Journal of Intelligent Learning Systems and Applications, 2026, 18(1), 11-21
https://www.scirp.org/journal/jilsa
ISSN Online: 2150-8410
ISSN Print: 2150-8402

A Multi-Modal Approach for Arabic Sign Language Gesture Recognition Using Deep Learning

Nouf Alharbi

College of Computer Science and Engineering, Taibah University, Madinah, Saudi Arabia
Email: nmoharbi@taibahu.edu.sa

How to cite this paper: Alharbi, N. (2026) A Multi-Modal Approach for Arabic Sign Language Gesture Recognition Using Deep Learning. *Journal of Intelligent Learning Systems and Applications*, **18**, 11-21.
https://doi.org/10.4236/jilsa.2026.181002

Received: October 15, 2025
Accepted: January 26, 2026
Published: January 29, 2026

Abstract

This paper proposes a multi-modal deep learning framework for Arabic Sign Language (ArSL) recognition, addressing the challenges of both static and dynamic gesture recognition. The framework integrates spatial, temporal, and depth features using CNN, Transformer, and Depth-CNN models, combined via an attention-based fusion mechanism. A hierarchical recognition approach first classifies gestures as static or dynamic, then processes them with specialized models: MobileNetV3 for dynamic gestures and an MLP-KAN hybrid for static gestures. Evaluated on four ArSL datasets (Kaggle ASL, ArSL2018, DArSL50, KSU-ArSL), the system achieves 98.4% overall accuracy with real-time inference speeds of 0.007 seconds for static gestures and 0.02 seconds for dynamic gestures. Ablation studies confirm the importance of multi-modal fusion, with attention-based fusion improving accuracy by 11% compared to simple concatenation. The system demonstrates strong generalization across diverse datasets and conditions, making it suitable for real-world deployment in assistive communication technologies.

Keywords

Arabic Sign Language, Gesture Recognition, Deep Learning, Multi-Modal Feature Extraction, Attention-Based Fusion, CNN, Transformer, Depth-CNN, MLP, KAN

1. Introduction

Sign language is the primary mode of communication for millions of deaf and hard-of-hearing individuals worldwide. According to the World Health Organization (WHO), over 1.5 billion people globally experience some degree of hearing

loss, with approximately 430 million requiring rehabilitation for disabling hearing loss [1]. Within the Middle East and North Africa (MENA) region, the prevalence of hearing disabilities is substantial, with more than 11 million individuals affected [2].

Unlike American Sign Language (ASL) and British Sign Language (BSL), which have well-documented linguistic structures and large annotated datasets, Arabic Sign Language (ArSL) presents unique challenges due to dialectal variations, limited datasets, and the complexity of dynamic gestures [3]. ArSL lacks a standardized form, as different Arab countries have developed their own dialects. This variation makes it difficult to create a unified recognition model that generalizes across multiple regions. Existing ArSL recognition systems primarily focus on static gestures often neglecting dynamic movements, which are important for understanding continuous sign language communication [4].

Several approaches have been explored for Sign Language Recognition (SLR), including traditional computer vision techniques, Deep Learning (DL)-based image classification, and sequence modeling using Recurrent Neural Networks (RNNs), Long Short-Term Memory (LSTM) networks, and Transformers [5]. Early methods relied on handcrafted feature extraction using edge detection, Histogram of Oriented Gradients (HOG), and optical flow analysis [6]-[8]. However, these methods struggled with variations in lighting, background noise, and signer differences. With the advancement of DL, Convolutional Neural Networks (CNNs) have been widely used for spatial feature extraction from sign images [6] [8] [9]. While CNNs excel in static gesture recognition, they fail to capture temporal dependencies. To address this, RNNs and LSTMs have been employed for sequential modeling, but they suffer from vanishing gradient problems and high computational costs [5] [10]. More recently, Transformer-based models have gained popularity due to their ability to model long-range dependencies efficiently [11].

To address these challenges, this study proposes a multi-modal DL framework for ArSL recognition, incorporating spatial, temporal, and depth features. The key contributions of this work are:

- **Hierarchical Recognition Framework**: A novel two-tier approach that first classifies gestures into static or dynamic categories using a CNN classifier, followed by specialized models for precise recognition.
- **Attention-Based Multi-Modal Feature Fusion**: A fusion mechanism that integrates spatial (CNN), temporal (Transformer), and depth (Depth-CNN) features to achieve rich feature representation.
- **Extensive Benchmarking on Multiple ArSL Datasets**: Four distinct ArSL datasets are used to assess the proposed methodology.
- **Efficient Real-Time Processing**: The system achieves over 98% accuracy with an inference time of 0.007 seconds for static gestures and 0.02 seconds for dynamic gestures.

2. Related Works

SLR has emerged as a critical research area aimed at bridging communication bar-

riers for deaf and mute communities. A number of studies have explored different methodologies for ArSL recognition, employing Machine Learning (ML), DL, and hybrid models to address the inherent challenges in static and dynamic gesture recognition.

Tharwat, Ahmed, and Bouallegue (2021) proposed a vision-based system for recognizing ArSL alphabets, emphasizing the use of traditional ML techniques [12]. Their Arabic Alphabet Sign Language Recognition System (AArSLRS) employed a dataset of 9240 images and achieved 99.5% accuracy with KNN under controlled conditions, but was limited to static gestures.

Duwairi and Halloush (2022) employed transfer learning techniques for ArSL alphabet recognition, utilizing pre-trained models like AlexNet, VGGNet, and GoogleNet [6]. Using the ArSL2018 dataset comprising 54,049 images of 32 Arabic characters, VGGNet achieved an accuracy of 97%, but focused only on static gestures.

Noor *et al.* (2024) proposed a hybrid model integrating CNN and LSTM networks for both static and dynamic gestures [5]. Their framework utilized a custom dataset of 4000 images for static gestures and 500 videos for dynamic sequences, achieving accuracies of 94.4% and 82.7% respectively.

Ameer *et al.* (2024) extended the focus on dynamic gesture recognition with an attention-based LSTM model [10]. Their DArSL50 dataset, comprising 50 dynamic gestures across 7500 videos, served as the foundation, achieving accuracies of 85% for individual volunteers.

Zakariah *et al.* (2022) explored transfer learning for ArSL recognition using the EfficientNetB4 architecture [13]. Using the ArSL2018 dataset, EfficientNetB4 achieved a testing accuracy of 95%, but relied on single-hand static gestures with high computational requirements.

Hdioud and Tirari (2023) proposed a DL-based ArSL system designed to recognize Arabic letters [14]. Their approach combined pre-processing with MediaPipe and a custom CNN architecture, achieving 97.07% validation accuracy, but was limited to static gestures.

Alharthi and Alzahrani (2023) explored the integration of Vision Transformer (ViT) and transfer learning [7]. Their study utilized pretrained models like InceptionResNetV2, ViT, and Swin, achieving a maximum accuracy of 98.17% with InceptionResNetV2.

Al Khuzayem *et al.* (2024) focused on Saudi Sign Language (SSL) recognition with their Efhamni application, which applied a CNN and Bidirectional Long Short-Term Memory (BiLSTM) architecture [15]. The model was trained on the KSU-SSL dataset, achieving precision of 94.61%, recall of 94.56%, and F1-score of 94.52%.

Alsolai *et al.* (2024) proposed the SLDC-RSAHDL framework, which utilized MobileNet for feature extraction, coupled with a hybrid DL model integrating CNN and LSTM layers [8]. Using the ASL alphabet dataset, the system achieved an accuracy of 99.51%.

While these studies have advanced the state of the art in ArSL recognition, several gaps remain: limited handling of dynamic gestures, small or non-diverse datasets, high computational requirements, single-modality approaches, and lack of attention to dialectal variations [16] [17].

3. Proposed Methodology

3.1. Overview

The proposed methodology diagram is illustrated in Figure 1. The proposed methodology for ArSL integrates multiple phases: dataset collection, pre-processing, multi-modal feature extraction, attention-based fusion, and hierarchical recognition. The methodology begins with collecting four datasets: Kaggle ASL, ArSL2018, DArSL50, and KSU-ArSL. These datasets cover static and dynamic gestures, essential for training the model. Pre-processing involves normalization of data, keypoint extraction using Mediapipe, and depth map generation. Feature extraction uses three models: CNN for spatial features, Transformer for temporal features, and Depth-CNN for depth features. The extracted features are fused using an attention-based mechanism. The framework consists of two tiers. Tier 1 classifies gestures as static or dynamic using a CNN classifier. Tier 2 processes dynamic gestures through MobileNetV3 and static gestures through a hybrid MLP and KAN model for accurate gesture recognition.

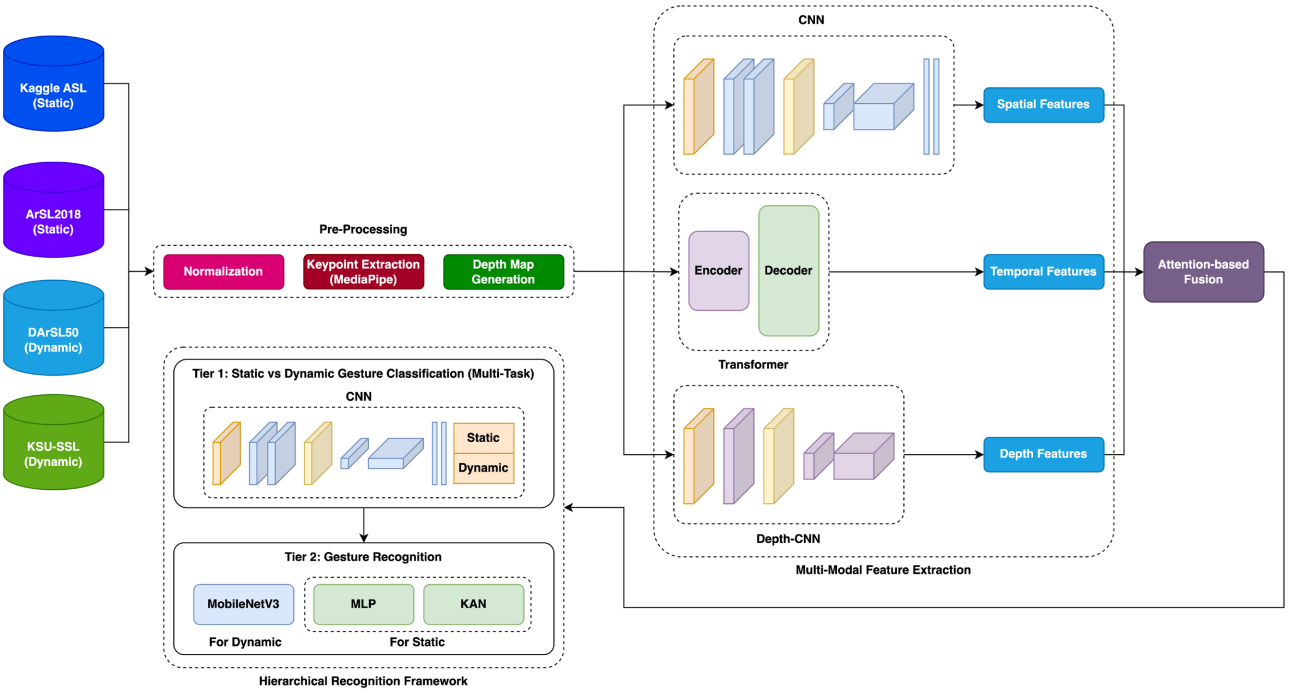

Figure 1. Overview of the proposed methodology for Arabic SLR.

3.2. Multi-Modal Feature Extraction

Multi-Modal Feature Extraction plays a climactic role in capturing diverse aspects of ArSL gestures. This phase involves the extraction of three distinct feature types:

spatial, temporal, and depth features. Spatial features are extracted using a CNN, focusing on the image-based representation of static gestures. Temporal features are captured through a Transformer model, which processes the time-dependent characteristics of dynamic gestures. Depth features are obtained through a Depth-CNN, which analyzes the depth information from depth maps.

3.2.1. Spatial Features

Spatial features capture essential patterns, shapes, and edges from hand gesture images. The CNN model consists of four convolutional layers, each followed by a max-pooling operation. Figure 2 illustrates the architecture. Table 1 presents the detailed hyperparameter settings.

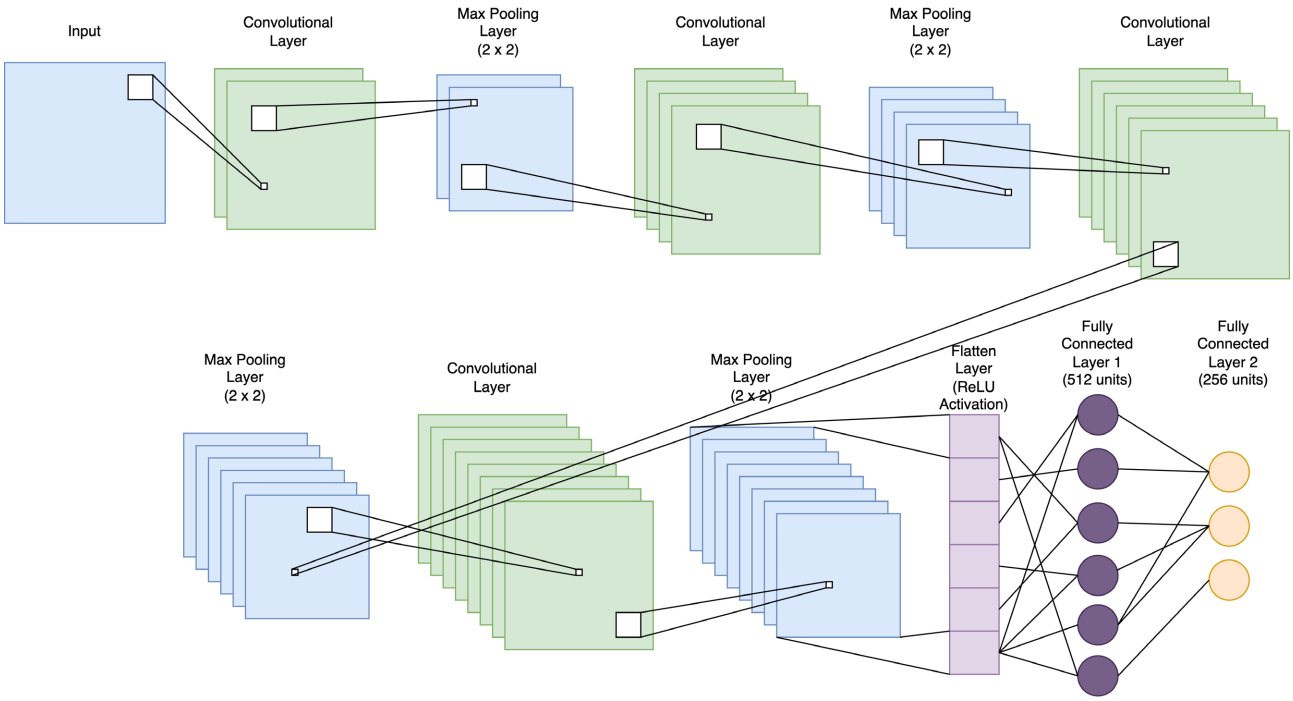

Figure 2. Visual architecture illustration of the CNN model for spatial feature extraction.

Table 1. Hyperparameter details for CNN architecture (Spatial Feature Extraction).

Hyperparameter	Value
Learning Rate	0.001
Batch Size	32
Epochs	50
Optimizer	Adam
Dropout Rate	0.25
Filter Sizes	3 × 3
Number of Filters	32 (1st), 64 (2nd), 128 (3rd), 256 (4th)
Pooling Type	Max Pooling
Pooling Window Size	2 × 2

Continued

Activation Function	ReLU (hidden layers)
Fully Connected Layer 1 Units	512
Fully Connected Layer 2 Units	256

3.2.2. Temporal Features

Temporal features capture motion patterns and sequential dependencies within dynamic gestures. The Transformer model processes sequential input to learn contextual relationships.

Table 2 presents the hyperparameter settings.

Table 2. Hyperparameter settings for the transformer model.

Hyperparameter	Value
Embedding Dimension	512
Number of Attention Heads	8
Number of Transformer Layers	6
Feedforward Dimension	2048
Dropout Rate	0.1
Optimizer	AdamW
Learning Rate	0.0001
Batch Size	32
Number of Epochs	50

3.2.3. Depth Features

Depth features capture the three-dimensional structure of hand movements. The Depth-CNN model consists of four convolutional layers followed by max-pooling operations. Figure 3 illustrates the architecture. Table 3 presents the hyperparameters.

Table 3. Hyperparameter settings for the Depth-CNN model.

Hyperparameter	Value
Input Depth Map Size	224 × 224
Number of Convolutional Layers	4
Filter Sizes	3 × 3
Number of Filters	32, 64, 128, 256
Dropout Rate	0.3
Optimizer	Adam
Learning Rate	0.001
Batch Size	32
Number of Epochs	50

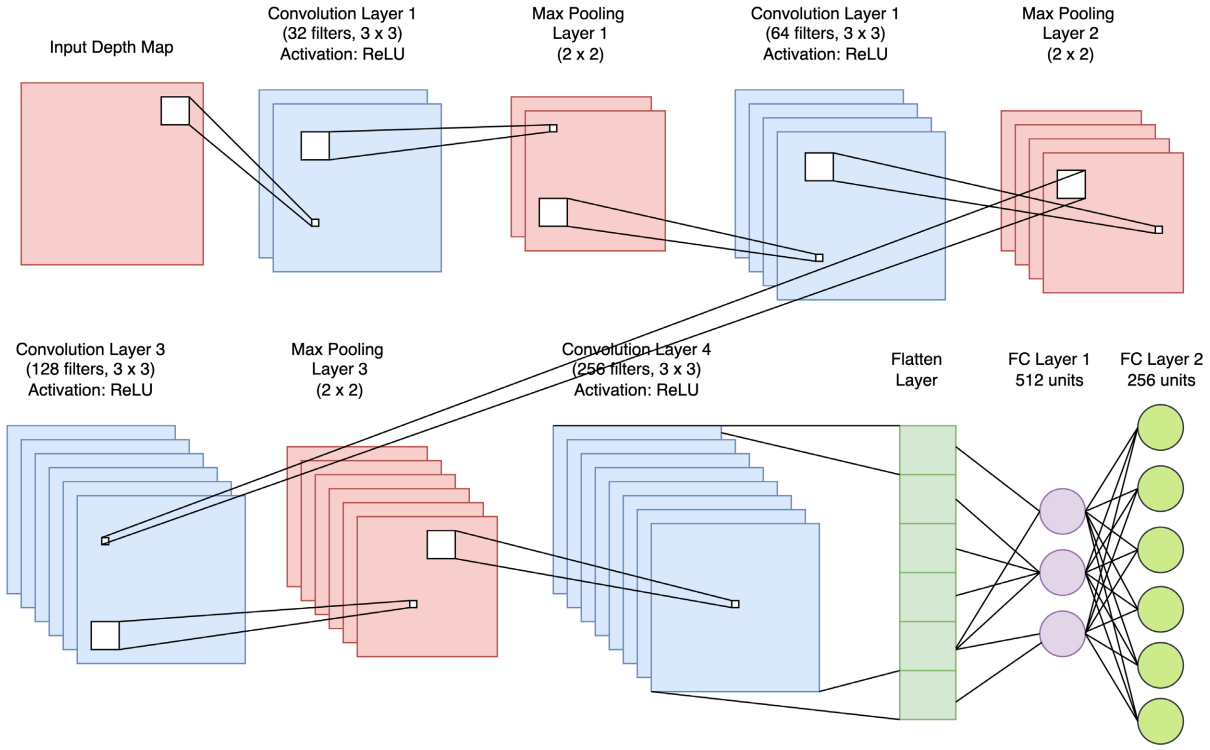

Figure 3. Visual architecture illustration of the depth-CNN model for depth feature extraction.

3.2.4. Attention-Based Fusion

An attention-based fusion mechanism is employed to assign adaptive importance to each feature representation. Let the extracted feature vectors be:

$$F_s \in \mathbb{R}^{d_s},\ F_t \in \mathbb{R}^{d_t},\ F_d \in \mathbb{R}^{\alpha_d} \tag{1}$$

These are concatenated: $F = \left[F_s; F_t; F_d\right]$. Attention weights are computed:

$$\alpha = \text{softmax}\left(WF\right) \tag{2}$$

where $\alpha = \left[\alpha_s, \alpha_t, \alpha_d\right]$. The weighted fusion is:

$$F_{\text{fusion}} = \alpha_s F_s + \alpha_t F_t + \alpha_d F_d \tag{3}$$

3.3. Hierarchical Recognition Framework

The framework consists of two tiers. Tier 1 classifies gestures as static or dynamic. Tier 2 processes them with specialized models.

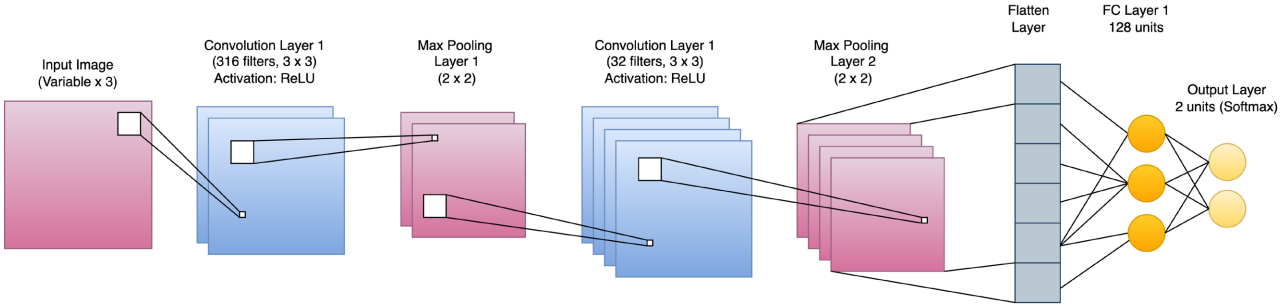

Figure 4. Architecture of the CNN model used for static vs. dynamic gesture classification.

3.3.1. Tier 1—Static vs. Dynamic Gesture Classification

A CNN classifier with two convolutional layers is used. Figure 4 illustrates the architecture. Table 4 presents the hyperparameters.

Table 4. Hyperparameters of the CNN model for static vs. dynamic gesture classification.

Hyperparameter	Value
Learning Rate	0.001
Batch Size	32
Epochs	20
Optimizer	Adam
Dropout Rate	0.2
Number of Filters	16 (1st), 32 (2nd)
Fully Connected Layer Units	128

3.3.2. Tier 2—Gesture Recognition

Dynamic gestures are recognized using MobileNetV3. Static gestures are classified using a hybrid MLP and KAN model.

Dynamic Gesture Recognition

MobileNetV3 is used with hyperparameters in Table 5.

Table 5. Hyperparameters of MobileNetV3 for dynamic gesture recognition.

Hyperparameter	Value
Architecture Type	MobileNetV3-Small
Activation Function	Hard-Swish
Dropout Rate	0.2
Optimizer	Adam
Learning Rate	0.0005
Batch Size	32
Number of Epochs	50

Static Gesture Recognition

The hybrid MLP-KAN model is used. Hyperparameters for MLP and KAN are in Table 6 and Table 7.

Table 6. Hyperparameters of MLP for static gesture recognition.

Hyperparameter	Value
Number of Hidden Layers	2
Hidden Layer 1 Units	512
Hidden Layer 2 Units	256
Dropout Rate	0.3
Optimizer	Adam
Learning Rate	0.001
Batch Size	32
Number of Epochs	50

Table 7. Hyperparameters of KAN for static gesture recognition.

Hyperparameter	Value
Number of Layers	3
Knowledge Module Type	Graph-Based
Hidden Layer 1 Units	512
Hidden Layer 2 Units	256
Dropout Rate	0.3
Optimizer	AdamW
Learning Rate	0.0001
Batch Size	32
Number of Epochs	50

4. Experimental Setup

The experiments were conducted using Python and TensorFlow on a system with an NVIDIA GPU. Datasets were split into 70% training, 15% validation, and 15% testing. Evaluation metrics included accuracy, precision, recall, F1-score, training time, and inference speed.

5. Results and Discussion

5.1. Performance Evaluation

The proposed multi-modal model achieved an exceptional 98.4% overall accuracy. For static gesture recognition: precision = 98.7%, recall = 98.4%, F1-score = 98.5%. For dynamic gestures: precision = 97.9%, recall = 98.2%, F1-score = 98.0%. Figure 5 shows the performance metrics.

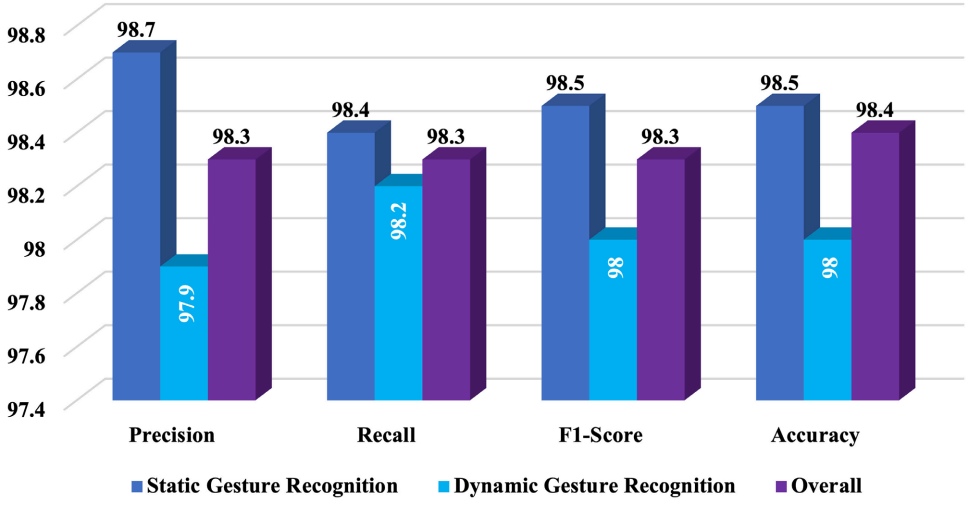

Figure 5. Performance evaluation metrics of the proposed model.

The model exhibited impressive computational efficiency. Inference time was 0.007 seconds for static and 0.02 seconds for dynamic gestures (Figure 6).

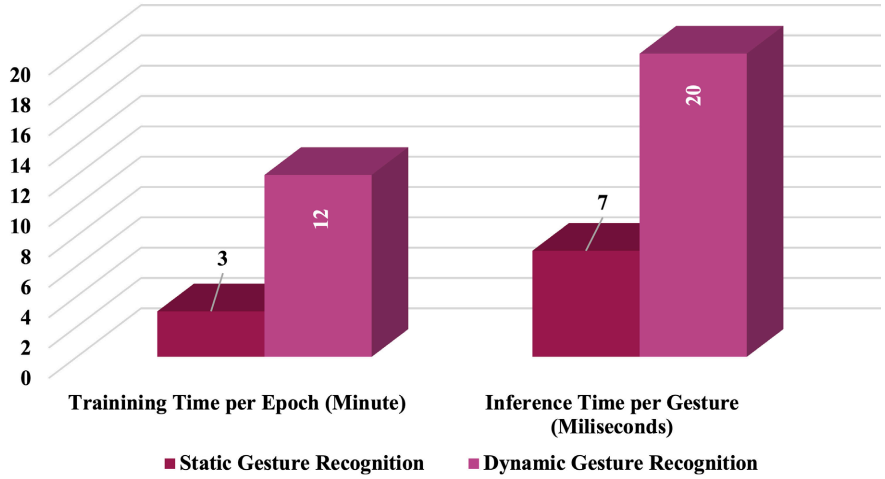

Figure 6. Training and inference time comparison.

5.2. Ablation Study

Ablation studies confirmed the importance of each component:

- Without attention-based fusion: Accuracy dropped to 87.4%.
- Spatial-only (CNN): 92.3%.
- Temporal-only (Transformer): 93.1%.
- Depth-only (Depth-CNN): 90.2%.
- All modalities combined: 98.4%.

6. Conclusion

This paper presents a comprehensive approach for ArSL recognition, addressing challenges associated with both static and dynamic gestures. The proposed methodology leverages multi-modal feature extraction (CNN, Transformer, Depth-CNN) with attention-based fusion, followed by a hierarchical recognition framework. The system achieves 98.4% accuracy with low inference times, demonstrating its potential for real-world applications. Future work will focus on expanding dialectal coverage and optimizing for mobile deployment.

Conflicts of Interest

The author declares no conflicts of interest regarding the publication of this paper.

References

[1] World Health Organization (2025) Deafness and Hearing Loss. https://www.who.int/news-room/fact-sheets/detail/deafness-and-hearing-loss

[2] Center for Strategic and International Studies (2025) Disability Inclusion in Foreign Policy: Special Advisor Sara Minkara. https://www.csis.org/events/disability-inclusion-foreign-policy-special-advisor-sara-minkara

[3] Shin, J., Miah, A.S.M., Kabir, M.H., Rahim, M.A. and Al Shiam, A. (2024) A Methodological and Structural Review of Hand Gesture Recognition across Diverse Data Modalities. *IEEE Access*, **12**, 142606-142639.

https://doi.org/10.1109/access.2024.3456436

[4] Al Abdullah, B.A., Amoudi, G.A. and 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

[5] Noor, T.H., Noor, A., Alharbi, A.F., Faisal, A., Alrashidi, R., Alsaedi, A.S., et al. (2024) Real-Time Arabic Sign Language Recognition Using a Hybrid Deep Learning Model. *Sensors*, **24**, Article 3683. https://doi.org/10.3390/s24113683

[6] Duwairi, R.M. and Halloush, Z.A. (2022) Automatic Recognition of Arabic Alphabets Sign Language Using Deep Learning. *International Journal of Electrical and Computer Engineering* (*IJECE*), **12**, 2996-3004. https://doi.org/10.11591/ijece.v12i3.pp2996-3004

[7] Alharthi, N.M. and Alzahrani, S.M. (2023) Vision Transformers and Transfer Learning Approaches for Arabic Sign Language Recognition. *Applied Sciences*, **13**, Article 11625. https://doi.org/10.3390/app132111625

[8] Alsolai, H., Alsolai, L., Al-Wesabi, F.N., Othman, M., Rizwanullah, M. and Abdelmageed, A.A. (2024) Automated Sign Language Detection and Classification Using Reptile Search Algorithm with Hybrid Deep Learning. *Heliyon*, **10**, e23252. https://doi.org/10.1016/j.heliyon.2023.e23252

[9] Kong, F., Hu, K., Li, Y., Li, D., Liu, X. and Durrani, T.S. (2022) A Spectral-Spatial Feature Extraction Method with Polydirectional CNN for Multispectral Image Compression. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, **15**, 2745-2758. https://doi.org/10.1109/jstars.2022.3158281

[10] Abdul Ameer, R.S., Ahmed, M.A., Al-Qaysi, Z.T., Salih, M.M. and Shuwandy, M.L. (2024) Empowering Communication: A Deep Learning Framework for Arabic Sign Language Recognition with an Attention Mechanism. *Computers*, **13**, Article 153. https://doi.org/10.3390/computers13060153

[11] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A., Kaiser, L. and Polosukhin, I. (2017) Attention Is All You Need. arXiv: 1706.03762.

[12] Tharwat, G., Ahmed, A.M. and Bouallegue, B. (2021) Arabic Sign Language Recognition System for Alphabets Using Machine Learning Techniques. *Journal of Electrical and Computer Engineering*, **2021**, Article ID: 2995851. https://doi.org/10.1155/2021/2995851

[13] Zakariah, M., Alotaibi, Y.A., Koundal, D., Guo, Y. and Mamun Elahi, M. (2022) Sign Language Recognition for Arabic Alphabets Using Transfer Learning Technique. *Computational Intelligence and Neuroscience*, **2022**, Article ID: 4567989. https://doi.org/10.1155/2022/4567989

[14] Hdioud, B. and Tirari, M.E.H. (2023) A Deep Learning Based Approach for Recognition of Arabic Sign Language Letters. *International Journal of Advanced Computer Science and Applications*, **14**, 424-429. https://doi.org/10.14569/ijacsa.2023.0140447

[15] Al Khuzayem, L., Shafi, S., Aljahdali, S., Alkhamesie, R. and Alzamzami, O. (2024) Efhamni: A Deep Learning-Based Saudi Sign Language Recognition Application. *Sensors*, **24**, Article 3112. https://doi.org/10.3390/s24103112

[16] Zhang, Y. and Jiang, X. (2024) Recent Advances on Deep Learning for Sign Language Recognition. *Computer Modeling in Engineering & Sciences*, **139**, 2399-2450. https://doi.org/10.32604/cmes.2023.045731

[17] Gao, Q., Zhang, M. and Ju, Z. (2025) LGF-SLR: Hand Local-Global Fusion Network for Skeleton-Based Sign Language Recognition. *IEEE Sensors Journal*, **25**, 8586-8597. https://doi.org/10.1109/jsen.2025.3527198