HNUE JOURNAL OF SCIENCE
Natural Sciences 2026, Volume 71, Issue 1, pp. 65-76
This paper is available online at http://hnuejs.edu.vn/ns
DOI: 10.18173/2354-1059.2026-0006

RECOGNIZING AND UNDERSTANDING AMERICAN SIGN LANGUAGE USING DEEP LEARNING AND MEDIAPIPE

Nguyen Tu Anh[1], Luong Thai Le[1,*], Tran Trung Hieu[1],
Nguyen Thi Lan Huong[2] and Nguyen Thien Minh[2]
[1]*Faculty of Information Technology, University of Transport and Communications, Hanoi, Vietnam*
[2]*Department of Information Technology - Communication, University of Science and Technology of Hanoi, Hanoi, Vietnam*
[*]Corresponding author: Luong Thai Le, e-mail: luongthaile80@utc.edu.vn

Received: November 19, 2025. Revised: February 1, 2026. Accepted: March 30, 2026.

Abstract. Sign language serves as the primary medium of expression serves as the primary medium of expression for deaf and hard-of-hearing individuals. However, interpersonal interaction remains challenging, as they primarily rely on sign language to express their thoughts. To address this issue, this study proposes an automated sign language recognition and interpretation framework, integrating static and dynamic recognition components. Specifically, a Convolutional Neural Network (CNN) is employed for static gesture classification, while a hybrid CNN- Long Short-Term Memory (CNN-LSTM) architecture is utilized to capture the spatiotemporal features of dynamic signs. Furthermore, MediaPipe is leveraged for robust landmark localization to enhance feature extraction. The American Sign Language (ASL) dataset used in this research ensures diversity in sign representation, including variations in hand shapes, positions, and movements. The proposed models achieved high accuracy, with the CNN model reaching 93.1% and the CNN-LSTM model achieving 94.1% on test datasets, confirming their effectiveness in ASL recognition tasks.

Keywords: American Sign Language, convolutional neural network, recurrent neural network, long short-term memory, MediaPipe.

1. Introduction

The advancement of artificial intelligence and computer vision technologies has significantly accelerated the development of applications designed to facilitate human communication, particularly for individuals with hearing and speech impairments. American Sign Language (ASL) serves as the primary medium of expression for the deaf community in the United States and several other countries. However, due to the limited

number of ASL-proficient individuals, deaf individuals often face severe linguistic barriers in daily life, education, healthcare, and employment [1], [2].

Recent global research has explored deep learning-based methods for ASL recognition [3], [4], leveraging convolutional neural networks (CNNs) for visual feature extraction and recurrent neural networks (RNNs), especially Long Short-Term Memory (LSTM) networks, for modeling gesture sequences. Despite promising outcomes, challenges remain, including variability in signing styles, environmental noise, and the scarcity of large-scale annotated ASL dataset for training robust models [5], [6].

Research on ASL recognition remains in its infancy [7]. Existing approaches primarily focus on classifying static ASL characters using basic CNN architectures [8]. Only a limited number of studies [9] have explored hybrid CNN-LSTM frameworks capable of recognizing dynamic gestures representing full words. This direction, while showing significant potential in global research, remains under-explored within the local academic context.

This study focuses on the design and implementation of an ASL recognition system utilizing a hybrid architecture that combines CNN and LSTM models. Specifically, the system processes spatial hand landmarks (comprising 21 three-dimensional coordinates) extracted via the MediaPipe framework [10] to recognize both static signs (letters and digits) and dynamic signs (words). A key feature of this study is the integration of temporal modeling through LSTM layers to capture sequential gesture patterns – a technique that enhances the system's ability to interpret word-level signs in real-time.

The main objectives of this study are as follows: (1) To develop a CNN-based model for the recognition of individual ASL letters and digits; (2) To implement a hybrid CNN-LSTM architecture for integrating dynamic ASL word gestures; (3) To design and evaluate an integrated software prototype with a user-friendly interface to support real-time interaction for ASL users.

This study focuses on the application of CNN and CNN-LSTM models to MediaPipe-derived 3D landmark representations for American Sign Language recognition. Rather than proposing novel network architectures, the contribution lies in demonstrating the effectiveness of this landmark-based methodology for integrating both static and dynamic gestures.

By focusing exclusively on American Sign Language and employing state-of-the-art (SOTA) deep learning techniques, this research presents a practical and scalable solution to ASL recognition, with potential applications in assistive technologies for the deaf and hard-of-hearing.

2. Content

2.1. Problem statement

The task of sign language integration involves processing input images or videos that capture hand gestures and facial expressions of users communicating in ASL. The objective is to transcribe these gestures into meaningful text or speech. The problem can be mathematically modeled as follows:

Input: A data pair (x_i, c_i) for the i-th sample, where

$\boldsymbol{x_i} \in \boldsymbol{R^{21\times 3}}$*, denotes the feature vector representing the spatial coordinates of 21 reference landmarks (each consisting of x, y, z coordinates).*

$\boldsymbol{c_i} \in \{\mathbf{0}, \mathbf{1}\}^{\boldsymbol{K}}$*, is a one-hot encoded vector representing the ground-truth class label, spanning a category set of letters, digits, or words across K classes.*

Output: A predicted vector $s_i \in \{0,1\}^K$*, representing the posterior probability distribution over the K class labels given the input* x_i

Proposed methodology

The proposed solution consists of the following core stages:

- *Frame Acquisition and Processing*: Each frame from the webcam is analyzed to detect key landmarks on the hands, face, and upper body using the MediaPipe framework.

- *Feature Extraction*: The system extracts the 3D landmark coordinates, which represent the spatial configuration of gestures.

- *Data Preprocessing*: The extracted coordinates are normalized to ensure consistency and compatibility with the deep learning models. This stage includes coordinate transformation, feature scaling, and temporal alignment.

- *Sign Recognition*:

+ Static gestures: For individual letters and digits, CNNs are utilized to classify each frame based on its spatial characteristics.

+ Dynamic gestures: For full words – requiring the interpretation of sequential movements, the system employs a hybrid CNN-LSTM architecture. In this configuration, CNN layers extract frame-level spatial features, while LSTM networks model the temporal dependencies across frames to perform sequence-level classification.

- *Translation and Output*: The recognized gesture labels are translated into natural language, rendered as either a textual display or speech synthesis.

2.2. Proposed methods

2.2.1. Feature extraction with MediaPipe

Instead of using raw image input, the system leverages MediaPipe to extract 21 three-dimensional (3D) landmarks from the hand [11]. Each point is represented by (*x, y, z*) coordinates. This approach mitigates noise, simplifies data representation, and focuses on hand geometry – the core element of sign language recognition.

The extracted feature vectors are normalized and scored per frame for input into the deep learning models.

Before being fed into the neural networks, the extracted landmark coordinates are normalized to minimize variations caused by spatial position and scale. Normalization is performed independently on each frame to ensure consistent representation across different samples and recording conditions. This preprocessing step enhances model stability and generalization without introducing additional computational overhead.

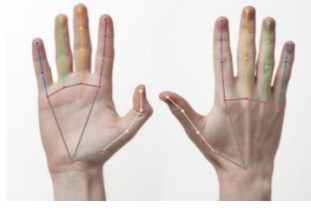

Figure 1. Landmark extraction using MediaPipe

2.2.2. CNN for static sign recognition

In this study, the CNN architecture for static sign recognition is designed to process three-dimensional (3D) spatial coordinates (*x, y, z*) of 21 hand landmarks extracted from each input frame [12]. These landmark coordinates serve as a compact and feature-rich representation of the hand's spatial configuration and are used as direct input to the model.

The network comprises a series of three convolutional layers. The first convolutional layer applies 32 filters, each with a kernel size of 3×3, followed by a Rectified Linear Unit (ReLU) activation function. This layer produces a feature map of size $21\times3\times32$. The second convolutional layer utilizes 64 filters, also with a kernel 3×3, and outputs a feature map of size $11\times2\times64$, again followed by ReLU. The third convolutional layer increases the depth to 128 filters, continuing with the ReLU activation and yielding a final feature map of dimensions $6\times1\times128$.

After each convolutional layer, a max pooling operation with a 2×2 A window is applied to reduce the spatial dimensions while retaining the most salient features. These pooling operations contribute to lowering computational complexity and mitigating overfitting.

Following the convolutional and pooling stages, the three-dimensional feature map is flattened into a one-dimensional (1D) vector to prepare it for fully connected (dense) layers. The dense layer that follows consists of 128 units and employs the ReLU activation function. This layer captures high-level abstract features through matrix multiplications with trainable weights, transforming the input into a format suitable for final classification.

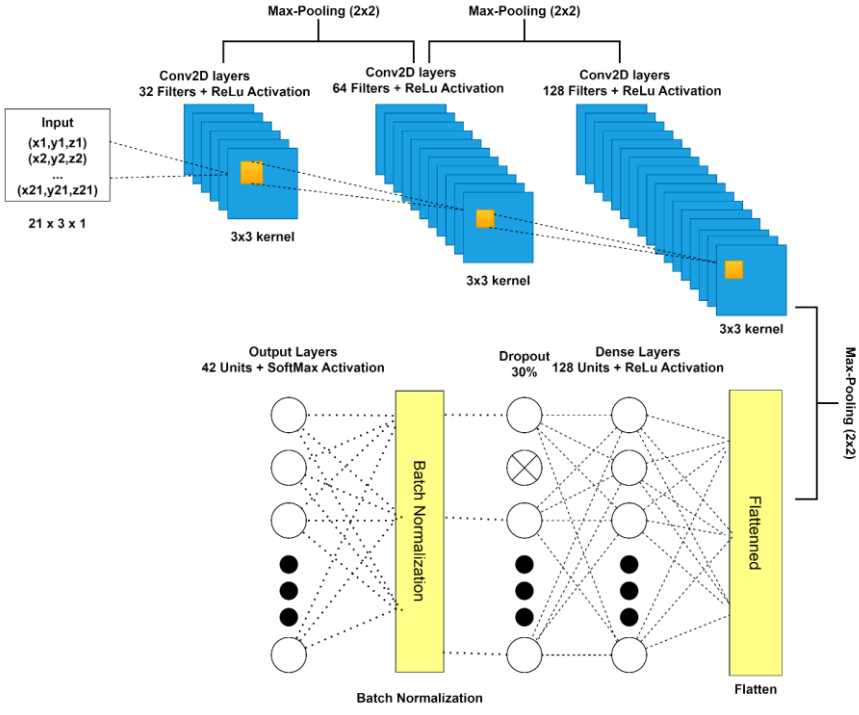

Figure 2. The convolutional neural network model used for recognizing letters and digits

To further improve generalization and model robustness, a dropout layer is applied with a dropout rate of 30%. This technique randomly deactivates a fraction of neurons during training, effectively mitigating the risk of overfitting. In addition, batch normalization (BN) is employed after certain layers to normalize the activations and gradients, which accelerates convergence and stabilizes the learning process.

The final output layer is a fully connected dense layer with 42 units, corresponding to the 42 static ASL classes (letters and digits) included in the dataset. A softmax activation function is applied to this output to produce a probability distribution over all classes, with the highest probability class selected as the predicted sign.

This architecture, combining spatial feature extraction, dimensionality reduction, and regularization techniques, provides an effective and robust solution for real-time static sign recognition using hand landmark data.

2.2.3. CNN-LSTM model for word-level sign language recognition

To recognize dynamic word-level gestures in ASL, a hybrid CNN- LSTM networks is proposed. This model is specifically engineered to capture both spatial features from individual frames and temporal dependencies across the entire gesture sequence.

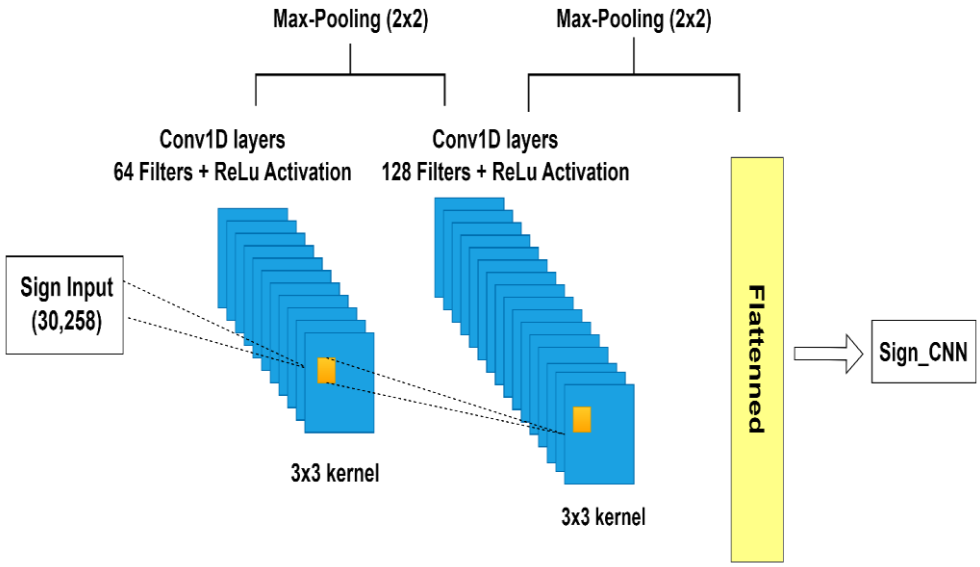

Figure 3. Convolutional layer architecture

The CNN component is responsible for extracting spatial features from the input sequences of 3D hand landmark coordinates. The configuration is as follows:

- *1D convolutional layers*: The first 1D-CNN layer consists of 64 filters with a kernel size of 3, using the ReLU activation function. The second 1D-CNN layer consists of 128 filters, also with a kernel size of 3 and ReLU activation.

- *1D max pooling layers***:** After each convolutional layer, a 1D max pooling operation is applied to reduce the feature map dimensions while retaining the most salient spatial features.

- *Flatten layer:* The resulting feature map is passed through a flattening layer, converting it into a 1D feature vector to be used for the subsequent LSTM layers.

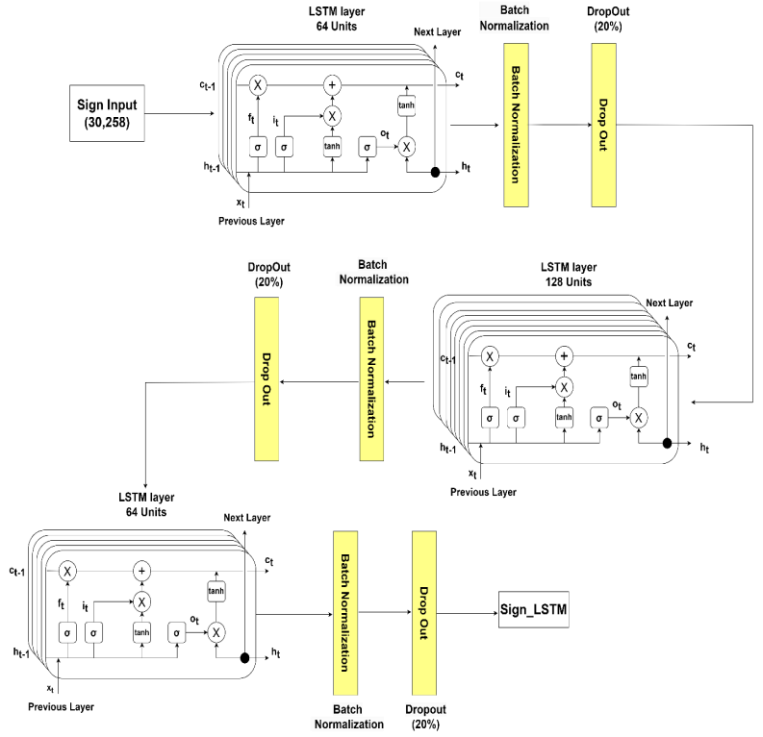

Figure 4. LSTM layer architecture

The LSTM component is utilized to model the temporal sequence of gestures [13]. The sub-network includes:

- *Three Stacked LSTM layers*: The architecture features three sequential LSTM layers with 64, 128, and 64 units, respectively. These layers are designed to capture long-term dependencies within the gesture data.

- *Dropout (20%):* After each LSTM layer, a dropout layer with a rate of 20% is applied to mitigate overfitting by randomly deactivating a fraction of neurons during the training phase.

- *Batch Normalization (BN):* This is employed to normalize the activations from the previous layer, thereby enhancing convergence and model stability.

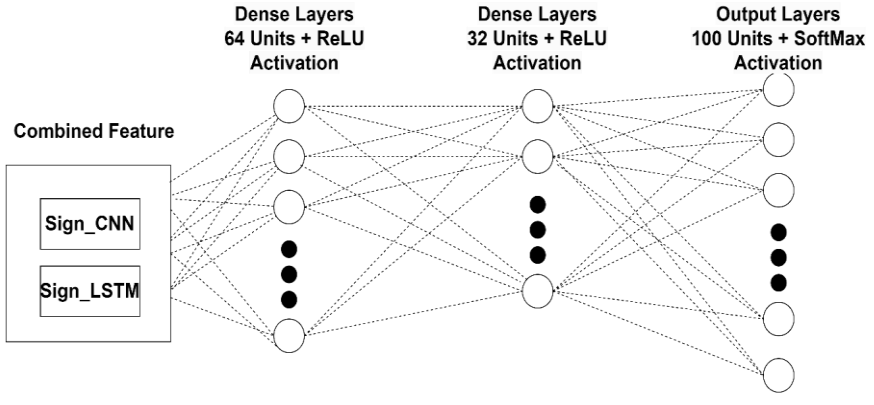

Figure 5. Dense layers architecture

The final classification is handled by a series of dense layers that process the integrated feature from the CNN and LSTM components. The first dense layer contains 64 units with ReLU activation. The second dense layer consists of 32 units, also employing ReLU activation. The final output layer comprises 100 units, corresponding to 100 dynamic ASL word classes, and utilizes the Softmax activation function to produce a probability distribution over the target labels.

This hybrid architecture enables the model to effectively learn spatial patterns within individual frames and temporal dynamics across engine gesture sequences, enabling high-accuracy recognition of word-level ASL gestures.

2.3. Experiments and results

2.3.1. Experimental data

The dataset utilized in this study was self-collected and organized into two distinct subsets: a static gesture set comprising 42 labels (ASL letters and digits) and a dynamic gesture set containing 100 labels representing individual ASL vocabulary words.

The data was acquired by four participants using high-dimension (HD) cameras with a minimum resolution of 1080p. Video recordings were conducted under controlled, stable lighting conditions to ensure image quality. A standardized recording protocol was adopted, with the camera positioned perpendicular to the signer. This setup ensured a consistent viewpoint across samples, facilitating robust and reliable landmark extraction via MediaPipe.

For the static dataset, each class was recorded with 400 samples, resulting in a total of 16800 samples. Each sample comprises 21 hand landmarks, where each landmark is defined by its three-dimensional (3D) coordinates (x, y, z). Data acquisition was performed using a webcam, where the hand region was first detected and localized using Google's MediaPipe framework. Subsequently, each gesture representing letters and digits was captured, and the resulting landmark vectors were serialized into NumPy arrays, systematically organized by respective class labels.

For the dynamic vocabulary dataset, 100 distinct ASL words were selected, with 100 samples collected for each word, resulting in a total of 10,000 sequences. Each sample consists of a sequence of 30 frames, where each frame includes 258 landmarks representing both hand and body posture. The acquisition process utilized the MediaPipe holistic framework to track multi-modal landmarks. Gestures corresponding to words such as "Hello" and "Goodbye" were performed in front of the webcam, captured as 30-frame sequences, and serialized as NumPy arrays within class-specific directories.

In this study, no explicit spatial or temporal data augmentation techniques were implemented. The extracted landmark features were utilized directly for model training and evaluation following normalization. Augmentation methods such as random cropping, temporal jittering, occlusion simulation, or illumination variation were omitted, as the primary focus of this work is to evaluate the baseline performance of landmark-based representations under strictly-controlled experimental conditions. This standardized setup also ensures the reproducibility of the experimental results.

This dataset serves as the primary foundation for training and evaluating the CNN and CNN-LSTM models within the proposed sign language recognition framework.

2.3.2. Experimental design

The experimental process for both static and dynamic ASL gesture recognition tasks comprises three main phases: data processing, model construction, and training.

For the static sign recognition model (CNN-based), raw data folders were loaded, and corrupted or invalid samples were removed. The resulting dataset of 16800 samples was split into training (60%), validation (20%), and testing (20%) sets, with all landmark coordinates normalized. The dataset was partitioned using random sampling at the sample level. Notably, the current experimental setting does not enforce cross-subject separation, meaning that samples from the same subject may appear in different splits. While this strategy is suitable for evaluating overall recognition performance, subject-independent evaluation remains an important direction for future work.

To improve model generalization, techniques such as 5-fold cross-validation, early stopping, and dropout regularization were applied. In addition, we experimented with Support Vector Machines (SVMs) as a baseline to compare with the CNN for static sign recognition. We utilized the Gridsearch method to select the optimal hyperparameter set for the SVMs, ultimately selecting C = 0,1 and the RFB (Radial Basis Function) kernel. After performing a 5-fold cross-validation on the SVMs, the highest F1 score obtained was 88.4%.

For the word-level recognition task using the CNN-LSTM model, a similar data processing pipeline was followed. The 10000 video-based gesture sequences were validated and split into training, validation, and test sets. The training process also incorporated cross-validation, early stopping, and batch normalization to reduce overfitting and improve learning stability. For this case, we also experimented with a baseline LSTM model. The results demonstrate that using CNN for feature extraction significantly enhances the LSTM model's recognition performance in this context.

2.3.3. Results and discussion

The developed CNN model for ASL character and digit recognition demonstrated strong performance across both validation and test datasets. The average validation accuracy achieved via 5-Fold cross-validation was 0.9314**,** while the test accuracy reached 0.9309**,** indicating a high level of generalization. The model also yielded a low test loss of 0.2177**,** suggesting effective error minimization throughout training. Furthermore, the model attained a precision of 0.9342, a recall of 0.9309, and an F1-score of 0.9307, reflecting its balanced ability to correctly classify relevant instances while minimizing false positives and false negatives. These metrics collectively confirm the model's robustness and reliability in static ASL recognition tasks, rendering it suitable for real-world applications where both accuracy and consistency are critical.

For dynamic word recognition, the CNN-LSTM model produced similarly impressive results. With an average validation accuracy of 0.9142 across 5-Fold splits, the model maintained consistent performance across various data subsets. The test accuracy was recorded at 0.9420, demonstrating the model's exceptional generalization capacity to generalize to unseen sequences. Additionally, the model achieved a test loss of 0.3158**,** underscoring its ability to minimize prediction errors on complex sequential inputs. The precision, recall, and F1-score for the CNN-LSTM model were 0.9526**,** 0.9420, and 0.9414, respectively. These metrics emphasize the model's efficacy in

accurately identifying relevant samples and producing highly reliable predictions. Such performance highlights the model's suitability for real-time sequence classification tasks, particularly in scenarios involving complex spatiotemporal patterns, such as ASL word recognition.

The superior accuracy of the dynamic ASL model relative to its static counterpart can be attributed to two primary factors. First, although we maintained a fixed camera angle, the involvement of four different signers introduced inherent variations. The disparity in accuracy between the two datasets stems from individual signing styles and slight inconsistencies in hand formations, both across subjects and during repeated sessions by the same individual. Second, while dynamic data presents inherent complexities, the model effectively decomposes these into sequential representations. By leveraging Long Short-Term Memory (LSTM) networks - an architecture specifically optimized for temporal modeling - the system captures critical motion dynamics that are absent in static frameworks.

The table below presents the F1-scores of the proposed methods in comparison with the baseline models. In this context, Static ASL refers to the task of recognizing letters and digits, whereas Dynamic ASL involves the recognition and interpretation of full words in American Sign Language.

Table 1. F1-score of the models

Static ASL	SVMs	88.4%
	CNN	***93.1%***
Dynamic ASL	LSTM	89.7%
	CNN-LSTM	***94.1%***

To further analyze class-level behavior, we incorporated confusion matrix analysis into the evaluation. Notable misclassifications were observed between visually and temporally similar sign pairs, such as THANK YOU and GOOD, as well as WHERE and WHAT. These signs share comparable hand trajectories and subtle variations in motion direction, which pose challenges for classification, especially within landmark-based representations.

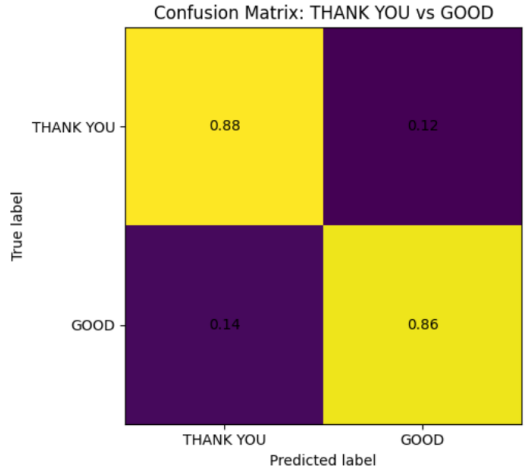

Figure 6. Confusion matrix for THANK YOU vs GOOD

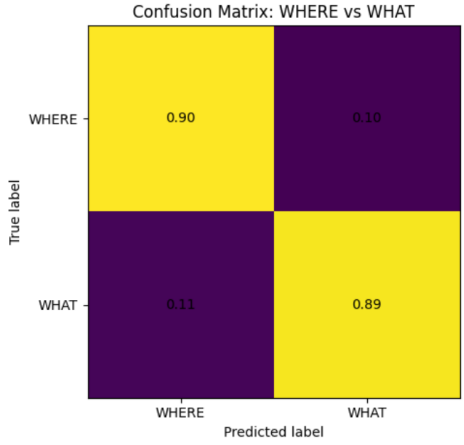

Figure 7. Confusion matrix for WHERE vs WHAT

To evaluate the real-time applicability of the proposed system, runtime performance metrics were analyzed. The system operates at an average rate of 30 frames per second (FPS). The average inference latency is approximately 17 ms (0.017 seconds) for letter and digit recognition, and 310ms (0.31 seconds) for word-level recognition using the CNN–LSTM model. The model size for static character prediction is 1,771 KB, while the word-level recognition model occupies 21.47GB (21,985 KB). Experiments were conducted on a workstation equipped with a 1080p camera, 16 GB RAM, an Intel Core i5 processor, and a GPU with at least 4 GB of VRAM. These results demonstrate that the proposed landmark-based approach is highly efficient and suitable for real-time sign language recognition applications.

To assess robustness under practical conditions, we further analyzed the sensitivity of the proposed system to lighting degradation and partial occlusion. When the illumination level was reduced by approximately 50% compared to the standard recording condition, the recognition performance decreased by 20.3% in terms of F1-score, primarily due to reduced landmark detection reliability. In occlusion scenarios, where parts of the hand or fingertips were partially obstructed, performance degradation was approximately 17.6%, depending on the occlusion severity and duration of the occlusion. For word-level recognition, the CNN–LSTM model exhibited superior robustness to short-term occlusions, leveraging temporal information across frames to compensate for missing data. These observations underscore the impact of real-world environmental factors on system performance and provide a rationale for future improvements through advanced data augmentation and robustness-oriented training strategies.

Compared to existing sign language recognition approaches that rely on raw images or video streams, the proposed method adopts a landmark-based representation, which significantly reduces input dimensionality and computational complexity while preserving essential spatiotemporal information. This design choice enables the model to focus on intrinsic motion patterns and hand configurations, mitigating the impact of appearance variations, such as lighting, skin tone, or background clutter. Although a direct quantitative comparison with other methods was not conducted due to dataset discrepancies, this qualitative comparison highlights the distinctive attributes and potential advantages of the proposed approach.

Overall, both models demonstrated robust and competitive performance within the scope of the conducted experiments, with the CNN-LSTM model exhibiting superior capability in handling temporally structured sequences. The results validate the feasibility of employing a landmark-based deep learning architecture for effective sign language recognition under controlled experimental settings.

This study focuses on evaluating the proposed landmark-based recognition pipeline on a self-curated dataset. Due to discrepancies in data acquisition protocols, class definitions, and feature representations (specifically landmark-based features versus raw pixels), a direct quantitative comparison with results reported on public ASL or hand-gesture datasets was not conducted. Nevertheless, the model design is firmly grounded in commonly adopted architectures within the existing literature. Benchmarking against public datasets and state-of-the-art architectures remains an important direction for future work to further assess the generalization and competitive standing of the proposed approach.

3. Conclusions

This study proposed an ASL recognition system leveraging 3D hand and body landmarks extracted via the MediaPipe framework, incorporating two deep learning architectures: a CNN for static gesture recognition and a CNN–LSTM for dynamic word recognition. Both models achieved outstanding performance, with test accuracies of 93.09% for static signs and 94.20% for dynamic gestures, underscoring their robust generalization and reliability in practical sign language interpretation.

In real-world testing, the CNN model demonstrated high-speed and precise predictions for both hands, while the CNN–LSTM model exhibited robust performance under optimal lighting. However, suboptimal illumination adversely affected recognition accuracy, primarily due to failures in landmark detection.

Future work will focus on enhancing model robustness by expanding the dataset, incorporating a more diverse demographic of users, and implementing advanced architectures. These enhancements aim to bolster performance and adaptability in unconstrained environments, thereby contributing to the development of sophisticated assistive communication technologies.

REFERENCES

[1] Zambian Ministry of Health & Hapunda R, (2024). Addressing the rising prevalence of hearing loss. *World Health Organization.*

[2] Alyami S, Luqman H & Hammoudeh M, (2024). Reviewing 25 years of continuous sign language recognition research: Advances, challenges, and prospects. *Information Processing & Management,* 61, 103774. DOI: 10.1016/j.ipm.2024.103774.

[3] Bantupalli K, Xie Y, (2018). American Sign Language Recognition using Deep Learning and Computer Vision. *IEEE International Conference on Big Data (Big Data)*. IEEE, Seattle, WA, USA, 2018, 4896-4899. DOI: 10.1109/BigData.2018.8622141.

[4] Paul SK, Walid MAA, Paul RR, Uddin MJ, Rana MS, Devnath MK, Dipu IR & Haque MM, (2024). An Adam based CNN and LSTM approach for sign language recognition in real time for deaf people. *Bulletin of Electrical Engineering and Informatics*, 13(1), 499-509. DOI: 10.11591/eei.v13i1.6059.

[5] Ahmed MA, Zaidan BB, Zaidan AA, Salih MM & Lakulu MMB, (2018). A review on systems-based sensory gloves for sign language recognition: State of the art between 2007 and 2017. *Sensors.* Basel, Switzerland, 18(7), 2208. DOI: 10.3390/e20110809.

[6] Zhang T & Xie L, (2016). Continuous sign language recognition based on 3D hand and body pose data. *Proceedings of the IEEE International Conference on Computer Vision (ICCV),* 4606-4614.

[7] Li Y & Zhao M, (2015). A study of continuous sign language recognition using joint feature fusion. *Proceedings of the IEEE International Conference on Computer Vision (ICCV),* 4241-4249.

[8] Ahmed K, Ahmed EAE, Omar A & Arif Y, (2022). DeepASLR: A CNN based human computer interface for American Sign Language recognition for hearing-impaired individuals. *Computer Methods and Programs in Biomedicine Update* (Vol. 2). DOI: 10.1016/j.cmpbup.2021.100048.

[9] Gupta A, Sawan A, Singh S & Kumari S, (2024). Dynamic Sign Language Recognition with Hybrid CNN-LSTM and 1D Convolutional Layers. *2024 11th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO)*. India, 1-6. DOI: 10.1109/ICRITO61523.2024.10522339.

[10] Sign All Engineering Team, (2021). SignAll SDK: Sign language interface using Mediapipe is now available for developers. *Google Developers Blog.* https://developers.googleblog.com/en/signall-sdk-sign-language-interface-usingmediapipe-is-now-available-for-developers/

[11] Lugaresi C, Tang J, Nash H, McClanahan C, Uboweja E, Hays M, Zhang F, Chang CL, Yong MG, Lee J, Chang WT, Hua W, Georg M & Grundmann M, (2019). MediaPipe: A Framework for Building Perception Pipelines. arXiv preprint arXiv: 1906.08172.

[12] Simonyan K & Zisserman A, (2014). Very deep Convolutional Networks for Large-Scale Image Recognition. arXiv preprint arXiv: 1409.1556.

[13] Hefron RG, Borghetti BJ, Christensen JC & Kabban CM, (2017). Deep long short-term memory structures model temporal dependencies improving cognitive workload estimation. *Pattern Recognition Letters*, 94, 96-104. DOI: 10.1016/j.patrec.2017.05.020.