# A Real-Time Thai Sign Language Recognition System Using Skeletal Keypoints and LSTM Networks

Yosvaris Pisansawanya
*Artificial Intelligence & Internet of Things Research Laboratory*
*Faculty of Information Technology*
*Thai-Nichi Institute of Technology*
Bangkok, Thailand
2463110334@tni.ac.th; yosvaris@gmail.com

Pikul Vejjanugraha
*Software Engineering Program*
*College of Arts, Media and Technology*
*Chiang Mai University*
Chiang Mai, Thailand
pikul.v@cmu.ac.th

Pramuk Boonsieng
*Artificial Intelligence & Internet of Things Research Laboratory*
*Faculty of Information Technology*
*Thai-Nichi Institute of Technology*
Bangkok, Thailand
*Corresponding author: b.pramuk@tni.ac.th

***Abstract*—Communication barriers remain a significant challenge for individuals who rely on sign languages, particularly in regions where language-specific recognition systems are limited. Thai Sign Language (TSL) exhibits distinctive spatial and temporal characteristics that require specialized modeling approaches for accurate and efficient real-world deployment. This paper presents a lightweight Thai Sign Language recognition framework based on skeletal landmarks extracted using MediaPipe and modeled through Long Short-Term Memory (LSTM) networks. Instead of relying on high-dimensional raw images, the proposed approach employs a structured feature representation derived from hand and upper-body keypoints, explicitly encoding geometric relationships, finger states, and inter-hand interactions. This design improves discrimination of visually similar gestures while maintaining computational efficiency. An end-to-end pipeline is developed, including landmark detection, feature extraction, temporal modeling, training, evaluation, and real-time inference. Experimental results show that the proposed method achieves an F1-score of 0.980 and an overall accuracy of 97.7% under a signer-dependent evaluation setting. Comparative experiments demonstrate that the proposed representation significantly outperforms raw landmark inputs under identical model configurations. The system operates at approximately 7.31 FPS with 72 ms inference latency on a standard CPU, indicating near real-time performance. Although evaluation is limited to a signer-dependent setting, the results demonstrate a robust and efficient solution suitable for deployment on resource-constrained devices.**

***Keywords— thai sign language recognition, skeletal keypoints, long short-term memory (LSTM), real-time gesture recognition, assistive communication systems***

## I. INTRODUCTION

Effective communication is essential for social interaction. For individuals who are deaf or hard of hearing, sign language serves as a primary means of communication. However, limited public proficiency creates barriers in everyday interactions.

Recent advances in deep learning have improved sign language recognition (SLR), particularly for widely used languages such as American Sign Language [1]. However, many approaches rely on high-dimensional visual inputs and computationally intensive models, limiting real-time deployment on resource-constrained devices. In addition, these models cannot be directly applied to Thai Sign Language (TSL) due to linguistic and structural differences.

Thai Sign Language (TSL) presents additional challenges, including complex hand configurations, two-handed coordination, and dynamic temporal patterns. Existing studies have focused on isolated gestures or limited datasets, leaving a gap for lightweight systems capable of real-time operation.

To address these limitations, this paper proposes a near real-time TSL recognition system based on skeletal keypoints and Long Short-Term Memory (LSTM) networks. The framework uses MediaPipe to extract hand and upper-body landmarks and transforms them into structured spatial–temporal features, which are processed as fixed-length sequences by an LSTM model. This skeletal representation reduces input dimensionality and enables efficient inference on CPU-based hardware.

Unlike prior MediaPipe-LSTM approaches that rely on raw landmark sequences, this work introduces a structured representation that encodes geometric relationships, finger states, and inter-hand interactions. This structured design enhances discriminative capability, particularly for visually similar gestures, while maintaining computational efficiency for lightweight, real-time deployment.

The main contributions of this work are summarized as follows:

- A structured skeletal feature engineering framework that encodes geometric relationships, finger states, and inter-hand interactions beyond raw landmark inputs.
- A compact spatial–temporal representation (214 features per frame) that balances discriminative capability and computational efficiency.

- A lightweight LSTM-based temporal modeling approach optimized for real-time inference on CPU-based systems.
- An empirical evaluation demonstrating improved performance over baseline landmark-only representations.
- A practical real-time implementation achieving 97.7% accuracy with near real-time performance on standard hardware.

## II. Related Work

Sign language recognition (SLR) has evolved from static image-based approaches to temporally aware models that leverage skeletal representations and deep learning.

Early SLR systems relied on Convolutional Neural Networks (CNNs) for static image classification, achieving strong performance in finger-spelling and digit recognition tasks under controlled conditions [2]-[4]. However, these approaches process each frame independently and cannot effectively capture temporal dynamics in continuous gestures.

To address this limitation, recurrent architectures such as Long Short-Term Memory (LSTM) networks have been widely adopted for temporal modeling. By processing sequences of frames, LSTM-based models capture motion patterns and temporal dependencies between hand poses. Hybrid CNN–LSTM approaches further enhance performance by combining spatial feature extraction with temporal modeling [5]-[7].

More recent studies employ skeletal landmark representations extracted using frameworks such as MediaPipe. Instead of raw images, these methods utilize structured keypoint coordinates, reducing computational complexity and improving robustness to background variations [8], [9]. MediaPipe-based approaches combined with LSTM or BiLSTM have demonstrated strong recognition performance, with reported accuracy of approximately 0.83–0.91 on Thai Sign Language datasets [10]-[12]. Similar approaches have also been applied to other sign language datasets, demonstrating the effectiveness of landmark-based temporal modeling [13], [14]. In addition, recent studies on Thai finger spelling have highlighted the importance of dataset design and multimodal representations for improving recognition performance [15].

Despite these advancements, several challenges remain. Existing datasets are often limited in size and signer diversity, restricting model generalization. In addition, variations in lighting, viewpoint, and gesture execution can degrade real-time performance. While advanced architectures such as graph-based and transformer-based models have shown promising results, they typically require high computational resources, limiting their applicability to lightweight and real-time systems [16]-[19].

These limitations highlight the need for efficient skeletal-based approaches that balance recognition accuracy, computational efficiency, and real-time deployability for practical Thai Sign Language applications. However, most existing MediaPipe-based approaches rely on raw landmark sequences without explicitly modeling geometric relationships and inter-hand interactions, limiting their effectiveness for complex gesture discrimination.

## III. Methodology

This section describes the methodology used to develop the proposed Thai Sign Language (TSL) recognition system. An overview of the overall workflow is illustrated in Fig. 1.

### *A. Video Input and Data Acquisition*

The system uses short video sequences of isolated Thai Sign Language (TSL) gestures recorded under controlled indoor conditions with consistent lighting, background, and a fixed camera setup.

The dataset consists of 40 gesture classes (30 words and 10 numerical signs), including both one-hand and two-hand gestures. Data were collected from three signers, with approximately 30 samples per class, resulting in approximately 1,200 samples in total and a relatively balanced dataset.

For evaluation, a fixed split of 70% training, 15% validation, and 15% testing was applied. The evaluation follows a signer-dependent protocol, where samples from the same signers may appear across all subsets.

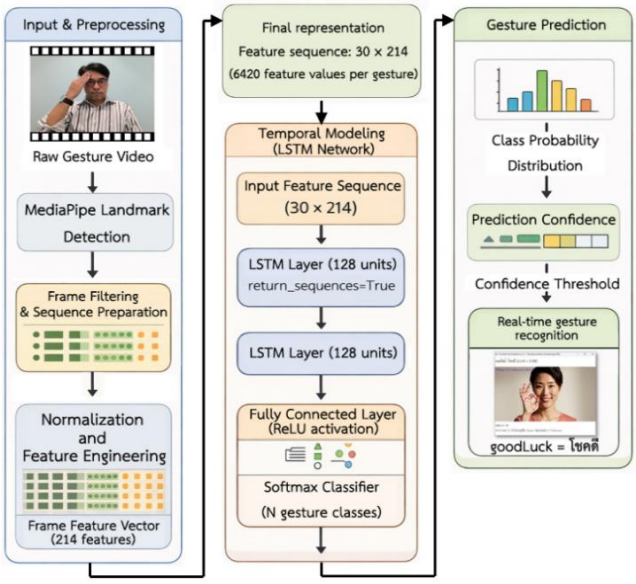


Fig. 1. Overall workflow of the proposed Thai Sign Language recognition system

Each gesture is segmented into short clips prior to feature extraction to ensure consistent temporal modeling.

### *B. Hand Landmark Detection and Data Preparation*

For each frame in the input video, hand landmarks are detected using a real-time landmark estimation framework [9], [20]. The detector identifies 21 anatomical keypoints per hand, corresponding to major finger joints and fingertips. These landmarks provide a compact skeletal representation of hand posture and motion.

During preprocessing, frames without detected hands are discarded to prevent invalid landmark inputs. The remaining frames are processed sequentially to construct stable landmark sequences for each gesture sample.

### C. Feature Construction and Normalization

The detected landmark coordinates are transformed into normalized skeletal representations to ensure spatial consistency across samples. Landmark coordinates are translated relative to the wrist joint and scaled using a reference bone length.

Let $\mathbf{x}_i$ denote the coordinate of the i-th landmark, and x_w represent the wrist landmark. Normalized coordinates are computed as:

$$\hat{\mathbf{x}}_i = \frac{\mathbf{x}_i - \mathbf{x}_w}{\| \mathbf{x}_{ref} - \mathbf{x}_w \|} \quad (1)$$

where $\mathbf{x}_{ref}$ is a reference joint used for scale normalization.

To ensure consistent temporal input for the LSTM model, each gesture sequence is standardized to 30 frames. Frames without detected hands are removed to avoid invalid inputs. The remaining sequence is then adjusted as follows: longer sequences are downsampled, shorter sequences are padded, while sequences already containing 30 frames remain unchanged.

The overall preprocessing and feature construction pipeline, including normalization and subsequent feature engineering, is illustrated in Fig. 2.

### D. Feature Engineering

To capture discriminative hand configurations and motion dynamics, a set of engineered skeletal features is derived from normalized landmark sequences. These features encode spatial relationships between joints, finger states, and inter-hand interactions.

Each frame is represented using a compact feature vector consisting of 214 features, including:

1) normalized joint coordinates,
2) geometric relationships (distances and joint angles),
3) finger state indicators,
4) hand configuration descriptors (e.g., openness and palm orientation), and
5) inter-hand spatial relationships for two-hand gestures.

Representative features include Euclidean distances and joint angles:

$$d_{ij} = \| \hat{\mathbf{x}}_i - \hat{\mathbf{x}}_j \| \quad (2)$$

and angular relationships between adjacent finger segments

$$\theta = \cos^{-1}\left(\frac{\mathbf{a} \cdot \mathbf{b}}{\| \mathbf{a} \| \| \mathbf{b} \|}\right) \quad (3)$$

Where $\mathbf{x}_i$ denotes landmark coordinates and $\mathbf{a}, \mathbf{b}$ represent adjacent joint vectors.

Each gesture is represented as a temporal feature sequence of size 30 × 214, which serves as input to the LSTM model (Fig. 2).

Unlike conventional landmark-based representations, the proposed feature design explicitly incorporates geometric and relational descriptors to enhance discriminative power for visually similar gestures.

### E. Temporal Modeling with LSTM

The engineered feature sequences are modeled using a Long Short-Term Memory (LSTM) network to capture temporal dependencies in gesture motion. In the proposed framework, the input to the temporal model is a feature sequence of size 30 × 214, representing 30 temporal frames with 214 engineered features per frame.

The baseline architecture consists of two stacked LSTM layers designed to capture hierarchical temporal patterns in gesture dynamics. The first LSTM layer contains 128 hidden units and returns the full sequence representation, while the second summarizes the temporal information into a fixed-length feature vector.

The resulting representation is passed through a fully connected dense layer with ReLU activation, followed by a softmax classifier that produces probability distributions over the gesture classes. The architecture of the proposed temporal model is illustrated in Fig. 3.

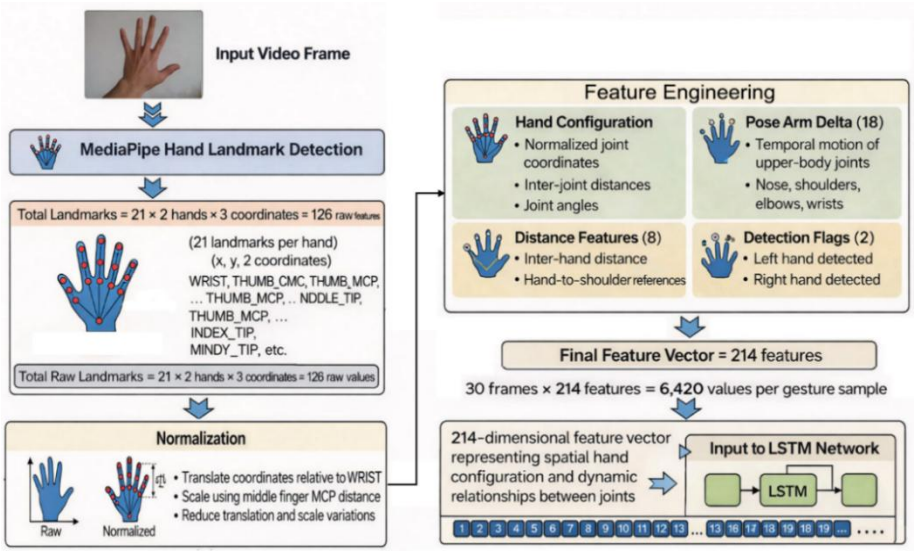


Fig. 2. Feature engineering pipeline for generating 214 features from MediaPipe landmarks

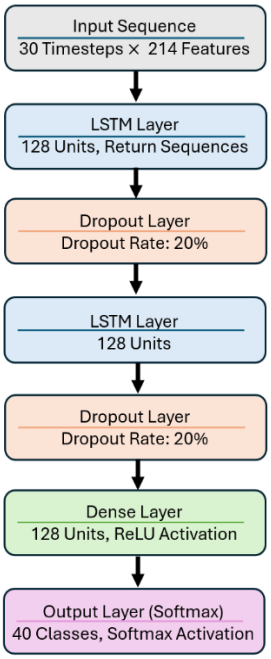


Fig. 3. Architecture of the proposed LSTM-based temporal model for Thai Sign Language recognition

### F. *Model Training and Evaluation*

The model is trained using categorical cross-entropy loss and the Adam optimizer. Dropout regularization is applied to reduce overfitting.

Training Configuration:

The model was trained using the Adam optimizer with learning rates of $1\times10^{-4}$ and $3\times10^{-4}$. The batch size was set to 32, and training was conducted for up to 50 epochs with early stopping to prevent overfitting. A dropout rate of 0.2 was applied within the network.

The dataset is divided into training, validation, and testing subsets using a 70% / 15% / 15% split. Model performance is evaluated using standard multi-class classification metrics including accuracy, precision, recall, and F1-score. Confusion matrix analysis is also used to examine misclassification patterns between visually similar gestures.

### G. *Prediction*

During inference, the trained model processes an input gesture sequence and produces class probabilities through the softmax layer. The class with the highest probability is selected as the predicted gesture.

## IV. Experimental Results

### A. *Model Configuration Comparison*

The proposed Thai Sign Language recognition system is evaluated using accuracy, precision, recall, and F1-score, as defined in Section III.

To identify an effective temporal modeling configuration, three LSTM architectures were evaluated with two learning rates, resulting in six configurations. The results are summarized in Table I.

TABLE I. Performance Comparison of LSTM Configurations

| LSTM Arch. | Learning Rate | Accuracy | Precision | Recall | F1-Score |
|---|---|---|---|---|---|
| 64 | $1\times10^{-4}$ | 0.903 | 0.909 | 0.904 | 0.906 |
| 64 | $3\times10^{-4}$ | 0.938 | 0.952 | 0.942 | 0.947 |
| 64-128 | $1\times10^{-4}$ | 0.943 | 0.935 | 0.945 | 0.940 |
| 64-128 | $3\times10^{-4}$ | 0.972 | 0.975 | 0.974 | 0.975 |
| 128-128 | $1\times10^{-4}$ | 0.955 | 0.965 | 0.955 | 0.960 |
| **128-128** | $3\times10^{-4}$ | **0.977** | **0.984** | **0.977** | **0.980** |

The results show that deeper LSTM architectures consistently outperform the single-layer model. The best configuration uses two stacked LSTM layers (128–128) with a learning rate of $3\times10^{-4}$, achieving an F1-score of 0.980. This configuration is selected for further analysis.

To evaluate the contribution of feature engineering, a baseline model using raw landmark inputs was implemented, where raw landmark inputs refer to the original (x, y, z) coordinates obtained from MediaPipe without normalization or additional feature processing. Both models were evaluated using the same LSTM architecture (128–128) and learning rate ($3\times10^{-4}$), differing only in input representation. The comparison results are presented in Table II.

TABLE II. COMPARISON OF FEATURE REPRESENTATIONS

| Feature Type | Learning Rate | Accuracy | Precision | Recall | F1-Score |
|---|---|---|---|---|---|
| Raw Landmarks | $3\times10^{-4}$ | 0.692 | 0.732 | 0.676 | 0.703 |
| **Proposed Features (214)** | $3\times10^{-4}$ | **0.977** | **0.984** | **0.977** | **0.980** |

The raw landmark baseline achieved an accuracy of 0.692, precision of 0.732, recall of 0.676, and F1-score of 0.703 on the external dataset. In contrast, the proposed feature representation achieved an accuracy of 0.977, precision of 0.984, recall of 0.977, and F1-score of 0.980.

Both models use identical model configurations to ensure a fair comparison, and performance differences are attributed solely to the input feature representation.

### B. *Training Behavior of the Selected Model*

To analyze the training behavior of the selected configuration, Fig. 4 illustrates the training and validation accuracy and loss curves of the best-performing model (128–128 hidden units with a learning rate of $3\times10^{-4}$).

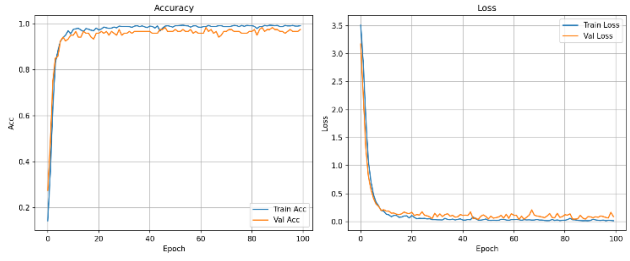


Fig. 4. Training and validation accuracy and loss curves of the best-performing LSTM model.

Both training and validation accuracy increase rapidly during early epochs and stabilize after approximately 20 epochs, indicating effective convergence. The validation curve closely follows the training curve, suggesting minimal overfitting.

### C. *Confusion Matrix Analysis*

To further examine class-wise recognition behavior, the confusion matrix of the best-performing configuration is presented in Fig. 5.

Strong diagonal dominance in the confusion matrix indicates that most gesture classes are correctly recognized, demonstrating the effectiveness of the proposed skeletal feature representation and temporal modeling approach.

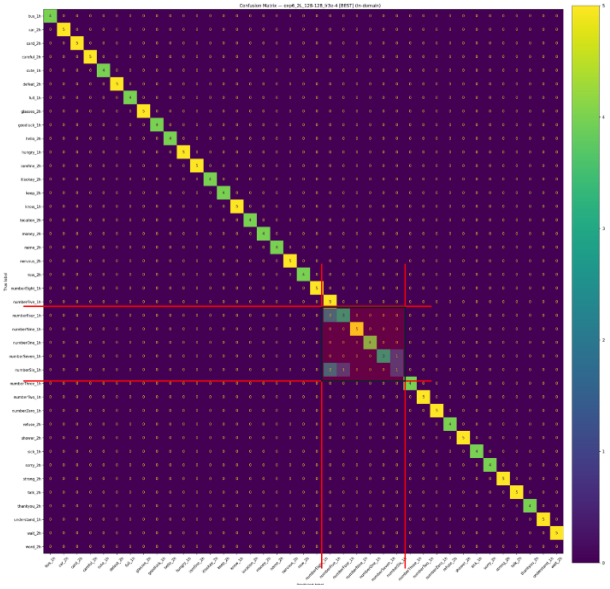

Fig. 5. Confusion matrix of the proposed Thai Sign Language recognition system

However, minor misclassifications are observed among visually similar gestures, particularly numerical signs such as six, seven, and eight. These errors can be attributed to subtle differences in finger configurations and overlapping geometric structures. Specifically, these gestures share similar spatial arrangements of extended fingers, resulting in highly similar skeletal representations.

In addition, variations in hand orientation, execution speed, and slight inconsistencies in finger articulation further increase intra-class variability and reduce inter-class separability. From a feature perspective, the overlap in distance-based and angle-based descriptors makes these gestures difficult to distinguish under certain conditions.

Despite these challenges, the overall impact on system performance remains limited, as reflected by the strong overall classification metrics. These findings suggest that while the proposed feature engineering approach is effective, further improvements could be achieved by incorporating more discriminative temporal or fine-grained finger motion features. This limitation highlights the challenge of distinguishing gestures with high structural similarity in skeletal-based representations.

### D. *Comparison with Related Work*

Table III compares the proposed approach with representative studies in sign language recognition. Although datasets and experimental settings vary across studies, the results indicate that the proposed skeletal feature engineering combined with LSTM-based temporal modeling achieves competitive performance.

The proposed system achieves an overall accuracy of 97.7% on the TSL dataset while maintaining lightweight computation suitable for near real-time or practical low-cost deployment. Direct comparison should be interpreted cautiously due to differences in datasets and experimental protocols.

TABLE III. COMPARISON WITH RELATED WORK

| Study | Method | Dataset | Accuracy |
|---|---|---|---|
| CNN-based TSL [2] | CNN | Finger spelling | 0.813 |
| MediaPipe + BiLSTM [11] | Landmark + BiLSTM | TSL gestures | 0.910 |
| CNN-LSTM SLRNet [5] | CNN + LSTM | Sign language dataset | 0.863 |
| **Proposed Method** | **MediaPipe + engineered features + LSTM** | **TSL dataset (40 gestures)** | **0.977** |

### E. *Discussion*

The results confirm that skeletal keypoint-based modeling provides an effective and computationally efficient solution for Thai Sign Language recognition, achieving strong performance with a compact representation suitable for CPU-based deployment.

The proposed feature engineering enhances discriminative capability beyond raw landmark inputs, while deeper LSTM architectures improve temporal modeling of gesture dynamics.

However, certain visually similar gestures, particularly numerical signs, remain difficult to distinguish due to overlapping geometric structures and variations in hand orientation and articulation. This reflects the inherent limitation of skeletal representations in capturing fine-grained finger differences.

Despite this, the overall impact on performance is minimal, as indicated by high classification metrics. Future improvements may focus on incorporating finer temporal cues or motion-aware features to better capture subtle gesture variations. These findings indicate that feature-level design plays a critical role in improving both accuracy and robustness in lightweight sign language recognition systems.

### F. *Real-Time Prototype Demonstration*

A real-time prototype was implemented using webcam-based video input to evaluate practical applicability.

The system captures video frames, extracts MediaPipe landmarks, constructs the 214-dimensional feature representation, and processes 30-frame sequences using the trained LSTM model.


Fig. 6. Real-time recognition interface of the proposed Thai Sign Language recognition system.

Fig. 6 shows the real-time recognition interface displaying the camera feed, detected landmarks, and predicted gesture with confidence score.

The prototype was evaluated on a standard laptop. As summarized in Table IV, the system achieved 7.3 FPS, 72 ms inference latency, 127 ms pipeline latency, and 22% CPU usage without GPU acceleration. These results demonstrate that the proposed framework can achieve near real-time performance on consumer-grade hardware, supporting practical assistive communication applications.

While the system demonstrates near real-time performance, the current frame rate may still be insufficient for certain interactive applications, and further optimization is required.

TABLE IV. REAL-TIME SYSTEM PERFORMANCE

| Metric | Value |
|---|---|
| Average Pipeline FPS | 7.31 frames/s |
| Average Inference Latency | 72 ms |
| Average Pipeline Latency | 127 ms |
| Average System CPU Usage | 22 % |
| Hardware Platform | Intel-based laptop CPU |

## V. CONCLUSION

This paper presented a Thai Sign Language (TSL) recognition system based on skeletal keypoints and Long Short-Term Memory (LSTM) networks. By leveraging structured feature representations instead of raw landmark sequences, the proposed approach effectively captures both spatial hand configurations and temporal gesture dynamics while maintaining low computational complexity.

Experimental results demonstrate that the proposed method achieves strong recognition performance, with an F1-score of 0.980 and an overall accuracy of 97.7% on the TSL dataset. Comparative experiments further confirm that the proposed structured feature representation significantly improves performance over raw landmark inputs under identical model configurations, highlighting the importance of feature-level design in skeletal-based recognition systems.

The system supports near real-time inference on standard CPU hardware, making it suitable for practical assistive communication applications. However, the evaluation remains limited to a signer-dependent setting with a relatively small number of participants, and visually similar gestures remain challenging due to subtle finger articulation and overlapping geometric structures.

Future work will focus on expanding the dataset and conducting signer-independent evaluations to improve generalization across users. Furthermore, incorporating more discriminative temporal and motion-aware features, as well as optimizing the inference pipeline, will be explored to enhance both recognition accuracy and real-time performance in more complex real-world environments.

## ACKNOWLEDGMENT

The author would like to express sincere gratitude to Dr. Pramuk Boonsieng from Thai-Nichi Institute of Technology for his valuable supervision and continuous support throughout this research. The author also thanks Dr. Pikul Vejjanugraha from the Software Engineering Program, College of Arts, Media and Technology, Chiang Mai University, for her valuable comments and constructive suggestions that helped improve this work.

## REFERENCES

[1] S. Tan et al., “A review of deep learning-based approaches to sign language processing,” Advanced Robotics, vol. 38, no. 23, pp. 1649–1667, 2024, doi: 10.1080/01691864.2024.2442721.

[2] W. Vijitkunsawat et al., “Video-based sign language digit recognition for the Thai language: A new dataset and method comparisons,” in Proc. Int. Conf. Pattern Recognition Applications and Methods (ICPRAM), 2023, pp. 775–782, doi: 10.5220/0011643700003411.

[3] T. Pariwat and P. Seresangtakul, “Multi-stroke Thai finger-spelling sign language recognition system with deep learning,” Symmetry, vol. 13, no. 2, p. 262, 2021, doi: 10.3390/sym13020262.

[4] P. Kittigul and S. Suvonvorn, “Thai sign language recognition: An application of deep neural network,” in Proc. ECTI DAMT & NCON, 2021, doi: 10.1109/ECTI-DAMT&NCON51128.2021.9425711.

[5] J. Huang, et al., “Video-based sign language recognition via residual network and long short-term memory,” Journal of Imaging, vol. 10, no. 6, p. 149, Jun. 2024, doi: 10.3390/jimaging10060149.

[6] W. Jintanachaiwat et al., “Using LSTM to translate Thai sign language to text in real time,” SN Computer Science, vol. 4, no. 17, 2024, doi: 10.1007/s44163-024-00113-8.

[7] S. K. Gupta and G. S. Walia, “Sign language recognition using LSTM and MediaPipe,” in Proc. Int. Conf. Intelligent Computing and Control Systems (ICICCS), 2023, doi: 10.1109/ICICCS56967.2023.10142638.

[8] S. Renjith et al., “An effective skeleton-based approach for multilingual sign language recognition,” Engineering Applications of Artificial Intelligence, vol. 129, Art. no. 109995, 2025, doi: 10.1016/j.engappai.2024.109995.

[9] Google, “MediaPipe solutions guide,” Google AI Edge. [Online]. Available: https://ai.google.dev/edge/mediapipe/solutions/guide.

[10] C. Damrongekarun et al., “Development of Thai sign language detection and conversion system into Thai with deep learning,” KKU Science Journal, vol. 51, no. 3, pp. 216–225, Sep. 2023, doi: 10.14456/kkuscij.2023.19.

[11] C. Guettas, “AI-driven sign language recognition system with NLP-enhanced transcription,” ECTI Transactions on Computer and Information Technology (ECTI-CIT), 2025, doi: 10.37936/ecti-cit.2025194.260940.

[12] N. Anithadevi and S. Palanisamy, “MediaPipe-LSTM-enhanced framework for real-time dynamic sign language recognition in inclusive communication systems,” Engineering Reports, vol. 7, no. 7, 2025, doi: 10.1002/eng2.70142.

[13] M. Z. Uddin et al., “Real-time Norwegian sign language recognition using MediaPipe and LSTM,” Multimodal Technologies and Interaction, vol. 9, no. 3, p. 23, 2025, doi: 10.3390/mti9030023.

[14] T. S. Varshini and P. Rukmani, “MP-GestLSTM: Real-time gesture detection using MediaPipe and LSTM,” Systems Science & Control Engineering, vol. 13, no. 1, Art. no. 2587853, Nov. 2025, doi: 10.1080/21642583.2025.2587853.

[15] J. Sanalohit, T. Katanyukul, et al., “A comprehensive benchmark and evaluation of Thai finger spelling in multi-modal deep learning models,” IEEE Access, 2024, doi: 10.1109/ACCESS.2024.10736595.

[16] Y. Han et al., “A study on the STGCN-LSTM sign language recognition model based on phonological features of sign language,” IEEE Access, 2025, doi: 10.1109/ACCESS.2025.3560779.

[17] S. Srisuwan et al., “Advancements in sign language recognition: A comprehensive review and future prospects,” IEEE Access, 2024, doi: 10.1109/ACCESS.2024.3464528.

[18] C. Luna-Jiménez et al., “Interpreting sign language recognition using transformers and MediaPipe landmarks,” in Proc. ACM Int. Conf. Multimodal Interaction (ICMI), 2023, pp. 373–377, doi: 10.1145/3577190.3614143.

[19] S. Yan et al., “Skeleton-based sign language recognition with part-mixing (SKIM),” IEEE Trans. Multimedia, 2024, doi: 10.1109/TMM.2024.10464382.

[20] Y. Fan, “The Gesture Recognition Improvement of MediaPipe Model Based on Historical Trajectory Assist Tracking, Kalman Filtering and Smooth Filtering,” in Proc. CISAI, 2024.