Journal of Innovation and Community Engagement (ICE) **e-ISSN: 2776-0421**
Vol. 07, No. 02, (May, 2026) **https://doi.org/10.28932/ice.v7i2.12818**

Implementation of Hand Movement Tracking for Real-Time Sign Language Translation

Bernardus Anggodho Aryudhawan Hadi, Wilfridus Bambang Triadi Handaya*, Suyoto
Faculty of Industrial Technology, Universitas Atma Jaya, Yogyakarta, 55281, Indonesia

*Correspondence should be addressed to Wilfridus Bambang Triadi Handaya;
wilfridus.handaya@uajy.ac.id

(Received July 29, 2025; Revised May 9, 2026; Accepted May 10, 2026)

Abstract

This study developed a real-time American Sign Language (ASL) sign language identification and interpretation system based on deep learning. The system used two data sources: the ASL alphabet dataset for individual character recognition and the WLASL dataset for vocabulary recognition. The WLASL dataset was chosen as the benchmark for evaluating complex word gestures because it encompasses a wide range of users and extensive movement dynamics. Data processing involved extracting hand-gesture and body-posture markers using MediaPipe, followed by preprocessing and augmentation. Two learning architectures were implemented: a Feedforward Neural Network for alphabet classification and a BiLSTM integrated with an Attention Mechanism for vocabulary recognition. The system was evaluated using accuracy, precision, recall, F1 Score, and K-fold cross-validation. The results demonstrated promising performance: 99% accuracy for alphabet recognition and 78% for vocabulary recognition, with the Attention Mechanism contributing substantially to vocabulary recognition. The system operates in real time at 15-20 FPS and is efficient on mid-range devices, potentially becoming an inclusive communication alternative for the sign language community.

Keywords: *alphabet, attention mechanism, BiLSTM, deep learning, mediapipe*

How to Cite:
Hadi, B. A. A., Handaya, W. B. T., & Suyoto, S. (2026). Implementation of hand movement tracking for real-time sign language translation. *Journal of Innovation and Community Engagement, 7*(2), 173-187. https://doi.org/10.28932/ice.v7i2.12818

Introduction

Sign language is the primary means of communication for deaf people to convey information and express themselves. This research aims to build a real-time system capable of recognizing ASL letters and word gestures using a regular webcam and to evaluate its performance in terms of accuracy, stability, and computational efficiency. This study represents the development of an initial prototype intended to support future collaboration with deaf communities. Future work will involve collecting usability feedback through collaboration with sign language users and accessibility practitioners. Real-time sign language translation systems enable more effective communication across sectors such as education, healthcare, public services, and workplaces, as demonstrated by Alsharif et al. (2025). Abdullah et al. (2024) discuss key issues in building a sign language recognition system. These include model accuracy, dynamic gesture processing, and real-time applications (Abdullah et al., 2024).

This study emphasizes the importance of technology that supports communication for people with disabilities. As discussed by Al-Qurishi et al. (2021), deep learning techniques are one of the most widely used methods (Al-Qurishi et al., 2021). They highlight the efficacy of LSTM models in handling time-series data and the importance of dataset diversity for training the model as a whole. Gu et al. (2022) used deep learning to recognize ASL alphabets using an inertial motion capture system accurately (Gu et al., 2022). Sundar and Bagyammal (2022) strengthened the landmark-based approach, which showed potential for exemplary performance in gesture classification (Sundar & Bagyammal, 2022).

Abdulhamied et al. (2023) demonstrated the ability of LSTM to recognize gestures directly by developing a real-time LSTM-based ASL system (Abdulhamied et al., 2023). From a visual perspective, Zhang et al. (2024) suggested the use of a dual-path CNN with background removal to reduce visual noise from the background (Zhang et al., 2024). Zhang et al. (2022) developed a practical FFNN approach named FNNS for large-scale data processing in gesture classification (Zhang et al., 2022). Kazbekova et al. (2025) also proposed a hybrid Deep CNN-BiLSTM system with Attention, which combines the power of spatial and temporal feature extraction in gesture recognition on edge devices (Kazbekova et al. 2025). This study focuses on the implementation of a lightweight real-time sign language recognition system. Combines MediaPipe-based landmark extraction with FNN and BiLSTM-Attention architectures for both static and dynamic gesture recognition using a standard webcam.

MediaPipe has been used in conjunction with deep learning models in several studies. Bora et al. (2023) used MediaPipe and deep learning to recognize Assamese gestures in real-time (Bora et al., 2023). Samaan et al. (2022) integrated MediaPipe landmark output with an RNN to recognize dynamic gestures. This method demonstrated the effectiveness of a pipeline that relies on visual features (Samaan et al., 2022). Orovwode et al. (2023) emphasized that preprocessing and feature engineering are crucial for machine learning-based systems relevant to real-world applications (Orovwode et al., 2023). The system aims to achieve computational efficiency and real-time usability on mid-range devices while maintaining stable recognition performance under various lighting conditions.

Methods

The objective of this research is to develop a real-time letter and word recognition system in American Sign Language (ASL) using deep learning methods. MediaPipe, developed by Lugaresi et al. (2019), is crucial for gesture research because it can quickly and easily detect landmark points. The main steps in the methodology include dataset preparation, feature extraction, data preprocessing and augmentation, model training, real-time system integration, and system performance evaluation.

Dataset Preparation

The study utilized two types of datasets corresponding to the two categories of recognized gestures: (1) the ASL Alphabet dataset, made publicly available through Kaggle, which contains static hand images representing the letters A–Z, including additional classes such as “space” and “delete,” developed by Akash Nagaraj (2018); and (2) the WLASL (Word-Level ASL) dataset developed by Li et al. (2020), which contains short video clips of ten word-level gesture classes consisting of “Book,” “Happy,” “Hello,” “Please,” “Thank you,” “Why,” “Where,” “Who,” “Yes,” and “No.” Each video clip is approximately 2–3 seconds in duration. The use of these two datasets enables the system to recognize both static gestures (letters) and dynamic gestures (words).

Landmark Extraction

For letter gestures, MediaPipe Hands was used to extract 21 hand landmark points. MediaPipe Hands will generate 63 features (21 landmark points × 3 x, y, z coordinates). Furthermore, for word gestures, MediaPipe Holistic (a combination of MediaPipe Hands and MediaPipe Pose)

was used to extract 33 body poses, 21 for the left hand, and 21 for the right hand. Each landmark has four attributes (x, y, z, visibility), resulting in a total of 75 points × 4, or 300 features per frame. Each video was converted into a 3D sequence (30 frames, 300 features). The extraction results were saved in .json format for the letter dataset and in .csv format for the word dataset for processing efficiency.

Preprocessing and Augmentation

In the preprocessing stage, all landmark coordinates were normalized to the range [0, 1] against the maximum value per sample to reduce dependence on body size and camera position. To maintain consistent model input dimensions, the sequence length for word data was standardized to thirty frames per video. Next, all data was converted into an array or a tensor format. The letter dataset was transformed into a (n_samples, 63) NumPy array, and the word dataset was converted into a (batch_size, 30, 300) PyTorch tensor. Using CrossEntropyLoss, the letter data labels were encoded using one-hot encoding, while the word labels were encoded as integers according to their class index.

Data augmentation is used to improve data generalization and variation. To improve generalization and reduce overfitting, data augmentation is applied in the form of adding Gaussian noise, temporal shifting, and random masking of some landmark coordinates during training. For letter data, Gaussian noise with a standard deviation of 0.01 is randomly added to the landmark coordinates. Augmentation for word data is employed in the form of temporal shifting (reordering the frame), masking (omitting a small portion of features), and adding Gaussian noise. These three methods are applied probabilistically during training and have been shown to improve validation performance by 3–5%. To address the imbalance in the amount of data between classes, oversampling is performed on the word data, especially for gestures such as "Why," which have significantly less data than other classes.

Model Training

The learning system employs two distinct neural network architectures tailored to the characteristics of the data. Alphabet classification utilizes a Feedforward Neural Network architecture with three hidden layers and a ReLU activation function. The optimization process uses the Adam algorithm with an initial learning rate of 0.0005 over 30 training iterations. Vocabulary identification employs a double-layered, bidirectional, long-short-term memory structure integrated with the attention system. This system enables the network to focus on

each critical frame in a sequence of gestures differently. The vocabulary model was optimized using the Adam algorithm with a batch size of 16 and the CosineAnnealingLR scheduler for 50 learning cycles.

The alphabet model was implemented using the TensorFlow framework, while the vocabulary model was developed using the PyTorch platform. An initial comparison study evaluated the FNN and BiLSTM-Attention models against alternative architectures, including Convolutional Neural Networks and Transformers. In terms of model techniques in various studies, such as those conducted by Sherstinsky (2020), the structure of RNN, LSTM, and BiLSTM has been studied, each of which can process temporal information effectively. A BiLSTM can learn bidirectional context (forward and backward) within the data sequence, making it suitable for dynamic gestures. The results showed that the key point marker-based approach offered superior stability and efficiency for real-time implementation.

System Implementation

The final stage is the real-time implementation of the system using a laptop webcam. The system captures videos and extracts landmarks directly using MediaPipe. The data is then normalized and processed through the model to perform inference directly. To improve prediction stability, smoothing techniques such as the Exponential Moving Average and majority voting are applied to sequence frames. If the prediction confidence exceeds 60%, the results are displayed live. For letter models, the prediction results are also sent as automatic input to the typing system using the PyAutoGUI library, including the space and delete functions. The system was tested under various lighting conditions and hand positions, with tests showing system performance in the range of 15–20 FPS for word recognition, indicating sufficient efficiency for real-time applications.

Results and Discussions

This study produced two main models that were tested for two types of gestures: letter recognition (static) using FNN and word recognition (dynamic) using BiLSTM with and without an Attention Mechanism. Table 1 shows that the training results indicate the FNN model achieves very high performance on ASL letter data, with accuracy, precision, recall, and F1-score each reaching 99%. This indicates that the model can recognize static hand gestures effectively.

Table 1. FNN model classification results

	Precision	Recall	F1-Score	Support
a	1.00	0.99	0.99	2058
b	1.00	1.00	1.00	2111
c	0.99	0.99	0.99	1840
d	1.00	1.00	1.00	2074
delete	0.98	1.00	0.99	1441
e	1.00	0.99	0.99	2165
f	1.00	1.00	1.00	2654
g	0.99	1.00	0.99	2290
h	0.99	0.99	0.99	2226
i	0.99	0.99	0.99	2161
j	1.00	0.99	1.00	2180
k	1.00	0.99	1.00	2387
l	1.00	1.00	1.00	2391
m	0.96	0.97	0.97	1461
n	0.97	0.95	0.96	1155
o	0.99	0.99	0.99	2035
p	1.00	0.99	1.00	1783
q	1.00	0.99	0.99	2003
r	0.98	0.99	0.98	2228
s	0.98	0.99	0.99	2261
space	0.98	0.99	0.98	1385
t	0.90	0.96	0.93	80
u	0.98	0.98	0.98	2218
v	0.99	0.99	0.99	2271
w	1.00	0.99	0.99	2250
x	0.99	0.99	0.99	2003
y	1.00	1.00	1.00	2248
z	0.99	1.00	1.00	1957
accuracy			0.99	55316
macro avg	0.99	0.99	0.99	55316
weighted avg	0.99	0.99	0.99	55316

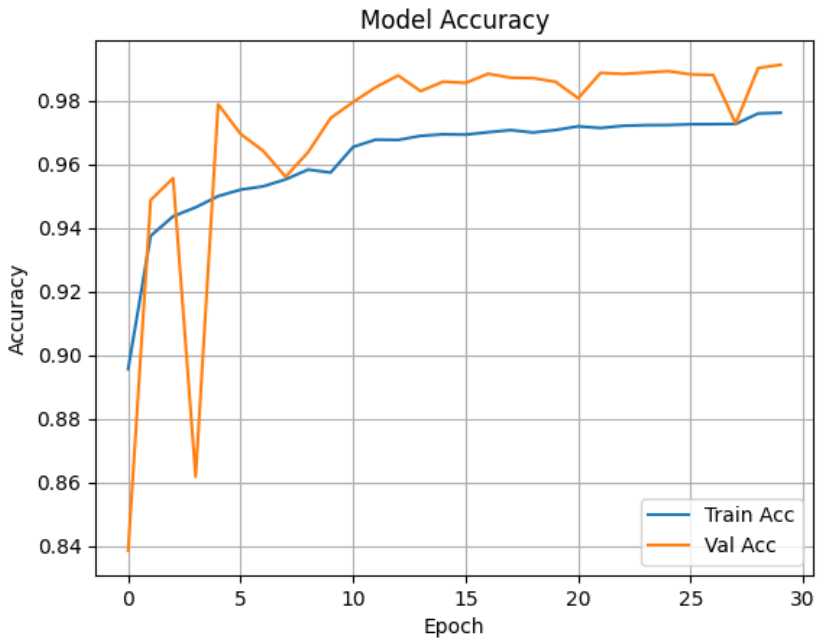

Fig. 1. Accuracy graph of each epoch of the FNN model

Figure 1 illustrates that the training accuracy gradually increases from approximately 89% in the first epoch to nearly 98% in the 30th epoch. Validation accuracy fluctuated at the beginning of training but then stabilized and even slightly exceeded the training accuracy for most epochs, hovering around 97% - 99%. The graph shows that the training process was stable, and the model effectively generalized to the validation data, with no significant overfitting, as evidenced by the narrow gap between training and validation accuracy.

The BiLSTM model with the Attention Mechanism for word recognition, when used in conjunction with K-Fold Cross-Validation, produces an accuracy of 78%, a Precision of 82%, a Recall of 80%, and an F1-score of 79%. The results are presented in Table 2.

Table 2. Classification results of BiLSTM + attention model, without cross-validation

	Precision	**Recall**	**F1-Score**	**Support**
Book	0.81	0.74	0.77	76
Happy	0.90	0.62	0.73	86
Hello	0.98	0.81	0.89	59
No	0.81	0.78	0.80	55
Please	0.92	0.84	0.88	81
Thank you	0.36	0.97	0.52	40
Where	0.90	0.84	0.87	67
Who	0.86	0.77	0.82	83
Why	0.90	0.78	0.83	58
Yes	0.74	0.82	0.78	51
Accuracy			0.78	656
Macro avg	0.82	0.80	0.79	656
Weighted avg	0.84	0.78	0.80	656

Figure 2 shows that the loss value consistently decreased from the beginning to the end of training, from around 2.25 to 0.45. This indicates that the training process was stable and the model successfully learned without underfitting. There were no sudden spikes indicating training instability. To evaluate the impact of implementing the Attention Mechanism on the BiLSTM model, experiments were conducted using the K-Fold Cross-Validation method with two scenarios: a pure BiLSTM model and a BiLSTM model with Attention. The initial hold-out evaluation produced an accuracy of 78%, while K-Fold Cross-Validation demonstrated improved generalization performance with an average accuracy of 92%. The results showed a significant performance improvement when Attention was integrated into the model architecture.

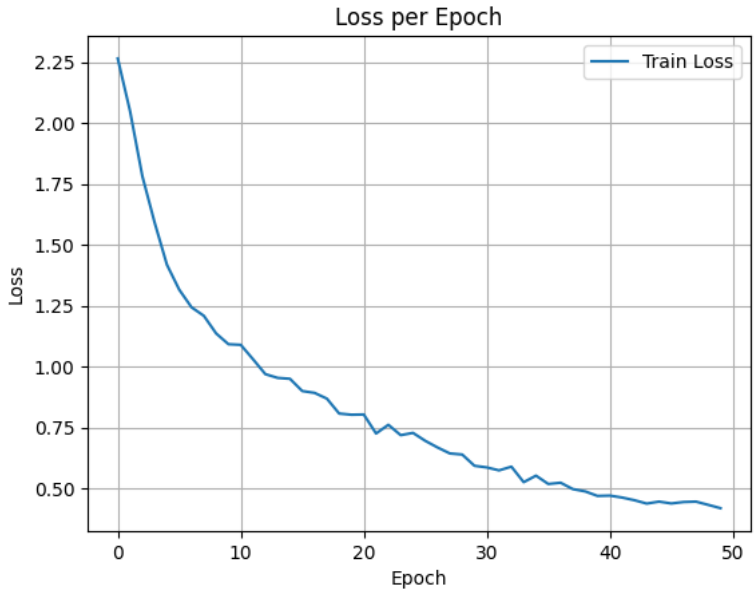

Fig. 2. Loss graph for each epoch of BiLSTM + attention model, without cross-validation

Table 3. Cross-Validation Classification Results of BiLSTM Model with Attention Mechanism

	Precision	Recall	F1-Score	Support
Book	0.96	0.88	0.91	49
Happy	0.91	1.00	0.95	50
Hello	0.98	0.95	0.96	42
No	1.00	0.90	0.95	42
Please	0.95	0.96	0.95	55
Thank you	0.81	0.96	0.88	49
Where	0.91	0.91	0.91	46
Who	0.85	0.85	0.85	55
Why	0.95	0.90	0.93	21
Yes	0.93	0.86	0.90	65
Accuracy			0.92	474
Macro avg	0.92	0.92	0.92	474
Weighted avg	0.92	0.92	0.92	474

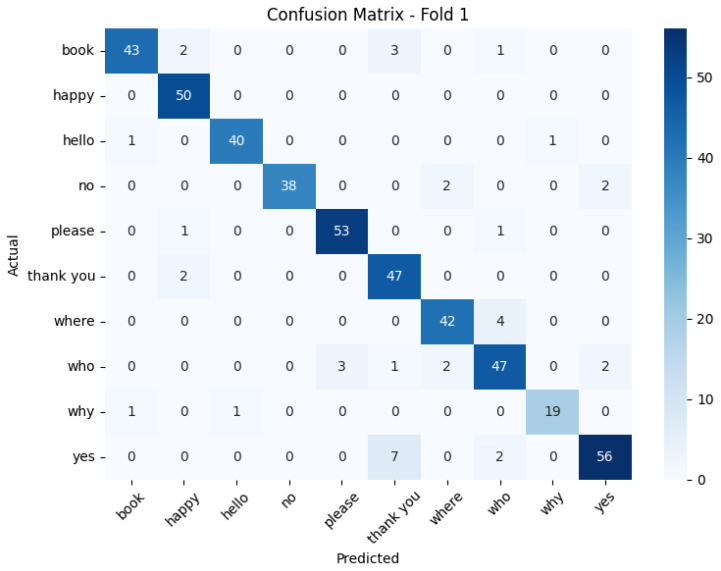

Fig. 1. Confusion matrix cross-validation of BiLSTM with attention mechanism

Based on the cross-validation results in Table 3, the classification metrics, and Figure 3, the confusion matrix of the BiLSTM + Attention model indicates that the model generally recognizes gestures well, with an overall accuracy of 92%. Some misclassifications still occurred for certain gestures, particularly those with similar movement patterns.

Based on the Cross-Validation results in table 4 regarding classification and figure 4 Confusion Matrix of the BiLSTM model without Attention, it can be observed that without Attention, the model experiences a decrease in accuracy in distinguishing gestures with similar spatial and temporal patterns of gestures such as “no”, “where”, “who”, and “yes” which is seen from the number of wrong predictions and a decrease in Recall and F1-Score values. For example, in the “no” class, Recall only reaches 0.64, and Precision is 0.75, which is far from when Attention is applied (Recall 0.90, Precision 1.00).

Table 4. Cross-validation classification results of BiLSTM model without attention mechanism

	Precision	**Recall**	**F1-Score**	**Support**
book	0.98	0.86	0.91	49
happy	0.87	0.90	0.88	50
hello	0.98	0.95	0.96	42
no	0.75	0.64	0.69	42
please	0.86	0.91	0.88	55
thank you	0.87	0.98	0.92	49
where	0.63	0.74	0.68	46
who	0.63	0.73	0.68	55
why	0.91	0.95	0.93	21
yes	0.86	0.66	0.75	65
Accuracy			**0.82**	474
Macro Avg	0.83	0.83	0.83	474
Weighted Avg	0.83	0.82	0.82	474

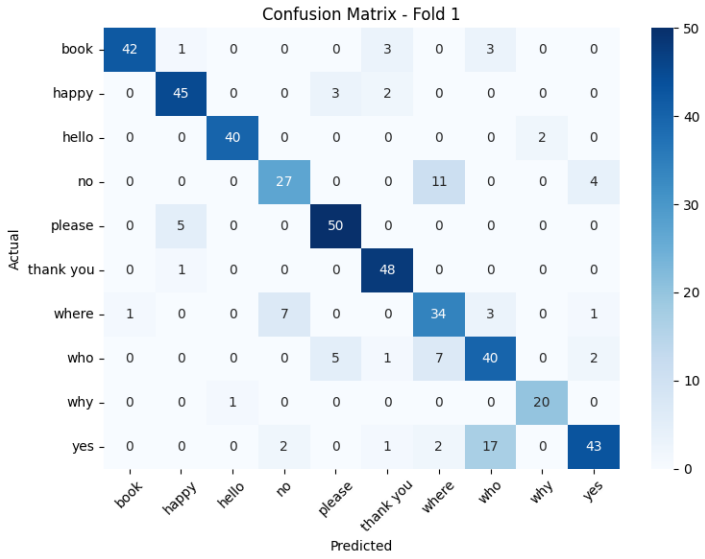

Fig. 2. Confusion matrix cross-validation of BiLSTM without attention mechanism

The addition of the Attention Mechanism has been shown to improve the performance of the BiLSTM model, particularly in recognizing gestures with similar shapes or complex movement sequences. Compared to the version without Attention, the model with Attention achieved an average F1-score improvement of 4% and demonstrated more stable convergence on the learning curve.

Confusion matrix evaluation reveals that specific categories, such as "hello," "thank you," and "yes," achieve very high recognition rates, with F1-scores exceeding 0.90. On the other hand, cues such as "why" and "book" experience performance degradation, indicated by lower Precision and Recall values. The highest identification errors occur in word gestures with visual similarities or limited training data, especially in the 'why' and 'who' classes, which require handling dataset imbalance through oversampling techniques. This phenomenon is triggered by the limited number of training samples and the similarity in visual characteristics between movements. To minimize identification errors in real-world operational conditions, the system is integrated with a voting buffer and confidence threshold mechanism, which only operates when both body and hand posture markers are detected simultaneously.

In terms of system performance, real-time testing demonstrated that the model can run smoothly on a mid-range laptop, achieving 15-20 FPS for word gestures and 20-30 FPS for letter gestures. The CPU usage for the BiLSTM model was approximately 70%, while for the FNN model, it was only 10%. The system is also designed to recognize gestures in approximately 2 seconds (30 frames) and will display "No Gesture Detected" if no hand is detected within a few frames.

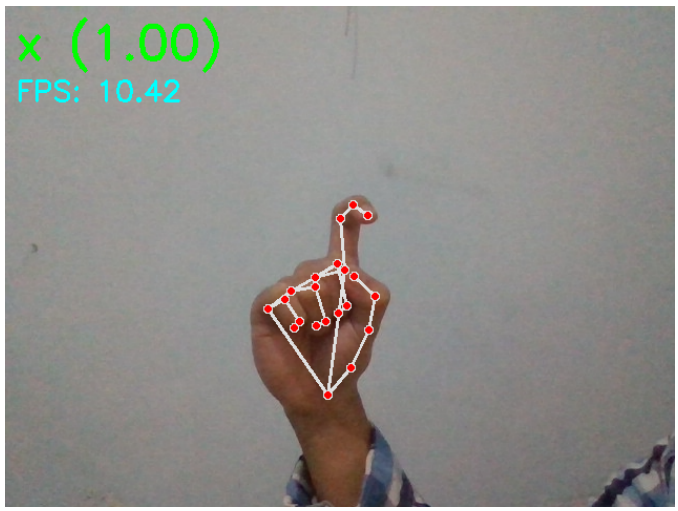

Fig. 5. Prediction letter x, right hand, bright light

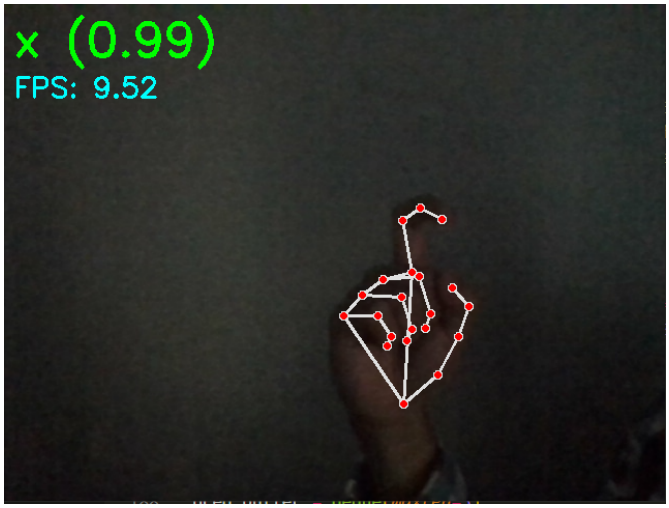

Fig. 3. Prediction letter x, right hand, dim light

Fig. 7. Prediction letter X, left Hand, bright light

Fig. 4. Prediction letter X, left hand, dim light

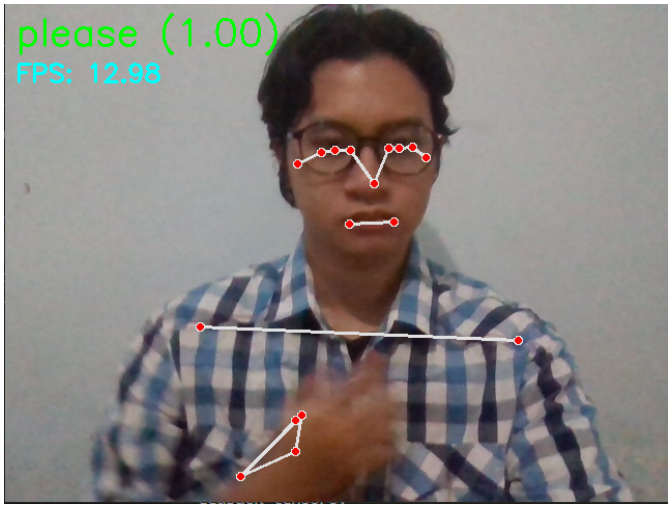

Fig. 5. Word prediction: 'please', right hand, bright light

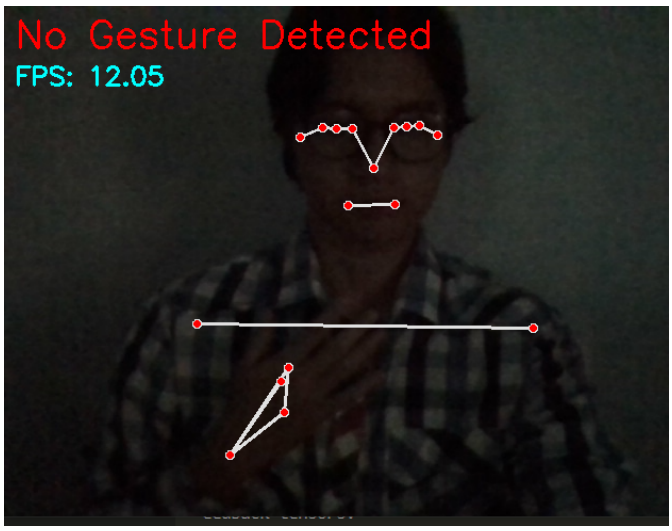

Fig. 6. Word prediction: 'please', right hand, dim light

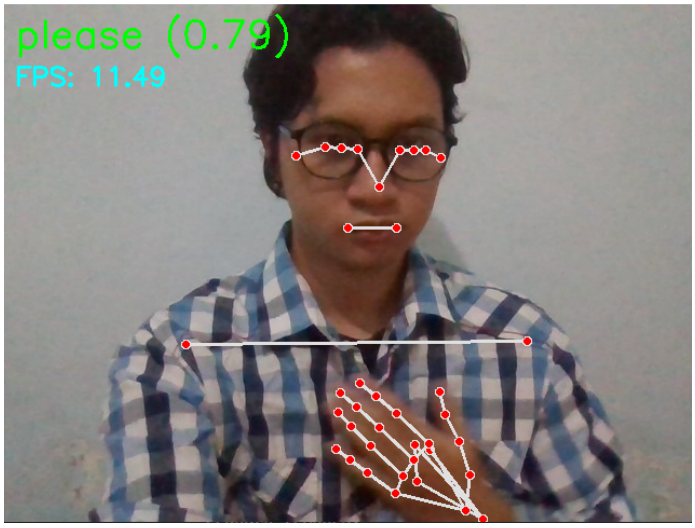

Fig. 7. Word prediction: 'please', left hand, bright light

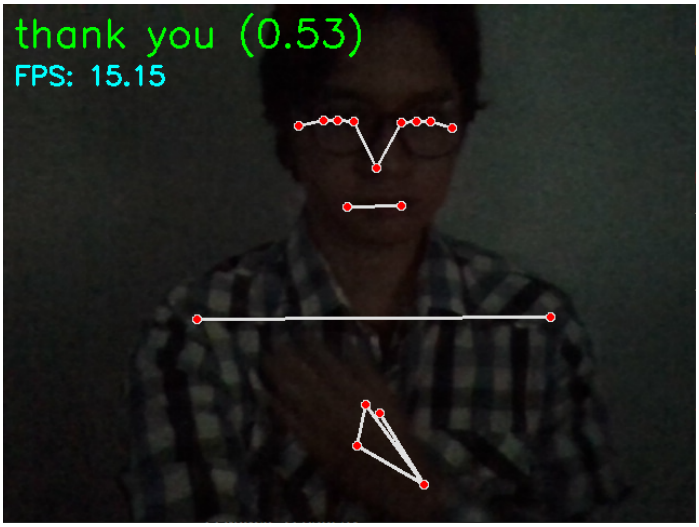

Fig. 8. Word prediction: 'please', left hand, dim light

Figures 5 through Figure 8 show predictions for the letter 'x' with high confidence. The 'x' gesture is still recognized even in low light, but is not detected against a dark background. Figures 9 through Figure 12 show gestures for the word 'please'. The 'please' gesture can still be recognized in bright light, although confidence decreases when the left hand is predicted. Testing in low light for the word 'please' results in 'No Gesture Detected' when predicting the right hand, while when the left hand is incorrectly predicted, it becomes 'thank you'.

These results demonstrate that the developed deep learning-based gesture recognition system is not only capable of accurately recognizing gestures but is also efficient enough to operate in real-time. Furthermore, the system has significant potential for application as a communication aid and sign language learning tool, especially if further developed with local datasets such as BISINDO or SIBI.

Conclusion

This research successfully designed and developed an American Sign Language (ASL) sign language recognition system for letters and words using a deep learning approach. The FNN model demonstrated very high performance in recognizing static ASL letters with an accuracy of up to 99%. The BiLSTM model with an Attention Mechanism showed good performance in recognizing sequence-based word gestures, with an accuracy of 78%. The use of MediaPipe for landmark extraction proved efficient and accurate under various lighting conditions and user positions.

Real-time testing demonstrated that the proposed system has sufficient speed and responsiveness for use in live interactions. The implementation of smoothing mechanisms, including the Exponential Moving Average and buffer majority voting, also improved prediction stability. These results show the effectiveness of combining MediaPipe-based landmark extraction with the BiLSTM-Attention architecture for real-time ASL translation. The proposed system serves as an adaptive prototype that can be further developed for local sign languages such as BISINDO or SIBI, involving direct collaboration with deaf communities and accessibility practitioners to evaluate usability and communication effectiveness in real-world environments.

Acknowledgements

The author would like to thank Universitas Atma Jaya Yogyakarta for its assistance during this research, particularly for granting access to the research site and providing facilities that contributed to the completion of this study.

References

Abdulhamied, R. M., Nasr, M. M., & Abdulkader, S. N. (2023). Real-time recognition of American sign language using long-short term memory neural network and hand detection. *Indonesian Journal of Electrical Engineering and Computer Science*, *30*(1), 545–556. https://doi.org/10.11591/ijeecs.v30.i1.pp545-556

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

Al-Qurishi, M., Khalid, T., & Souissi, R. (2021). Deep learning for sign language recognition: Current techniques, benchmarks, and open issues. *IEEE Access*, *9*, 126917–126951. https://doi.org/10.1109/ACCESS.2021.3110912

Alsharif, B., Alalwany, E., Ibrahim, A., Mahgoub, I., & Ilyas, M. (2025). Real-time American sign language interpretation using deep learning and keypoint tracking. *Sensors*, *25*(7), 2138. https://doi.org/10.3390/s25072138

Bora, J., Dehingia, S., Boruah, A., Chetia, A. A., & Gogoi, D. (2023). Real-time Assamese sign language recognition using MediaPipe and deep learning. *Procedia Computer Science*, *218*, 1384–1393. https://doi.org/10.1016/j.procs.2023.01.117

Gu, Y., Sherrine, Wei, W., Li, X., Yuan, J., & Todoh, M. (2022). American sign language alphabet recognition using inertial motion capture system with deep learning. *Inventions*,

7(4), 112. https://doi.org/10.3390/inventions7040112

Kazbekova, G., Ismagulova, Z., Ibrayeva, G., Sundetova, A., Abdrazakh, Y., & Baimurzayev, B. (2025). Real-time lightweight sign language recognition on hybrid deep CNN-BiLSTM neural network with attention mechanism. *International Journal of Advanced Computer Science and Applications (IJACSA)*, *16*(4). https://doi.org/10.14569/IJACSA.2025.0160452

Li, D., Rodriguez Opazo, C., Yu, X., & Li, H. (2020). Word-level deep sign language recognition from video: A new large-scale dataset and methods comparison (arXiv:1910.11006). *2020 IEEE Winter Conference on Applications of Computer Vision (WACV)*. https://doi.org/10.1109/WACV45572.2020.9093512

Lugaresi, C., Tang, J., Nash, H., McClanahan, C., Uboweja, E., Hays, M., Zhang, F., Chang, C.-L., Yong, M. G., Lee, J., Chang, W.-T., Hua, W., Georg, M., & Grundmann, M. (2019). MediaPipe: A framework for building perception pipelines (arXiv:1906.08172). *arXiv*. https://arxiv.org/abs/1906.08172

Nagaraj, A. (2018). *ASL Alphabet*. Kaggle. https://doi.org/10.34740/KAGGLE/DSV/29550

Orovwode, H., Oduntan, I. D., & Abubakar, J. (2023). Development of a sign language recognition system using machine learning. *2023 International Conference on Artificial Intelligence, Big Data, Computing and Data Communication Systems (icABCD) (pp. 1–8)*. https://doi.org/10.1109/icABCD59051.2023.10220456

Samaan, G. H., Wadie, A. R., Attia, A. K., Asaad, A. M., Kamel, A. E., Slim, S. O., Abdallah, M. S., & Cho, Y.-I. (2022). MediaPipe's landmarks with RNN for dynamic sign language recognition. *Electronics*, *11*(19), 3228. https://doi.org/10.3390/electronics11193228

Sherstinsky, A. (2020). Fundamentals of recurrent neural network (RNN) and long short-term memory (LSTM) network. *Physica D: Nonlinear Phenomena*, *404*, 132306. https://doi.org/10.1016/j.physd.2019.132306

Sundar, B., & Bagyammal, T. (2022). American sign language recognition for alphabets using MediaPipe and LSTM. *Procedia Computer Science*, *215*, 642–651. https://doi.org/10.1016/j.procs.2022.12.066

Zhang, J., Bu, X., Wang, Y., et al. (2024). Sign language recognition based on dual-path background erasure convolutional neural network. *Scientific Reports*, *14*, 11360. https://doi.org/10.1038/s41598-024-62008-z

Zhang, Z., Feng, F., & Huang, T. (2022). FNNS: An effective feedforward neural network scheme with random weights for processing large-scale datasets. *Applied Sciences*, *12*(23), 12478. https://doi.org/10.3390/app122312478