Journal of Artificial Intelligence and Data Mining (JAIDM), Vol. 14, No. 2, 2026, 209-220.

Shahrood University of Technology

Journal of Artificial Intelligence and Data Mining (JAIDM)
Journal homepage: http://jad.shahroodut.ac.ir

Research paper

Sign Language Recognition Using a Hybrid Model Based on Convolutional Neural Networks and Hidden Markov Models

Malihe Danesh* and Zahra Ahmadi

Department of Computer Engineering, University of Science and Technology of Mazandaran, Behshahr, Iran.

Article Info

Article History:
Received 02 July 2025
Revised 02 November2025
Accepted 19 December 2025

DOI:10.22044/jadm.2025.16424.2766

Keywords:
Sign Language, Convolutional Neural Networks, Hidden Markov Models, Recognition.

**Corresponding author: m.danesh@mazust.ac.ir (M. Danesh).*

Abstract

In recent years, sign language recognition has emerged as a major challenge in the fields of image processing and machine learning. People with hearing impairments use sign language to communicate, but the lack of automated tools to translate it has created significant communication barriers. This study presents a hybrid model based on convolutional neural networks (CNNs), transformers, and hidden Markov models (HMMs) to accurately recognize sign language gestures using the MNIST sign language dataset. The model first extracts image features from handwritten images using CNNs and then feeds these features into the transformer model to process complex and long-term dependencies in the feature sequence. In the next step, to smooth the predictions and improve accuracy, a hidden Markov model is employed, which adjusts the final predictions based on previous sequences. The results show that the proposed model utilizing HMM achieves an accuracy of 99% and a sign error rate of 0.0098, demonstrating its high efficiency in recognizing hand gestures. This research represents an important step toward developing assistive devices for the deaf and enhancing human interaction.

1. Introduction

Sign language is used as a fundamental communication system for deaf and hard-of-hearing individuals worldwide. However, one of the greatest challenges in communication for these individuals is the lack of accurate and efficient tools for translating and recognizing sign language into text or speech. This limitation causes communication difficulties in daily life and creates barriers to social, educational, and professional interactions [1]. For this reason, extensive research has been conducted in the development of automatic systems for sign language recognition.

In recent years, significant advancements in deep learning and image processing have provided new opportunities for designing more complex models for processing image and sequence data [2]. However, several challenges still remain. Specifically, the complex and diverse characteristics of sign language, including the varying speed and accuracy of hand movements, different types of gestures, and individual differences in how gestures are performed, significantly impact the accuracy of recognition systems. Furthermore, many previous models were only capable of recognizing specific or simple gestures and showed poor performance when faced with changes in environmental or temporal conditions [3]. Additionally, older methods relied on manual feature extraction and simpler algorithms, which typically struggled with recognizing the complex temporal sequences of sign language [4].

A review of studies between 2014 and 2020 shows that most research has focused on vision-based gesture recognition [5], emphasizing three main aspects: data collection, the data environment, and gesture representation.

Results from 98 papers indicate an average accuracy of 88.8% in signer-dependent mode and 78.2% in signer-independent mode. Despite significant progress, challenges such as accurate and continuous motion recognition remain.

Although significant progress has been made in hand gesture recognition, several challenges persist, particularly in achieving accurate and continuous motion recognition. Previous studies on Hand Gesture Recognition (HGR) from 2014 to 2024 [6] have investigated diverse input modalities such as RGB, skeletal, depth, audio, EMG, EEG, and hybrid data, highlighting the complexity and multimodal nature of this field. The findings show that despite significant advances, there is still room for improvement in continuous motion and vision-based systems. Additionally, due to the increasing presence of robots in daily life, creating natural interaction between humans and robots has become one of the key issues in the field of robotics. Continuing research on hand gesture recognition has focused on vision-based methods to make human-robot interaction as similar as possible to the interaction between two humans [7].

Hand gestures are considered one of the intuitive, creative, and non-verbal methods for communicating with robots, and processes such as data acquisition, hand recognition and separation, feature extraction, and movement classification have been investigated. The role of RGB and RGB-D cameras in improving this process has also been analyzed, along with the challenges and improvements needed to upgrade these systems for real human-robot interactions.

In recent years, significant progress has been made in the use of machine vision, sensor, and hybrid models for hand gesture recognition. This study systematically reviews research from 2018 to 2023, examining different approaches in this area, and shows that although the accuracy of models varies across conditions, challenges such as continuous gesture recognition and lack of training data still prevent widespread use in real-world environments [8].

In this study, to address the existing challenges, an advanced model based on Convolutional Neural Networks (CNN), Transformer, and Hidden Markov Models (HMM) is proposed. Using CNN for feature extraction from sign language images, modeling temporal dependencies with the Transformer, and enhancing predictions through temporal sequence analysis with HMM, as a comprehensive approach, significantly improves the system's accuracy in recognizing sign language gestures. The goal of this model is to design a high-precision automatic system for recognizing and translating sign language in various social and professional environments. In this model, the combination of image features extracted by CNN, temporal dependencies processed by the Transformer, along with prediction improvements through HMM, enables more accurate and adaptable recognition in varying conditions.

This research, aiming to enhance the accuracy and efficiency of sign language recognition systems, particularly in environments with changing conditions, proposes an innovative and comprehensive approach for precise sign language gesture recognition, which can play a crucial role in the development of assistive tools for the deaf and in improving human-machine interaction.

Despite recent progress in deep learning-based sign language recognition, existing models still struggle to handle temporal inconsistencies, visual ambiguities, and variations in gesture execution speed. Many current approaches either focus solely on spatial features or lack mechanisms for temporal refinement, leading to unstable or inaccurate predictions. Therefore, there is a pressing need for a robust yet computationally efficient framework that can effectively model spatial dependencies and temporal consistency in sign language gestures.

The primary objective of this study is to develop a hybrid architecture that combines Convolutional Neural Networks (CNNs), Transformers, and Hidden Markov Models (HMMs) to achieve accurate and temporally stable sign language recognition.

This work focuses on isolated sign language recognition, where each input represents a single static gesture corresponding to a letter in the ASL alphabet. Unlike continuous sign language recognition, which involves identifying gesture boundaries and temporal transitions in video sequences, isolated recognition simplifies the task to single-frame classification. This approach is appropriate for benchmarking model performance before extending to continuous sign scenarios.

2. Literature Review

In recent years, sign language recognition using deep learning and computer vision has attracted significant attention. In 2018, a hybrid CNN-HMM model [9] was introduced for continuous sign language recognition, which interprets the outputs of the convolutional neural network within a Bayesian framework in an end-to-end manner. The goal of this model was to improve recognition accuracy by combining the strong discriminative capabilities of CNN and the sequential modeling of HMM. Results showed that this method reduced word error rates by 15 to 38 percent and improved

accuracy by up to 20% compared to existing methods across three challenging sign language recognition tasks.

In another study [10], CNN was used to recognize movements from video frames. This model effectively helped translate sign language into text and provided an automated system for communication between individuals proficient in sign language and those who are not. In [11], Lee and Hkaran proposed a new approach for American Sign Language (ASL) training, utilizing a jump motion controller for real-time sign recognition. The system used a combination of LSTM and k-NN to recognize the 26 ASL characters. Authors in [12] proposed the optimization of CNN architectures using the PSO (Particle Swarm Optimization) algorithm for sign language recognition. This research addressed the challenges of optimizing parameters in convolutional neural networks and used this algorithm to find optimal architectures.

Results from three different sign language datasets showed that the optimized architectures achieved high accuracy in sign language recognition. Furthermore, a DeepCNN model [13] was employed for recognizing ASL. Data augmentation techniques were used to increase the size of the training dataset, demonstrating that the DeepCNN model outperformed other methods, particularly in addressing intra-class similarity and the complexities of ASL character recognition.

In [14], researchers developed a deep learning-based method to assist individuals with hearing or language impairments. Their approach used LSTM and GRU techniques to recognize signs in Indian Sign Language (ISL) videos, achieving approximately 97% accuracy for 11 different signs. This model simplified communication for those unfamiliar with sign language, potentially increasing accessibility and inclusion for disabled individuals. Asari and colleagues [15] introduced a hybrid LSTM and deep convolutional neural network model named LRCN for accurate sign language recognition. In this study, the VGG-19 model combined with LSTM showed the highest accuracy at 96.39%. Using BLSTM in this model improved continuous learning, which is crucial for real-time sign language translation.

Also, a sign language recognition model was explored in [16] using CNN models combined with LSTM and Self-Attention. This model, as a new approach for sign language recognition, was capable of extracting spatial features from VGG16 and motion features using Optical Flow. Hybrid optimization algorithms were also employed to improve the accuracy and performance of the model. The results showed that the proposed model achieved over 98.7% accuracy, providing a significant performance improvement over existing models.

One of the new approaches in sign language recognition is the use of deep learning, especially transfer learning. In [17], the authors examined several well-known deep learning models such as VGG16, ResNet50, MobileNetV2, InceptionV3, and CNN using the ASL dataset. The results showed that the InceptionV3 model performed best with an accuracy of 96%. Using pre-trained models and tuning them with sign language data increases recognition accuracy and is very useful in situations where labeled data are limited. These methods provide a strong foundation for developing intelligent sign language recognition systems that could be an effective step in reducing communication barriers for the deaf community.

In order to improve communication between deaf and hearing people, Islam et al [18] developed an intelligent system that combines deep learning and Natural Language Processing (NLP) to identify sign language with high accuracy. In this system, models such as MLP, CNN, and ResNet50v2 were examined for gesture recognition, among which ResNet50v2 had the best performance with 97% accuracy.

The BERT language model was also used to generate automatic responses, and the results indicate its high potential in natural language interaction with users. This approach paves the way for developing chatbots and smart assistants specifically for sign language users and can be an effective step in reducing communication barriers for the deaf community.

In the pursuit of developing lightweight and practical systems for sign language recognition, Zhang et al. [19] proposed an innovative model called the two-path background-removal convolutional neural network (DPCNN). This model, designed based on computer vision, is capable of extracting hand features with high accuracy without the need for expensive sensors and with a simple structure. In this architecture, one path is dedicated to learning general features, while the other path gradually extracts and removes background features to achieve a purer representation of hand gestures. The results show that this model, achieving an accuracy of 99.52%, is highly suitable both in terms of performance and efficiency on small devices.

Several studies have also explored hybrid or ensemble architectures for sign language recognition. More recently, Alkhoraif et al. [20] proposed a Transformer-based ensemble model

with multi-modal input fusion, focusing on word-level recognition. Zhang and Jiang [21] provided a comprehensive review of recent advances in deep learning for sign language recognition, highlighting the ongoing trend toward integrating spatial and temporal modeling techniques. In contrast to these methods, the proposed CNN–Transformer–HMM framework maintains a lightweight design while combining spatial feature extraction, temporal attention, and probabilistic sequence smoothing in a unified architecture.

Overall, previous studies have made substantial progress in sign language recognition through various architectures such as CNNs, RNNs (including LSTMs and GRUs), and Transformer-based models. However, most of these approaches either focus solely on spatial feature extraction or require complex recurrent structures that increase computational cost and training time. CNN-based methods effectively capture local spatial patterns but lack temporal awareness, while RNN-based models handle sequential data but often struggle with long-term dependencies and efficiency. More recent Transformer-based methods improve contextual modeling but typically demand large-scale data and extensive training. In contrast, the proposed CNN–Transformer–HMM framework integrates the strengths of these models by combining CNN-based spatial feature extraction, Transformer-driven temporal attention, and HMM-based sequence smoothing. This hybrid design provides a balanced trade-off between accuracy, temporal consistency, and computational efficiency.

3. Dataset

The Sign Language MNIST dataset is a valuable resource for training and evaluating machine learning models for finger-spelling sign language recognition [22]. It is specifically designed for American Sign Language alphabet letters and follows a classification structure similar to the well-known MNIST dataset; however, instead of numbers, it contains images of ASL hand gestures representing letters.

The dataset includes 28×28 pixel black-and-white images for each letter of the ASL alphabet. These images have been preprocessed in a standardized manner to ensure uniform size and format. The dataset covers 24 letters of the English alphabet, with letters J and Z excluded because they require movement in sign language and are not included in this dataset.

The dataset distribution includes two CSV files containing 27,455 training and 7,172 testing samples. Each record includes 784 pixel values representing a 28×28 grayscale image and one label column corresponding to an ASL letter.

The Sign Language MNIST dataset includes predefined training and testing splits and does not contain subject identifiers. Each record represents an isolated static hand gesture corresponding to a single ASL letter, captured from various anonymous contributors. As the dataset consists solely of static gestures rather than dynamic or continuous sign sequences, subject independence is not applicable. Accordingly, the proposed model performs letter-level classification instead of sentence-level interpretation, which defines the study's focus and experimental scope.

Figure 1 shows the letters of the American Sign Language alphabet, depicted in a static and manual form. These letters form the basis for the design of the Sign Language MNIST dataset. Machine learning models must be able to correctly recognize each of these gestures.

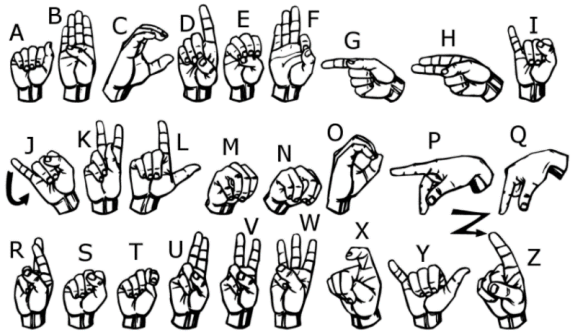

Figure 1. An example of the American Sign Language alphabet.

Figure 2(a) shows a colored sample image representing the sign language letters. The Sign Language MNIST dataset uses this structure to map live images to a simpler format, converting each original image into black and white. A sample of black-and-white images from the Sign Language MNIST dataset is shown in Figure 2(b). These images have been reduced to a 28×28 pixel resolution to make them easier to process in machine learning models.

Before feeding the data into the model, several preprocessing steps were applied to ensure uniformity and enhance the model's generalization ability. All images were normalized and resized to 28×28 grayscale format. To increase the diversity of the training data, augmentation techniques such as random rotation (up to ±15°) and horizontal flipping were applied to simulate natural variations in hand orientation and camera position. No augmentation was applied to the test set to maintain a consistent evaluation process.

This dataset is widely used for developing systems capable of recognizing sign language for various

purposes, including communication for deaf and hard-of-hearing individuals. The advantages of this dataset include standardized preprocessing and a wide range of samples, making it ideal for starting machine vision-related projects. The dataset contains static hand gesture images, each corresponding to a single ASL letter; therefore, no temporal segmentation was required before feeding the data into the model.

(a)

(b)

Figure 2. (a) Color and (b) black and white examples of sign language.

4. Proposed Method

With the aim of designing a high-accuracy model for recognizing hand gestures in sign language, a hybrid model is proposed (Figure 3) that incorporates Convolutional Neural Networks (CNN) for visual feature extraction, Transformers for learning long-term dependencies in the data [23], and Hidden Markov Models (HMM) for smoothing and reducing sudden variations in predictions [24].

As shown in Figure 3, the first step involves loading and preprocessing the data. The Sign Language MNIST dataset is first loaded from the CSV files. This data includes 28×28 images of hand gestures and their corresponding labels. The images are converted into tensors and prepared for use in the neural network model.

In the second step, a CNN is designed to extract spatial and geometric features from sign language images. Conv2D layers are used to apply convolutional filters, and MaxPooling layers are employed to reduce the dimensionality of the features. The extracted features from the CNN are then passed into the Transformer Encoder layers. These layers use self-attention mechanisms to learn temporal dependencies or sequences in the data.

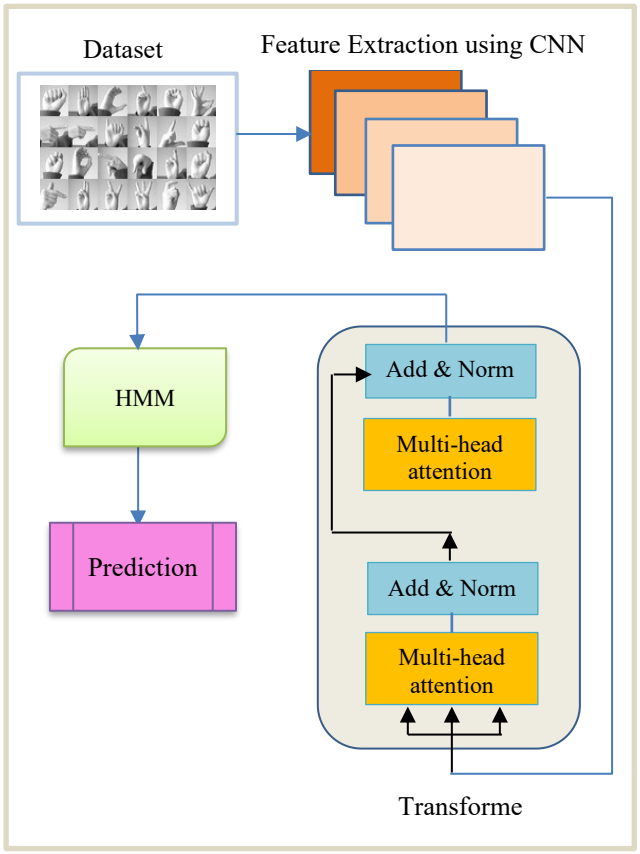

Figure 3. Overall framework of the proposed method.

Following that, an HMM is used for smoothing. The outputs of the CNN and Transformer are fed into the HMM, which smooths the predictions and prevents sudden fluctuations.

The proposed model consists of three main components, which are detailed in the subsections below.

4.1. Feature Extraction using Convolutional Neural Network

In the first step, raw grayscale images are fed into the model and processed through the CNN layers. The CNN is responsible for extracting spatial features from the sign language images. This network applies three convolutional blocks to the input images, each performing a convolution operation to capture local patterns, followed by an activation function, batch normalization to stabilize learning, and pooling operations to downsample the feature maps.

The first block uses 32 filters with a kernel size of 3×3, preserving the spatial dimensions through appropriate padding, and then a MaxPooling layer reduces the image dimension from 28×28 to 14×14. By (1), it performs the convolution operation on the input image I using filter weights $W^{(1)}$ and bias $b^{(1)}$, in order to capturing localized features such as edges and textures.

$$Z^{(1)} = I \times W^{(1)} + b^{(1)} \quad (1)$$

After generating the initial feature maps, the *ReLU* activation function is applied to the convolution output (as in (2)), introducing non-linearity by zeroing out negative values. This enables the network to learn more complex patterns from the feature maps. MaxPool2d reduces the dimensions from 28×28 to 14×14. This pooling operation selects the maximum value in each non-overlapping 2×2 region, reducing spatial dimensions and computational cost while preserving important features. It creates a more compact representation of the features before feeding them into subsequent layers.

$$A^{(1)} = RELU(Z^{(1)}) \quad (2)$$

Blocks 2 and 3 of the CNN follow the same operations as Block 1, taking the previous layer's output features as input. The second block similarly uses 64 filters and another pooling operation, further reducing the size to 7×7. The third block increases the number of filters to 128 and culminates with an adaptive pooling layer that compresses the features to a fixed size of 4×4. This comprehensive CNN structure extracts features such as edges, corners, and textures from the input images, preparing them for subsequent sequence modeling steps.

4.2. Learning Dependencies with Transformer Encoder

After feature extraction by the CNN, the features are passed to the Transformer Encoder. These layers utilize self-attention mechanisms to learn temporal and sequential dependencies. Self-attention allows the model to assign different weights to different parts of the input. This component is particularly useful for understanding complex and long-term dependencies in data sequences [25,26]. In this step, multi-head attention and feed-forward mechanisms are employed to process the data. Details of the operations are explained below.

Once the CNN has extracted the spatial features, the output tensor of size (batch_size, 128, 4, 4) is reshaped into a sequence suitable for capturing long-term and inter-token dependencies. The reshaping operation organizes the 4×4 spatial map into a sequence of 16 tokens, each with 128 features, and a learnable positional encoding $P \in R^{1\times16\times128}$ is added to the token embeddings (as in (3)) to incorporate positional information.

$$X_{pos} = X + P \quad (3)$$

The tokens X_{pos} are then processed through Transformer Encoder layers, which employ multi-head self-attention mechanisms. These mechanisms dynamically weight different parts of the input, allowing the model to capture complex temporal or sequential dependencies in the features.

In multi-head self-attention, the attention for each head is computed using (4). This formula computes self-attention by projecting the input into query (Q), key (K), and value (V) representations, enabling each token to attend to other tokens in the sequence. The *softmax* normalization (scaled by $\sqrt{d_k}$) guarantees that attention weights are properly scaled and sum to 1.

$$Attention(Q,K,V) = soft\max(\frac{QK^T)}{\sqrt{d_k}}) \times V \quad (4)$$

Following the self-attention stage, a feed-forward network further processes the output, and residual connections along with layer normalization are applied to ensure training stability. The processed features are then aggregated into a compact representation that will be used for classification and eventual smoothing by the HMM module.

4.3. Using Hidden Markov Model (HMM) for Smoothing Predictions

In the final stage, the raw predictions produced by the CNN–Transformer network are refined using an HMM to achieve a smoother and more consistent prediction sequence. This module enhances temporal correlation between successive predictions by estimating initial state probabilities and transition probabilities which are derived from the training label sequences.

The HMM computes the posterior probability of each prediction at successive time steps, employing the Viterbi algorithm to infer the optimal label sequence. In this way, HMM models the temporal transition probabilities between predicted classes as follow to reduce sudden changes.

Equation (5) computes the prior probability of label i (π_i) based on its frequency in the training data. It sets the baseline for the prediction sequence by establishing the likelihood of beginning with a given class.

$$\pi_i = \frac{\gamma_{0i}}{N} \quad (5)$$

where γ_{0i} is the frequency of label *i* at the start and *N* is the number of total sequences. Equation (6) calculates the transition probability from state *i* to state *j* (a_{ij}) by counting observed transitions, modeling the likelihood of a switch between classes in the sequential predictions. It is critical for smoothing out abrupt prediction changes.

$$a_{ij} = \frac{\gamma_{ij}}{\gamma_i} \tag{6}$$

where γ_{ij} is the number of transition from state *i* to *j* and γ_i is the number of total transitions from state *i*.

Through the recursive computation of the Viterbi algorithm (as in (7)), the optimal label sequence is determined across all time steps.

$$\delta_t(i) = \max_t[\delta_{t-1}(i) + \log a_{ij}] + \log b_j(O_t) \tag{7}$$

where $\delta_t(i)$ gives the maximum probability of a subsequence of labels and observations until time *t*, being at label *i* at time *t*. By Viterbi algorithm, we can identify the most probable sequence of labels over time.

In this study, the HMM was implemented with three hidden states representing temporal transitions between gesture predictions. The model used a discrete emission distribution derived from the Transformer outputs, with transition and initial probabilities initialized uniformly and optimized through Baum–Welch re-estimation during training.

Unlike deep temporal models such as LSTM or GRU, which are used for learning long-term dependencies from raw sequential data, the Hidden Markov Model (HMM) in our framework is not intended to replace these models. Instead, it acts as a post-processing probabilistic layer that refines and smooths the output predictions generated by the CNN–Transformer network. The rationale for using HMMs lies in their ability to explicitly model state transition probabilities and enforce temporal consistency between successive predictions. This mechanism is particularly effective in reducing abrupt label transitions and noise in static image sequences, while adding minimal computational overhead compared to training an additional recurrent layer.

5. Experiments and Results

All experiments were conducted on a personal computer equipped with an Intel® Core™ i3-4010U CPU @ 1.7 GHz and 16 GB of RAM, without GPU acceleration. The implementation was developed in Python 3.10 using the PyTorch 2.0.1 framework on a Windows 10 operating system. Model training and evaluation were performed under identical conditions for all experiments to ensure consistent and reproducible results.

In this section, we analyze the results of the proposed method and compare them with state-of-the-art approaches, including VGG16 [17], ML-CNN [18], DPCNN [19], LRCN [15], and CCNNSa-LSTM [16], as described in the literature review (Section 2).

To express the results, we first describe the evaluation criteria and how to set the parameters.

5.1. Evaluation Metrics

To evaluate the model's performance, the metrics of accuracy, precision, recall, F-score, and Sign Error Rate (SER) are used, as expressed in (8) to (12). The symbols employed are as follows: TP (True Positive): correctly predicted positive cases; TN (True Negative): correctly predicted negative cases; FP (False Positive): incorrectly predicted positive cases (actual negative); and FN (False Negative): incorrectly predicted negative cases (actual positive).

- Accuracy: It is calculated by dividing the number of correct predictions by the total number of predictions.

$$Acc = \frac{TP + TN}{TP + FN + FP + TN} \tag{8}$$

- Precision: Precision shows the correctness of the model's positive predictions.

$$\mathrm{Pr}ecision = \frac{TP}{TP + FP} \tag{9}$$

- Recall (Sensitivity): This metric indicates the model's ability to correctly identify positive samples.

$$\mathrm{Re}call = \frac{TP}{TP + FN} \tag{10}$$

- F-score: It is a composite metric that assesses the balance between precision and recall. This metric is especially useful when the data are unbalanced.

$$Fscore = 2 \times \frac{\mathrm{Pr}ecision \times \mathrm{Re}call}{\mathrm{Pr}ecision + \mathrm{Re}call} \tag{11}$$

- Sign Error Rate (*SER*): This metric represents the ratio of incorrect predictions to the total number of samples.

$$SER = \frac{FP + FN}{TP + FP + FN + TN} \tag{12}$$

5.2. Parameter Setting

To achieve optimal performance, we carefully tuned the model's hyperparameters through an iterative experimental process. The configuration

shown in Table 1 reflects a balanced trade-off between computational efficiency and representational capacity. A batch size of 64 and 20 training epochs ensured stable convergence without overfitting. The CNN stack, consisting of three convolutional layers, was designed to extract low- and mid-level spatial features prior to Transformer encoding. The Transformer component uses three encoder layers with four attention heads and a 256-dimensional feed-forward network, providing sufficient modeling power for capturing global relationships while maintaining a lightweight architecture. A dropout rate of 0.1 was applied to reduce overfitting. These hyperparameters were selected after iterative experimentation, where higher-capacity configurations yielded marginal gains at the cost of significantly increased computation, confirming that the chosen setup provides an optimal balance for the Sign Language MNIST dataset. The model is trained using the cross-entropy loss function, as shown in (13), with the aim of improving prediction accuracy by minimizing the loss.

$$Loss = -\frac{1}{N}\sum_{i=1}^{N}\sum_{c=1}^{C} y_{ic} \log p_{ic} \tag{13}$$

where N is the numbers of samples, C denotes the number of classes, y_{ic} represents the true label (ground truth) of the i-th sample (1 for the correct class and 0 otherwise), and p_{ic} is the predicted probability that sample i belongs to class c. Also, Adam optimizer is applied to adjust the weights and improve the model's performance.

5.3. Results and Discussion

The performance of the proposed CNNTRHMM model was analyzed in three phases: learning progression, comparative evaluation with state-of-the-art architectures, and confusion matrix inspection. Each phase demonstrate the model's superior performance compared to contemporary methods on the ASL MNIST benchmark.

The learning behavior of the proposed CNNTRHMM model was first examined through accuracy and loss curves over 20 epochs. As illustrated in Figures 4-5, the training accuracy steadily increased, reaching over 99.5% by the final epoch, while the test accuracy followed a similar trajectory, stabilizing around 98.9%. Simultaneously, the training loss demonstrated a consistent downward trend, indicating smooth convergence, whereas the test loss exhibited moderate oscillations, particularly between epochs 10 and 15. These fluctuations reflect minor generalization instability, which was effectively addressed through the integration of a post-processing HMM smoothing module. These dynamics, clearly depicted in Figure 4, confirm that the model did not suffer from overfitting and achieved generalizable learning performance across unseen samples.

To evaluate the effectiveness of the proposed model, a set of baseline and state-of-the-art methods were compared using multiple evaluation metrics. Table 2 summarizes the results in terms of accuracy, precision, recall, F-score, and Sign Error Rate (SER). These complementary metrics were included to provide a more comprehensive and reliable performance profile, ensuring that the evaluation captures both correctness (accuracy) and class-wise discrimination capability (precision, recall, and F1-score). The first group includes models based solely on convolutional architectures, such as VGG16 and ML-CNN. While effective in static spatial feature extraction, these models failed to capture temporal relationships between gesture components.

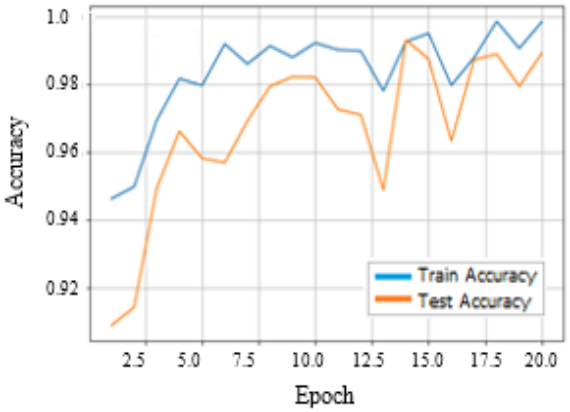

Figure 4. Accuracy curves for training and test sets.

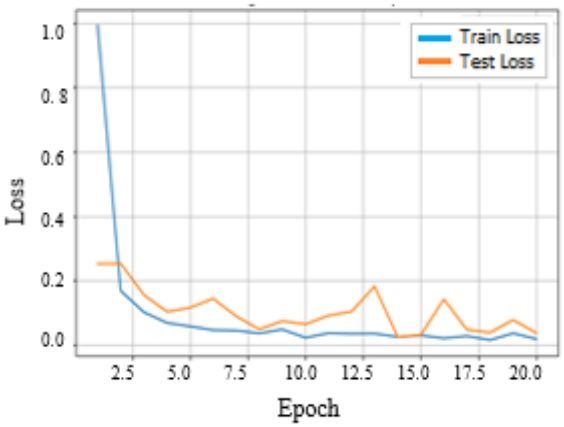

Figure 5. Loss curves for training and test sets.

Table 1. Hyperparameter of the proposed model.

Hyperparameter	Batch size	Epoch	Learning rate	CNN Layers	Transformer Layers	Dropout rate	Attention head	Feed-forward dim
Values	64	20	0.001	3	3	0.1	4	256

Notably, VGG16 achieved 86% accuracy (SER=0.40), whereas ML-CNN attained a modest improvement (88%, SER=0.32). However, both models frequently misclassified visually similar letters (e.g., 'D' and 'E'), highlighting the inherent limitation of static spatial methods in dynamic sign interpretation. Simultaneously, DPCNN's dual-path CNN architecture enhances hand feature representation through iterative global and background feature subtraction, achieving superior accuracy to VGG16 and ML-CNN methods.

The second category consists of hybrid architectures that integrate basic temporal modeling approaches with spatial feature extraction, leading to superior performance by leveraging both spatial and temporal dependencies within a unified framework.

In this regard, LRCN surpasses baseline methods (91% accuracy, 0.27 SER) by combining CNN's spatial processing with LSTM's temporal modeling. Despite this improvement, the model lacked sufficient capacity to model long-range temporal dependencies, resulting in instability in predictions over time.

Meanwhile, attention-based models demonstrated superior capabilities. The CCNNSa-LSTM model, which integrates LSTM with self-attention layers, achieved an accuracy of 98.7% and an SER of 0.114. Its attention mechanism enabled better focus on salient parts of the input sequence, improving performance, especially in ambiguous or noisy samples. Nonetheless, some inconsistencies were still observed in sequences where gesture transitions were abrupt.

In contrast, our proposed hybrid model, combining CNN, Transformer, and HMM, demonstrated superior performance across all evaluation metrics. The Transformer encoder provided a robust mechanism for capturing long-range dependencies across gesture sequences, significantly enhancing the consistency of classification results. Without the HMM, the model achieved an accuracy of 98.93% and an SER of 0.0107, already surpassing all baselines. The incorporation of the HMM further improved classification reliability by smoothing transitions and reducing erratic misclassifications, resulting in a final accuracy of 99.02%, an F1 score of 0.9903, and an exceptionally low SER of 0.0098. This marginal yet crucial improvement demonstrates the value of probabilistic post-processing in gesture recognition tasks, especially for mitigating noise and temporal irregularities.

It is worth noting that the comparative results reported in Table 2 implicitly represent an ablation analysis between the CNN–Transformer baseline and the full CNN–Transformer–HMM architecture. The addition of the HMM module yields a measurable improvement in both accuracy and Sign Error Rate while maintaining a lightweight model. Several hyperparameter settings were also tested during experimentation to ensure an optimal balance between model accuracy and computational efficiency.

To further support the effectiveness of the proposed model, Figures 6 present a comparative bar chart summarizing the performance metrics (Accuracy, Recall, Precision, and F1-score) alongside SER of several baseline and state-of-the-art models.

Table 2. Comparison the models on the American Sign Language (ASL) dataset.

Model	Accuracy (%)	Precision (%)	Recall (%)	F-score (%)	SER
VGG16 [17]	86	84	85	83	0.40
ML-CNN [18]	88	87	86	87.5	0.32
DPCNN [19]	90	89	87	88	0.30
LSTM-CNN [14]	91	90	89	89.5	0.27
CCNNSa-LSTM [16]	98.7	98.5	98.2	98.6	0.11
CNNTR*	98.93	99.02	98.93	98.93	0.0107
CNNTRHMM**	**99.02**	**99.10**	**99.01**	**99.03**	**0.0098**

* Proposed Method without HMM, ** Proposed Method with HMM

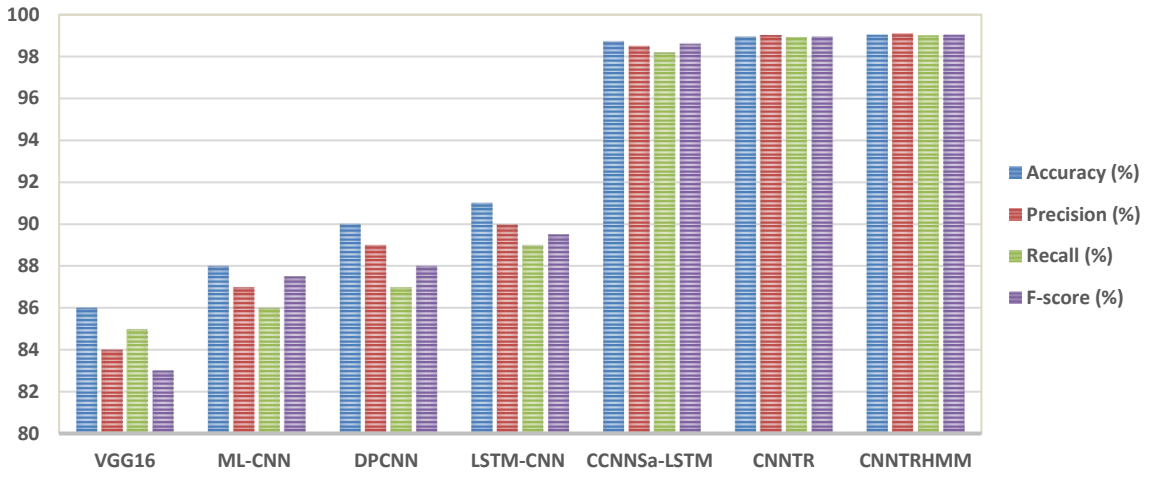

Figure 6. Comparative evaluation of the models based on accuracy, precision, recall and F-score metrics.

According to Figure 6, the proposed method with HMM achieves the highest performance across all metrics, each exceeding 99%. This surpasses all other evaluated models, including CCNNSa-LSTM [16], which performs closely but slightly below (above 98%). Convolutional-based architectures such as VGG16 and ML-CNN show noticeably lower performance, highlighting the limitations of using either spatial or shallow sequential models alone.

Figure 7 visualizes the Sign Error Rate (SER) across all models. Here, the Proposed Method with HMM again outperforms the others by achieving a remarkably low SER of 0.0098, demonstrating not only high accuracy but also extremely low prediction inconsistency. In contrast, the VGG16 model exhibits the highest SER, indicating its vulnerability to temporal noise and difficulty in modeling long-term dependencies.

These combined results validate that the proposed hybrid framework, integrating CNN, Transformer, and HMM, not only boosts classification accuracy but also ensures temporal coherence and robustness, making it a reliable solution for real-world sign language recognition systems.

Beyond accuracy, the proposed hybrid model was designed to remain computationally lightweight. The CNN–Transformer backbone allows for efficient parallel processing during feature extraction and sequence modeling, while the HMM operates only as a simple post-processing step with negligible computational overhead. Compared to larger transformer-based or recurrent architectures, the proposed model maintains a much lower parameter count and faster inference time. This balance between performance and efficiency ensures that the system can be deployed in real-time or resource-limited environments without significant computational costs.

To gain deeper insights into class-specific model performance, we conducted a detailed analysis of the final model's confusion matrix (Figure 8), revealing patterns of correct classifications and systematic errors across gesture categories.

The matrix reveals near-perfect classification across all 25 static ASL letters. Several classes, including 'A', 'B', 'L', and 'Y', were recognized with 100% precision, indicating robust representation learning and decision boundaries. However, a few classes exhibited mild confusion. Notably, the letter 'T' (label 17) was misclassified 19 times, predominantly as 'S' or 'M', likely due to overlapping visual characteristics. Similarly, ‘N’ and ‘M’ were occasionally confused, with the former misclassified as the latter in three instances. These misclassifications were infrequent and mostly occurred in edge cases where visual overlap was substantial. Importantly, the post-HMM smoothing led to a marked reduction in sporadic errors and a cleaner diagonal pattern in the matrix, indicating better sequence-level coherence in predictions.

Despite the minor inconsistencies, the dense diagonal structure of the matrix reflects a well-discriminating model. Qualitative behavior of the model can be inferred from the confusion matrix, where strong diagonal dominance indicates consistent recognition success, and sparse off-diagonal entries highlight rare misclassifications between visually similar signs.

Each component of the CNN–Transformer–HMM architecture contributes uniquely to the model’s performance. The CNN layers extract low- and mid-level spatial features from the hand gesture

images, capturing edges, contours, and textures. The Transformer encoder captures long-range dependencies among these extracted features, allowing the model to better understand global spatial relationships. Finally, the HMM layer smooths the prediction sequence, reducing sporadic errors and ensuring consistency across neighboring frames.

During the optimization process, excluding either the Transformer or the HMM component led to a measurable drop in accuracy, demonstrating their complementary roles in enhancing recognition reliability.

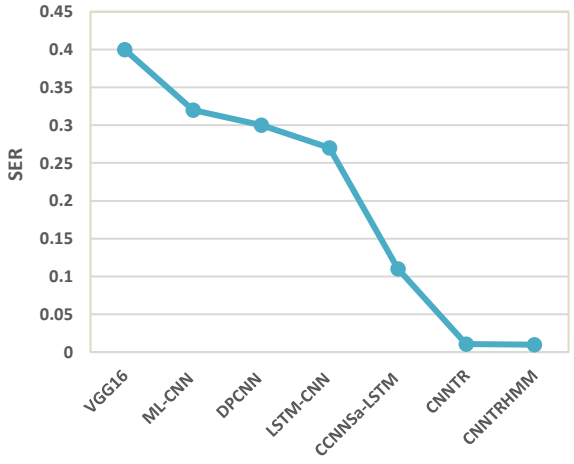

Figure 7. Comparative evaluation of the models based on SER metric.

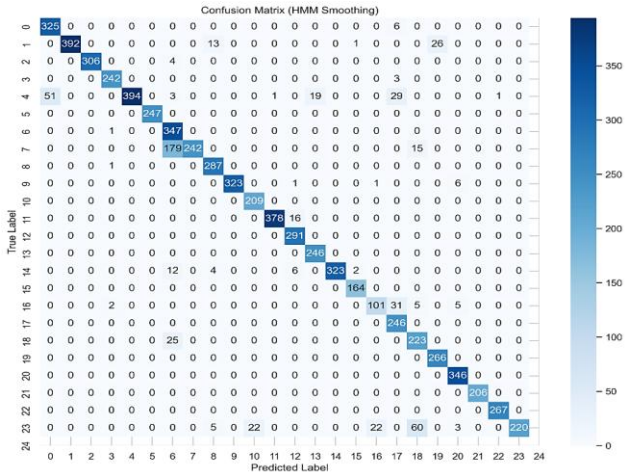

Figure 8. Confusion Matrix of the proposed model on test set.

In summary, the proposed model demonstrates outstanding capability in recognizing static sign language gestures, with significant improvements in both accuracy and prediction consistency over prior methods. Its layered design, combining CNNs for spatial learning, Transformers for long-term sequence modeling, and HMMs for probabilistic refinement, proves to be a highly effective architecture for real-world sign language recognition.

6. Conclusion

This study introduced a hybrid framework that integrates CNNs, Transformers, and HMMs for effective sign language recognition. The experimental analysis demonstrated that the HMM layer enhances prediction stability by reinforcing sequential consistency, while the Transformer component successfully captures long-range dependencies within feature sequences. Together, these elements contribute to the model's strong performance, achieving 99% accuracy and a sign error rate of 0.0098%.

Building on these promising results, several future directions can enhance the impact of this framework. Since the Sign Language MNIST dataset contains standardized static images, signer variation in terms of hand shape, background, or motion could not be analyzed. Future work will address this limitation using more diverse and dynamic datasets. Further developments may include real-time sign interpretation, multilingual gesture recognition, and adaptive human–computer interaction. Advancing the model in these areas can promote more inclusive technologies for deaf and hard-of-hearing users and support more natural multimodal communication.

Ethical Considerations and Data Availability

This study was conducted using the publicly available Sign Language MNIST dataset, which is an open-access benchmark designed for academic and non-commercial research. The dataset contains grayscale images of static hand gestures representing letters from the American Sign Language (ASL) alphabet and does not include any personally identifiable information.

References

[1] D. M. Cochran and S. L. Koster, "Challenges in automatic sign language recognition: A survey," *Journal of Signal Processing*, Vol. 44, No. 5, pp. 812-825, 2020.

[2] X. Zhang and Z. Liu, "Deep learning methods for sign language recognition," *IEEE Transactions on Neural Networks and Learning Systems*, Vol. 30, No. 4, pp. 1224-1237, 2019.

[3] S. Jordan and J. Hawkins, "A review of machine learning techniques for sign language recognition," *International Journal of Machine Learning and Computing*, Vol. 9, No. 3, pp. 367-374, 2018.

[4] K. Lee and S. Choi, "Hand gesture recognition for sign language using convolutional neural networks," *Pattern Recognition Letters*, Vol. 142, pp. 1-9, 2021.

[5] N. Mohamed, M. B. Mustafa and N. Jomhari, "A review of the hand gesture recognition system: Current progress and future directions," *IEEE access*, Vol. 9, pp. 157422-157436, 2021.

[6] J. Shin, A. S. M. Miah, M. H. Kabir, M. A. Rahim and A. Al Shiam, "A methodological and structural review of hand gesture recognition across diverse data modalities," *IEEE Access*, 2024.

[7] J. Qi, L. Ma, Z. Cui and Y. Yu, "Computer vision-based hand gesture recognition for human-robot interaction: a review," *Complex & Intelligent Systems*, Vol. 10, No. 1, pp. 1581-1606, 2024.

[8] A. O. Hashi, S. Z. M. Hashim and A. B. Asamah, "A Systematic Review of Hand Gesture Recognition: An Update From 2018 to 2024," *IEEE Access*, 2024.

[9] O. Koller, S. Zargaran, H. Ney and R. Bowden, "Deep sign: Enabling robust statistical continuous sign language recognition via hybrid CNN-HMMs," *International Journal of Computer Vision*, Vol. 126, pp. 1311-1325, 2018.

[10] R. S. Sabeenian, S. S. Bharathwaj and M. M. Aadhil, "Sign language recognition using deep learning and computer vision," *Journal of Advanced Research in Dynamical and Control Systems*, Vol. 12, No. 5, pp. 964-968, 2020.

[11] C. K. Lee et al., "American sign language recognition and training method with recurrent neural network," *Expert Systems with Applications*, Vol. 167, 2021.

[12] J. Fregoso, C. I. Gonzalez and G. E. Martinez, "Optimization of convolutional neural networks architectures using PSO for sign language recognition," *Axioms*, Vol. 10, No. 3, pp. 139, 2021.

[13] A. Mannan, A. Abbasi, A. R. Javed, A. Ahsan, T. R. Gadekallu and Q. Xin, "Hypertuned Deep Convolutional Neural Network for Sign Language Recognition," *Computational intelligence and neuroscience*, Vol. 2022, No. 1, 2022.

[14] D. Kothadiya et al, "Deepsign: Sign language detection and recognition using deep learning," *Electronics*, Vol. 11, No. 11, pp. 1780, 2022.

[15] M. A. Asari, N. A. Jasmin Sufri and G. Si Qi, "Emergency sign language recognition from variant of convolutional neural network (CNN) and long short term memory (LSTM) models," *International Journal of Advances in Intelligent Informatics*, Vol. 10, No. 1, 2024.

[16] A. Baihan, A. I. Alutaibi, M. Alshehri, M. and S. K. Sharma, "Sign language recognition using modified deep learning network and hybrid optimization: a hybrid optimizer (HO) based optimized CNNSa-LSTM approach," *Scientific Reports*, Vol. 14, No. 1, 2024.

[17] S. Mohsin, B. W. Salim, A. K. Mohamedsaeed, B. F. Ibrahim and S. R. Zeebaree, "American sign language recognition based on transfer learning algorithms," *International Journal of Intelligent Systems and Applications in Engineering*, Vol. 12, No. 5, pp. 390–399, 2024.

[18] N. Aslam, K. Abid and S. Munir, "Robot assist sign language recognition for hearing impaired persons using deep learning," *VAWKUM Transactions on Computer Sciences*, Vol. 11, No. 1, pp. 245–267, 2023.

[19] J. Zhang et al., "Sign language recognition based on dual-path background erasure convolutional neural network," *Scientific Reports*, Vol. 14, No. 1, 2024.

[20] ALKHORAIF, A. A., ALSULAIMAN, M., ABDUL, W., & BENCHERIF, M. (2025). Ensemble Transformer-based Word-Level Sign Language Recognition with Multi-Modal Input Fusion. Journal of Engineering Research.

[21] Zhang, Y., & Jiang, X. (2024). Recent Advances on Deep Learning for Sign Language Recognition. Computer Modeling in Engineering & Sciences (CMES), 139(3).

[22] Sign Language MNIST, Kaggle, Available: https://www.kaggle.com/datamunge/sign-language-mnist, 2017.

[23] Y. Zhao et al., "TRanspose: Towards Understanding and Simplifying Transformers in Computer Vision," *in Neural Information Processing Systems*, NeurIPS, 2021.

[24] L. R. Rabiner, "A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition," in Proceedings of the IEEE, Vol. 77, No. 2, pp. 257-286, 1989.

[25] R. Hosseinzadeh and M. Sadeghzadeh, "Attention Mechanisms in Transformers: A General Survey," *Journal of AI and Data Mining*, Vol. 13, No. 3, pp. 359-368, 2025.

[26] M. Allahgholi , H. Rahmani and P. Soltanzadeh, "ConSPro: Context-Aware Stance Detection Using Zero-Shot Prompting," *Journal of AI and Data Mining*, Vol. 13 ,No. 2, pp. 251-260, 2025

تشخیص زبان اشاره با استفاده از مدل ترکیبی مبتنی بر شبکه‌های عصبی کانولوشنی و مدل‌های پنهان مارکوف

ملیحه دانش* و زهرا احمدی

گروه مهندسی کامپیوتر، دانشگاه علم و فناوری مازندران، بهشهر، ایران.

ارسال ۲۰۲۵/۰۷/۰۲؛ بازنگری ۲۰۲۵/۱۱/۰۲؛ پذیرش ۲۰۲۵/۱۲/۱۹

چکیده:

در سال‌های اخیر، تشخیص زبان اشاره به یکی از چالش‌های مهم در حوزه‌های پردازش تصویر و یادگیری ماشین تبدیل شده است. افراد دارای اختلال شنوایی برای برقراری ارتباط از زبان اشاره استفاده می‌کنند، اما نبود ابزارهای خودکار برای ترجمه آن، موانع ارتباطی قابل‌توجهی ایجاد کرده است. در این پژوهش، یک مدل ترکیبی مبتنی بر شبکه‌های عصبی کانولوشنی (CNN)، ترنسفورمرها و مدل‌های پنهان مارکوف (HMM) برای شناسایی دقیق حرکات زبان اشاره با استفاده از مجموعه‌داده زبان اشاره MNIST ارائه شده است. در این مدل، ابتدا ویژگی‌های تصاویر دست‌نویس توسط شبکه‌های CNN استخراج می‌شود؛ سپس این ویژگی‌ها به مدل ترنسفورمر وارد می‌شوند تا وابستگی‌های پیچیده و بلندمدت موجود در توالی ویژگی‌ها پردازش شوند. در گام بعد، به‌منظور هموارسازی پیش‌بینی‌ها و بهبود دقت، از مدل پنهان مارکوف استفاده می‌شود که با در نظر گرفتن توالی‌های قبلی، پیش‌بینی‌های نهایی را تنظیم می‌کند. نتایج نشان می‌دهد مدل پیشنهادی با به‌کارگیری HMM به دقت ۹۹ درصد و نرخ خطای اشاره ۰.۰۰۹۸ دست یافته است که بیانگر کارایی بالای آن در تشخیص حرکات دست است. این پژوهش گامی مهم در جهت توسعه ابزارهای کمکی برای ناشنوایان و ارتقای تعامل انسان با سامانه‌های هوشمند به‌شمار می‌رود.

کلمات کلیدی: زبان اشاره، شبکه‌های عصبی کانولوشنی، مدل‌های پنهان مارکوف، تشخیص.