling frame interval. The reason is that larger intervals result in less frequent video sampling, reducing the number of frames available to capture dynamic gestures in sign language. This leads to a loss of critical temporal information, making it harder for the model to accurately identify and differentiate complex sign movements, thus diminishing its predictive accuracy. The average classification time also decreases with the sampling frame interval. The reason is that fewer frames need to be processed as the interval increases. Consequently, the system can execute the classification task more quickly because it handles a smaller volume of data at each step.

As shown in Fig. 6, these trends illustrate a trade-off between accuracy and average classification time. Shorter intervals offer more data points and finer granularity, improving accuracy but increasing computational load. Conversely, longer intervals reduce computational demands but lead to significant drops in performance. These insights are crucial for optimizing sign language recognition systems and balancing resource constraints with the need for high accuracy.

Figure 7 evaluates the classification performance of the Siamese LSTM under different threshold values (ϵ_3) and determines the optimal threshold that balances Precision and Recall. The value of ϵ_3 varies from 0.4 to 0.8 in increments of 0.05. As the threshold increases, precision gradually improves, while Recall decreases. This is because a lower ϵ_3 allows more gestures to be matched, increasing Recall but also leading to a higher risk of misclassification, thereby reducing Precision. Conversely, a higher ϵ_3 accepts only highly similar gesture matches, improving Precision but potentially rejecting some correct gestures, resulting in lower Recall. To achieve an optimal balance between Precision and Recall, the proposed SLTS adopts the F1-score as the comprehensive evaluation metric. Experimental results indicate that when ϵ_3 is 0.6, the F1-score reaches its highest value (0.826), demonstrating that this threshold provides the best trade-off between Precision and Recall. Consequently, this study selects $\epsilon_3 = 0.6$ as the optimal threshold for Siamese LSTM, enhancing the accuracy and robustness of sign language gesture recognition for learners.

Figures 8a–c compare the proposed SLTS against the TAM and ML-CNN in terms of precision, recall, and F1-Score by varying training data size and number of categories, respectively. The training data size varies from 50 to 600, while the number of categories varies from 2 to 10. As shown in Fig. 8, all mechanisms have the common trend that the precision, recall as well as F1-score increase with the training data size. The reason is that larger datasets emphasize the importance of sufficient training data in developing effective machine learning models. On the other hand, when the number of categories increases, the recall, precision, and F1-Score tend to decrease for all models. The reason is that a larger number of categories adds complexity and increases the potential for confusion during classification, making it



(a) Recall. (b) Precision. (c) F1-Score.

Fig. 8 Comparison of the proposed SLTS, ML-CNN, and TAM in terms of precision, recall, and F1-Score

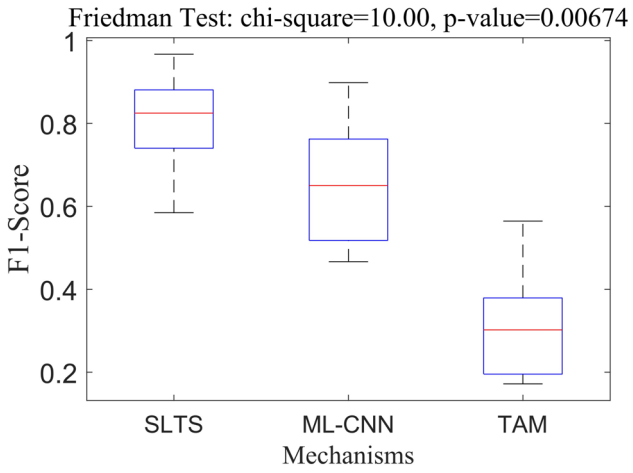


Fig. 9 The results of the Friedman test using different mechanisms

more challenging for the models to accurately distinguish between closely related categories.

As shown in Fig. 8, the proposed SLTS consistently outperforms the other two models across almost all scenarios. This indicates that the proposed SLTS has a more robust learning algorithm or perhaps leverages more sophisticated feature extraction techniques that enhance its sensitivity to relevant gestures.

To further validate the effectiveness of the proposed SLTS, the confidence interval (*CI*) uses 95% for the macro-averaged precision, recall, and F1-score. The *CI* is calculated as Eq. (44).

$$CI = \hat{y} \pm {}_{x}\sqrt{\frac{\hat{y}(1-\hat{y})}{n}}, \tag{44}$$

where $\hat{y}$ denotes the observed performance metric (e.g., precision, recall, or F1-score), n is the sample size, and ${}_{x}$ is the ${}_{x}$-score corresponding to the desired confidence level (1.96 for 95% confidence). These intervals provide insight into the statistical reliability and variability of the reported results.

Additionally, Fig. 9 presents the results of the Friedman Test to assess the statistical significance of performance differences among SLTS, ML-CNN, and TAM. The y-axis represents the F1-Score, while the *x*-axis lists the three compared methods. The Friedman test yields a chi-square value of 10 and a p-value of 0.00674, which is below the conventional significance threshold of 0.05, indicating statistically significant differences among the three mechanisms. As shown in Fig. 9, the proposed SLTS outperforms the other two mechanisms, demonstrating higher median and overall F1-Score, underscoring its enhanced capability in sign language identification.

To further support the error analysis, Fig. 10 depicts a normalized confusion matrix. The matrix reveals that

Fig. 10 The normalized confusion matrix

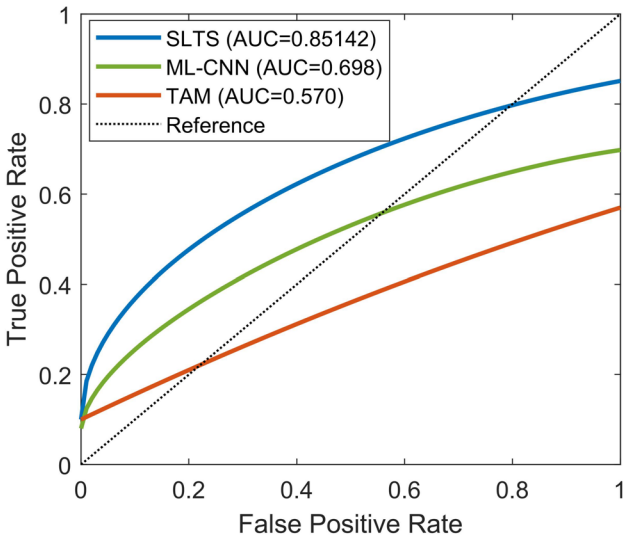


Fig. 11 ROC Curves for the proposed SLTS, ML-CNN, and TAM mechanisms

Table 2 The effectiveness of each component evaluated on the self-collected dataset

The proposed SLTS	Modules	F1-Score	t-value	p-value
Enhancing learning efficiency for beginners phase	LSTM	0.631	−3.872	0.0012
	Siamese LSTM	0.826		
Advanced sign language recognition phase	LSTM	0.651	−4.215	0.0008
	ConvLSTM	0.832		

misclassifications tend to occur between visually or structurally similar signs. For instance, class c1 is frequently misclassified as c7 and c9, suggesting a visual resemblance in hand posture or motion trajectory. Similarly, class c8 exhibits confusion with c4, c7, and c9, which may involve subtle finger movements or overlapping features. These patterns validate our earlier observation that errors predominantly arise among similar signs and indicate areas where incorporating temporal or spatial refinement could enhance classification accuracy.

Figure 11 illustrates the Receiver Operator Characteristic Curve (ROC) and Area Under the Curve (AUC) for the proposed SLTS, ML-CNN and TAM. The proposed SLTS exhibits a superior performance with an AUC value of 0.85142. The ROC curve ascends steeply towards the upper left corner, which suggests a high true positive rate at low false positive rates. The reason is that the proposed SLTS utilizes the Siamese LSTM, which computes the similarity between learners' gestures and reference signs, by capturing temporal characteristics such as motion smoothness, transition consistency, and execution speed. Unlike traditional classification models that rely solely on static frame comparisons, the proposed SLTS integrates both Siamese LSTM and ConvLSTM networks, leveraging sequential dependencies to assess how well a learner maintains the expected gesture trajectory over time. This combined approach enhances the model's ability to distinguish between correct and incorrect sign executions with greater accuracy.

Table 2 illustrates the effectiveness of each component in the proposed SLTS mechanism, evaluated on the self-collected dataset, comparing LSTM, Siamese LSTM, and ConvLSTM. All models used 80%-20% train-test split and trained using the Adam optimizer (learning rate is 1e−4), the batch size is 32, and a maximum of 50 epochs with early stopping (patience = 10). The models were evaluated based on Precision, Recall, and F1-score, with a focus on gesture classification accuracy. To provide a fair evaluation across all gesture classes, the proposed SLTS used the macro-F1 score, which calculates the F1-score for each class independently and then takes the average:

$$\mathrm{F1} = \frac{1}{n}\sum_{i=1}^{n}\frac{2 * \mathcal{P}_i * \mathcal{R}_i}{\mathcal{P}_i + \mathcal{R}_i}, \tag{45}$$

where n denotes the total number of gesture classes, $\mathcal{P}_i$ and $\mathcal{R}_i$ denote the precision and recall of the i-th gesture.

Table 3 Performance comparison on the PHOENIX-2014 dataset

Models	Backbone	Validation (%)		Test (%)		Inference time (ms)	Memory (MB)	FLOPs (G)
		D/*I*	WER	*D*/*I*	WER			
SLT [29]	Transformer	11.7/6.5	24.9	11.2/6.1	24.6	/	/	/
STTN [30]	Transformer	4.6/2.5	19.94	4.8/2.4	19.98	6642.84	6300	45.2
LSTM-only	Siamese LSTM	6.3/3	20.5	6.5/3.1	20.8	289	89	7.98
ConvLSTM-only	ConvLSTM	5.8/2/6	19.3	6/2.7	19.6	418	118	11.79
SLTS (ours)	Siamese LSTM + ConvLSTM	5/2	17.15	5.2/2.1	18.2	359	147	19.77

As shown in Table 2, in the beginner learning phase, the use of Siamese LSTM shows a substantial improvement in performance with an F1-score of 0.826, compared to 0.631 when using only LSTM, indicating enhanced learning efficiency for beginners. Similarly, in the advanced sign language recognition phase, ConvLSTM achieves an F1-score of 0.832, outperforming the standard LSTM with an F1-score of 0.651. These results demonstrate that the integration of Siamese LSTM and ConvLSTM effectively combines spatial and temporal information, leading to improved recognition accuracy. Statistical significance testing using paired t-tests confirms that these improvements are statistically significant, with t-values of -3.872 and -4.215, and p-values of 0.0012 and 0.0008, respectively. These results highlight the effectiveness of integrating Siamese LSTM and ConvLSTM into the proposed SLTS framework, as they effectively capture and combine spatial and temporal information, leading to enhanced recognition accuracy.

Table 3 presents a comparative analysis of the proposed SLTS, SLT [29], STTN [30], LSTM-only, and ConvLSTM-only models on the PHOENIX-2014 dataset. The evaluation includes deletion/insertion (D/I) errors, word error rate (WER), inference time, memory consumption, and computational cost measured in Floating-Point Operations Per Second (FLOPs). WER is computed using Eq. (46).

$$\text{WER} = \frac{S + D + I}{\mathbb{N}} \times 100\%, \tag{46}$$

where S denotes substitutions, D denotes deletions, I denotes insertions, and $\mathbb{N}$ is the total number of words in the reference text. As shown in Table 3, the proposed SLTS outperforms existing methods by achieving the lowest WER while maintaining efficient inference speed, significantly reducing the memory and computational burden compared to Transformer-based methods. This is attributed to the combination of Siamese LSTM and ConvLSTM, which not only retains LSTM's strength in modeling long-term dependencies but also enhances spatial feature learning through ConvLSTM. This hybrid approach maintains high recognition accuracy while striking a better balance in inference speed, making SLTS a promising candidate for real-time sign language recognition applications.

6 Conclusion

This paper proposed a novel sign language identification mechanism called SLTS, which effectively combines Siamese LSTM and ConvLSTM models. The proposed SLTS employs passive learning techniques by utilizing the Siamese LSTM for beginners, allowing them to compare their gestures with sign language actions to assess similarity. This approach reduces computational demands and aids beginners in developing foundational skills. On the other hand, it employs active learning strategies for advanced learners by leveraging ConvLSTM to capture both the spatial and temporal features of gestures. This enhances fluency and expressive abilities in real-world communication scenarios, enabling learners to master the intricate nuances of sign language. The effectiveness of SLTS has been demonstrated through the experiments, showcasing its ability to outperform existing models in terms of precision, recall, and F1-Score. Future work will focus on adapting the system for real-time applications, making it viable for use in live interactions and as a communication aid for the deaf and hard of hearing. This adaptation will require refining the model to achieve faster processing speeds while maintaining high accuracy, ensuring it can operate effectively in dynamic real-world environments.

Acknowledgements This work was supported in part by the Research Start-up Fund Project of Chuzhou University under Grant No. 2025qd12, Smart Home and Applied Industry Innovation Team, Higher Education Research Program Project under Grant No. 2022AH010067, Anhui Provincial Natural Science Foundation under Grant No. 2408085MF177, and Anhui Outstanding Youth Fund Project under Grant No. 2022AH030109.

Author contributions Qiaoyun Zhang: Conceptualization, Methodology, Formal analysis, Writing-original-draft, Experiment design. Chih-Yung Chang: Conceptualization, Methodology, Formal analysis, Writing-original-draft, Experiment design. Christopher Chuang: Writing-review-editing. Wen-Hwa Liao: Writing-review-editing. Diptendu Sinha Roy: Writing-review-editing, Supervision.

Data availability No datasets were generated or analysed during the current study.

Declarations

Conflict of interest The authors declare no competing interests.

References

1. Wang, Z., Zhao, T., et al.: Hear sign language: a real-time end-to-end sign language recognition system. IEEE Trans. Mob. Comput. **21**(7), 2398–2410 (2022)
2. Zhou, H., Zhou, W., et al.: Spatial-temporal multi-cue network for sign language recognition and translation. IEEE Trans. Multimedia **24**, 768–779 (2021)
3. Núñez-Marcos, A., Perez-de-Viñaspre, O., Labaka, G.: A survey on Sign Language machine translation. Expert Syst. Appl. **213**, 118993 (2023)
4. Adeyanju, I.A., Bello, O.O., Adegboye, M.A.: Machine learning methods for sign language recognition: a critical review and analysis. Intell. Syst. Appl. **12**, 200056 (2021)
5. Adaloglou, N., Chatzis, T., et al.: A comprehensive study on deep learning-based methods for sign language recognition. IEEE Trans. Multimedia **24**, 1750–1762 (2021)

6. Yirtici, T., Yurtkan, K.: Regional-CNN-based enhanced Turkish sign language recognition. SIViP **16**(5), 1305–1311 (2022)
7. Lee, M., Bae, J.: Real-time gesture recognition in the view of repeating characteristics of sign languages. IEEE Trans. Industr. Inf. **18**(12), 8818–8828 (2022)
8. Heidari, N., Iosifidis, A.: Temporal attention-augmented graph convolutional network for efficient skeleton-based human action recognition. In: 25th international conference on pattern recognition (ICPR), Milan, Italy, pp. 7907–7914 (2021).
9. Prasath, G.A., Annapurani, K.: Prediction of sign language recognition based on multi layered CNN. Multim. Tools Appl. **82**(19), 29649–29669 (2023)
10. Azadani, M.N., Azzedine, B.: Siamese temporal convolutional networks for driver identification using driver steering behavior analysis. IEEE Trans. Intell. Transp. Syst. **23**(10), 18076–18087 (2022)
11. Roy, D., Ishizaka, T., et al.: Detection of collision-prone vehicle behavior at intersections using siamese interaction LSTM. IEEE Trans. Intell. Transp. Syst. **23**(4), 3137–3147 (2020)
12. Gao, H., Kuenzel, S., Zhang, X.: A hybrid ConvLSTM-based anomaly detection approach for combating energy theft. IEEE Trans. Instrum. Meas. **71**, 2517110 (2022)
13. Zhang, L., Li, Y., et al.: Polarimetric HRRP recognition based on ConvLSTM with self-attention. IEEE Sens. J. **21**(6), 7884–7898 (2021)
14. Majd, M., Safabakhsh, R.: A motion-aware ConvLSTM network for action recognition. Appl. Intell. **49**, 2515–2521 (2019)
15. Talukdar, A.K., Bhuyan, M.K.: Vision-based continuous sign language spotting using Gaussian hidden Markov model. IEEE Sens. Lett. **6**(7), 1–4 (2022)
16. Xie, K., Zhang, Z., et al.: Efficient federated learning with spike neural networks for traffic sign recognition. IEEE Trans. Veh. Technol. **71**(9), 9980–9992 (2022)
17. Katoch, S., Singh, V., Tiwary, U.S.: Indian sign language recognition system using SURF with SVM and CNN. Array **14**, 100141 (2022)
18. Vu, H.N., Hoang, T., et al.: Sign language recognition with self-learning fusion model. IEEE Sens. J. **23**(22), 27828–27840 (2023)
19. Das, S., Imtiaz, M.S., et al.: A hybrid approach for Bangla sign language recognition using deep transfer learning model with random forest classifier. Expert Syst. Appl. **213**, 118914 (2023)
20. Wadhawan, A., Kumar, P.: Deep learning-based sign language recognition system for static signs. Neural Comput. Appl. **32**(12), 7957–7968 (2020)
21. Sharma, S., Gupta, R., Kumar, A.: Continuous sign language recognition using isolated signs data and deep transfer learning. J. Ambient. Intell. Humaniz. Comput. **14**, 1531–1542 (2023)
22. Zuo, R., Mak, B.: Improving continuous sign language recognition with consistency constraints and signer removal. ACM Trans. Multimed. Comput. Commun. Appl. **20**(6), 1–25 (2024)
23. Rastgoo, R., Kiani, K., Escalera, S.: Hand sign language recognition using multi-view hand skeleton. Expert Syst. Appl. **150**, 113336 (2020)
24. Bilge, Y.C., Cinbis, R.G., Ikizler-Cinbis, N.: Towards zero-shot sign language recognition. IEEE Trans. Pattern Anal. Mach. Intell. **45**(1), 1217–1232 (2023)
25. Abdullahi, S.B., et al.: Spatial–temporal feature-based end-to-end fourier network for 3D sign language recognition. Expert Syst. Appl. **248**, 123258 (2024)
26. Han, X., et al.: Sign language recognition based on R (2+1) D with spatial–temporal–channel attention. IEEE Trans. Hum.-Mach. Syst **52**(4), 687–698 (2022)
27. Pareek, G., Nigam, S., Singh, R.: Modeling transformer architecture with attention layer for human activity recognition. Neural Comput. Appl. **36**(10), 5515–5528 (2024)
28. Koller, O., Forster, J., Ney, H.: Continuous sign language recognition: Towards large vocabulary statistical recognition systems handling multiple signers. Comput. Vis. Image Underst. **141**, 108–125 (2015)
29. Camgoz, N. C., Koller, O., Hadfield, S., Bowden, R.: Sign language transformers: Joint end-to-end sign language recognition and translation. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 10020–10030, Seattle, WA, USA (2020).
30. Cui, Z., Zhang, W., Li, Z., Wang, Z.: Spatial–temporal transformer for end-to-end sign language recognition. Complex Intell. Syst. **9**(4), 4645–4656 (2023)