International Journal of Machine Learning and Cybernetics (2026) 17:192
https://doi.org/10.1007/s13042-025-02962-w

ORIGINAL ARTICLE

Empowering assisted living: ultra leap motion and deep learning for sign language recognition

Asma Khan[1] · Geon-Hee Lee[2] · L. Minh Dang[3,4,5] · Samee Ullah Khan[6] · Muhammad Attique Khan[7] · Woong Choi[8] · Hyeonjoon Moon[1]

Received: 5 November 2024 / Accepted: 22 September 2025

Abstract
Sign language recognition (SLR) is crucial for improving communication and accessibility for people who are deaf or hard of hearing, particularly in our aging population, which increasingly requires healthcare and inclusive environments. Current SLR systems face significant challenges, such as dynamic hand gestures, real-time boundary detection, variable lighting conditions, complex backgrounds, and a lack of diverse real-world datasets. To overcome these challenges, we propose a patch-based network (PBN) that effectively leverages features from various channel patches to handle complex sign language gestures. In addition, a new SLR dataset has been created using ultra leap motion technology that contains 7800 samples related to 26 different classes with a resolution of (224 × 224 pixels). In addition, it offers various contextually relevant information useful in health-oriented domains. Comprehensive experiments are conducted in terms of ablation studies for optimal module selection, showing a remarkable performance of 97% on the ASL-A dataset and 98% on the Massey dataset. These SLR developments not only improve communication for people with disabilities but also improve their overall quality of life and independence, highlighting the critical role of technology in supporting their health and well-being.

Keywords Computer vision · Motion recognition · Pattern recognition · Artificial intelligence · Gesture recognition

✉ Hyeonjoon Moon
hmoon@sejong.ac.kr

Asma Khan
Asmakhan28@sju.ac.kr

Geon-Hee Lee
NOT.Samsung@sju.ac.kr

Samee Ullah Khan
samee.khan@ku.ac.ae

Muhammad Attique Khan
attique.khan@ieee.org

Woong Choi
wchoi@kangnam.ac.kr

[1] Department of Computer Science and Engineering, Sejong University, Seoul 05006, Republic of Korea

[2] Department of Artificial Intelligence, Sejong University, Seoul 05006, Republic of Korea

[3] Institute of Research and Development, Duy Tan University, 550000, Da Nang, Viet Nam

[4] Faculty of Information Technology, Duy Tan University, 550000, Da Nang, Viet Nam

[5] Department of Information and Communication Engineering and Convergence Engineering for Intelligent Drone, Sejong University, Seoul 05006, Republic of Korea

[6] Advanced Research and Innovation Center (ARIC), Khalifa University of Science and Technology, Abu Dhabi, United Arab Emirates

[7] Department of Artificial Intelligence, College of Computer Engineering and Science, Prince Mohammad Bin Fahd University, Al Khobar, Saudi Arabia

[8] College of ICT Construction and Welfare Convergence, Kangnam University, Yongin-Si 16979, Republic of Korea

Published online: 16 March 2026

1 Introduction

In recent years, human-computer interaction (HCI) has grown significantly due to rapid progress in computational techniques, information processing, communication networks, hardware infrastructure, and related sectors [1, 2]. As user expectations become more sophisticated, technology companies are increasingly investing in information management and HCI research, particularly in areas such as healthcare and assisted living [3], where the percentage of individuals requiring support and a secure environment is steadily rising. These investments are essential for enhancing user experiences, fostering innovation, and preserving a competitive edge in a constantly evolving industry [4]. Improving communication for individuals who primarily use sign language is a key area of research. As the demand for healthcare solutions and assisted living technologies grows in an aging society, sign language users also require technological advancements to support them in medical and social contexts [5]. In social, educational, personal, and media contexts, the community relies strongly on sign language for communication. Figure 1 illustrates the proposed ASL recognition framework, integrating a hand detection model (HDM), a cropped hand (CH), and a sign language model (SLM) to enable accurate and efficient recognition. Even with these developments, there is still a long way to go before SLR can be used to facilitate communication in both everyday and healthcare settings, including assisted living environments. Therefore, creating user-friendly tools for sign language translation is vital for both deaf individuals and their social connections, as well as for their access to essential healthcare and assisted living services [6].

Based on the core hardware, SLR technology can be classified into two main types: camera-based gesture recognition and data glove-based gesture recognition. The advantages of glove-based SLR technology include high detection rates, resistance to interference, and exact data collection. However, it involves costly devices and requires users to wear data gloves and supervision when communicating. In contrast, camera-based SLR technology eliminates the need for specialized equipment and provides a more natural interface between humans and computers by using computer vision techniques [7]. Nevertheless, it is highly sensitive to environmental conditions, necessitating continuous research to enhance computational approaches and mitigate these limitations [8]. In interpersonal communication, gestures serve as a form of nonverbal language, often complementing verbal communication. For the deaf and mute community, gestures are the primary mode of interaction with others [9, 10]. Deep learning, a rapidly advancing field in information processing, has made significant strides in machine vision and natural language[11] understanding by aiming to replicate the complex functions of the human brain through multi-layered neural networks [12]. However, the application of deep learning algorithms for gesture recognition is still relatively limited in performance and poses the following major challenges that need to be overcome.

Major challenges:

- Mainstream approaches often employ well-established CNN models, where the entire image is processed by the network to extract features. These models apply filters of varying sizes to capture information related to the region of interest. However, effectively learning these patterns can be challenging due to the presence of both foreground and background elements, particularly when the target object is small relative to the surrounding irrelevant pixel information.
- Existing SLR systems are limited by several factors, including the variability of dynamic hand gestures, difficulties in real-time gesture boundary detection, fluctuations in lighting conditions, and the presence of complex and cluttered backgrounds. Additionally, the lack of diverse, real-world datasets impedes the system's ability to generalize across different environments and users, further limiting performance and precision.

Accomplishing these objectives will contribute substantially to the progress of American Sign Language–based recognition and communication technology for the deaf and hard-of-hearing community and serve as a basis for further studies in gesture recognition systems. The main contributions of this study include:

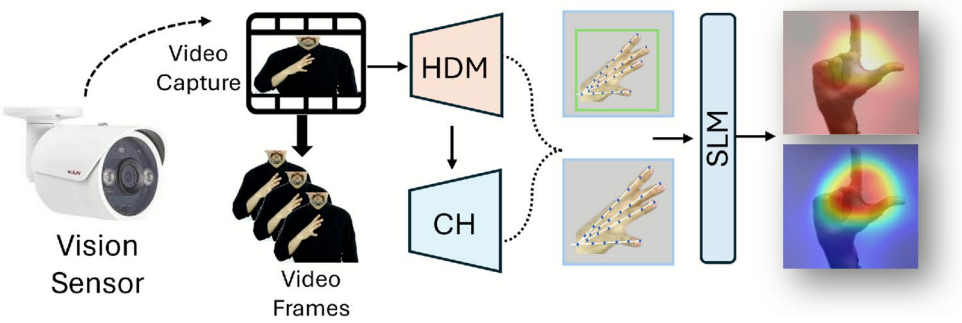

Fig. 1 Overview of the proposed network for ASL recognition, integrating hand detection model (HDM), cropped hand (CH), and sign language model (SLM). Illustrates sequential information processing for accurate ASL recognition

Main contributions:

- We introduce the PBN framework, specifically designed for American Sign Language (ASL) recognition, embedding cross-patch attention instead of the standard soft attention in vision transformers. This allows the model to focus on relevant foreground information while suppressing irrelevant components. Additionally, advanced channel processing enhances feature learning, improving classification accuracy.
- We built a comprehensive ASL alphabet dataset using ultra leap motion in a Unity-based virtual reality environment, capturing high-resolution hand gestures. This dataset accurately reproduces the complexity and variation of ASL signs, providing a valuable resource for future research and facilitating scalable dataset generation.
- We conducted experiments comparing different CNN-based architectures and the proposed transformer-based PBN. Ablation studies and performance analyses demonstrate the effectiveness of our approach. Furthermore, we evaluated the system's practical applicability, highlighting its potential for real-world ASL recognition tasks.

Research objectives

- To create robust solutions for addressing challenges in ASL recognition through the application of deep learning, computer vision, and advanced image processing techniques. By synthesizing these artificial intelligence technologies, the aim is to significantly improve the accuracy and operational efficiency of image-based ASL interpretation systems within the context of assisted living applications.
- To enhance the attention module of the current vision transformer network for ASL recognition. This refinement aims to prioritize the foreground elements of sign language gestures, thereby reducing the influence of redundant or extraneous information on the model's performance in scenarios involving healthcare and rehabilitation.
- To create a comprehensive and high-quality collection of ASL alphabet gestures, captured using UltraLeap devices, to enable rigorous evaluation and comparison of various deep learning models for improved ASL recognition accuracy and performance that supports effective communication in assisted living environments.

The rest of the paper is organized as follows: Sect. 2 is devoted to a deep review of state-of-the-art SLR approaches. Then, the technical details of the proposed framework are provided in Sect. 3, while Sect. 4 outlines the experimental setup, including datasets and ASL assessment metrics. Finally, Sect. 5 concludes with some final remarks and mentions future directions.

2 Literature review

Gesture recognition research has advanced significantly through the integration of vision-based techniques with artificial intelligence methodologies, resulting in enhanced accuracy and efficiency in gesture interpretation. Researchers have tested different structures, including Inflated 3D and CNN, to accurately identify gestures in sign languages such as ISL, ASL, and Turkish. Despite progress, challenges remain in standardizing datasets and developing real-time systems, highlighting the necessity for realistic solutions in real-world scenarios. Additional information about the literature review can be found in the sections below.

3 Deep learning approach for SLR

In recent years, major improvements have been made in SLR through the application of deep learning methodologies, making significant progress in improving accessibility for people with disabilities. Kumar al. [13] have focused on several aspects of SLR, such as integrating deep learning with text-to-speech technology to improve the identification of ASL. Convolutional neural networks (CNNs) are explored for the feature extraction and classification, where they achieved 80% accuracy. In comparison, comparably, Aly et al. [14] proposed a Hybrid Transformer-CNN model for ASL gesture recognition, combining CNN-based local feature extraction with a Vision Transformer for global context. The model achieves 99.97% accuracy on the ASL Alphabet dataset, runs at 110 FPS, and is computationally efficient (5.0 GFLOPs), using feature fusion and advanced augmentation to enhance robustness. For instance, Carneiro et al. [15] developed a relatively inexpensive SLR system that uses handcrafted descriptors in conjunction with a deep learning architecture, achieving an improvement of 7.96% in accuracy on the AUTSL dataset. Kumar et al. [16] proposed a VGG16-based CNN with an attention mechanism for Indian Sign Language recognition, achieving high accuracy (99.8%) in real-time without using gloves or external sensors, thereby facilitating efficient communication for hearing-impaired users.. Similarly, Khan et al. [17] proposed a Hybrid Efficient Convolution (HEC) model combining EfficientNet-B3 with custom dense layers for isolated dynamic Bangla sign language recognition. The model achieves 93.17% accuracy on a 6000-video dataset while handling cluttered backgrounds and illumination variations

efficiently. Seong et al. [18] enhances sign language translation performance by combining transformers with 3D-CNN to improve recognition accuracy, demonstrating comparable efficiency to existing models using the PHOENIX-Weather-2014T dataset. Table 1 provides an overview of these mainstream approaches for gesture recognition, summarizing the methods, accuracies, and key contributions discussed above. A three-stream hybrid model for dynamic hand gesture recognition. Rahim et al. [19] utilized deep learning to improve the detection and classification of sign language gestures. Furthermore, the augmentation of sign language poses through deep learning models, particularly focusing on specific body parts, has been explored by Kim et al. [20] to enhance recognition. The development of all- encompassing datasets designed for languages such as Bengali, as well as their integration into recognition models, Khan et al. [17] demonstrated a growing interest in the development of all-encompassing systems. While these advances are remarkable, problems remain, such as the need for larger datasets and the integration of complex inputs. These challenges indicate potential areas for further research. Beyond vision-based techniques, recent studies have explored alternative input modalities such as surface electromyography (sEMG), which captures muscle activity signals for gesture classification. Rezaee et al. [21] proposed a hybrid deep learning framework combining BiLSTM with metaheuristic optimization and a U-Net-MobileNetV2 encoder, achieving an average accuracy of 90.23% across six datasets. Similarly, Singh and Chaturvedi et al. [22] developed a machine learning pipeline using EMG sensors and ensemble feature selection to recognize American Sign Language gestures, reaching up to 99.91% accuracy. These biosignal-based approaches offer promising alternatives in scenarios where visual data may be unreliable due to occlusion, lighting variability, or privacy constraints. Incorporating such modalities could enhance the adaptability and precision of future SLR systems. These problems can be the result of the advanced accuracy and efficiency of SLR systems, hence developing their availability to a larger user population.

Furthermore, to address recent developments in hand gesture recognition, several notable contributions should be highlighted. Cheok et al. [23] offered a foundational review outlining gesture recognition systems across multiple stages, including data acquisition, segmentation, feature

Table 1 An overview of mainstream approaches for gesture recognition

Refs.	Method	Accuracy	Key contributions	Refs.	Method	Accuracy	Key contributions
[13]	CNN for feature extraction, classification	80.0%	CNN-based spatial feature extraction and classification	[14]	Hybrid CNN Transformer	99.97%	Combines CNN for local features and Transformer for global context; feature fusion with element-wise multiplication; efficient (5 flFLOPs, 110 FPS); robust via contrastive learning and domain adaptation
[16]	Vgg16 CNN with attention mechanism for ISL classification	99.8% (with at- tention), 97.5% (without atten- tion)	Achieves high accuracy in real-time ISL recognition without sensors or gloves; leverages transfer learning and attention mechanism to enhance performance	[18]	3D CNN integrated with Vision Transformer	93%	Application of vision transformers to enhance sign language translation performance
[15]	Hybrid CNN–RNN with handcrafted triangle features	7.96%	Low-cost system combining handcrafted features with deep learning	[20]	CNN with pose-augmented training using domain knowledge	93%	Improved accuracy through targeted pose augmentation
[17]	Hybrid Efficient Convolution (HEC) combining EfficientNet-B3 with custom dense layers for dynamic Bangla Sign Language recognition	93.17%	High accuracy; robust to clutter and illumination variations	[28]	CNN for real-time Bengali Sign Language digit recognition (custom feed-forward CNN with Conv2D, MaxPooling, Flatten, Dense, Dropout layers)	99.75% (with rotation), 94.17% (without rotation)	Achieves high real-time recognition accuracy for Bengali digit gestures; image rotation improves performance; uses binary and grayscale images
[29]	DeepSign: Deep learning-based model using sequential LSTM and flRU layers (1 LSTM + 1 flRU)	97%	Detects and recognizes ISL gestures from video frames; feedback-based sequential learning enhances recognition; effective on 11 different signs	[30]	Leap Motion input + Euclidean Distance with k-NN	95.0%	High-accuracy recognition system for Javanese sign scripts

extraction, and classification, and emphasized existing challenges in real-world generalization. Dey et al. [24] proposed an attention-driven C3D-BiLSTM model that specifically targets Wh-question gestures in video streams, incorporating multi-head attention to capture intricate spatial-temporal cues, achieving state-of-the-art results on the AQSVd dataset. Additionally, Dey et al. [25] developed an attention-based DC-GRU model for recognizing umpire signals in cricket, demonstrating the domain flexibility of such architectures in complex visual environments. These works align with the core motivation of our proposed PBN to enhance recognition precision through attention-based and patch-wise feature extraction. While recent studies such as [26] and [16] have advanced real-time SLR using deep learning, they primarily rely on traditional 2D visual input and do not leverage depth-based motion tracking. These limitations affect spatial understanding and generalization in complex environments. Our work addresses this gap by integrating ultra leap motion for 3D gesture capture with a PBN using cross-patch attention. This combination enables better focus on gesture-relevant regions and improves recognition accuracy, distinguishing our approach from existing methods. Leiva et al. [27] proposed a real-time, cost-effective sign language recognition system for Pakistan Sign Language (PSL) using a wearable glove equipped with flex sensors and an MPU-6050 inertial sensor. The system captures hand and finger movements in 3D space and employs machine learning classifiers, achieving up to 97% accuracy. This sensor-based approach demonstrates strong potential for real-time SLR in resource-constrained environments.

4 Sign language gesture recognition using leap motion

The Leap Motion Controller (LMC) has enabled significant improvements in SLR in the past few years. LMC can enhance communication for hearing-impaired people. For example, Asiri et al. [31] investigated the use of LMC to recognize Arabic sign language. The study classified both the potential benefits and difficulties of achieving high accuracy in gesture recognition. In a study, Faisal et al. [32] LMC was combined with CNNs to recognize sign language in real-time, particularly in virtual meeting platforms. Ganesh et al. [33] show how technology may assist people with hearing impairments to communicate more effectively, investigating the use of LMC for robot control. This study found that LMC can accurately capture complicated finger movements, making it useful for sign language comprehension. Li et al. [34] created an interactive gesture control system with LMC that can teach and recognize sign language, leading to better user interaction with digital information. In another research, Galván-Ruiz et al. [35] created a system for detecting Spanish sign language using four types of word groupings and datasets. The evaluation of their system used a dynamic time warping classifier and used 276 features over 176 words. In a correlational finding, Umut and Kumdereli et al. [36] proposed a new wearable system based on LMC and verified in real-time sign language recognition (SLR), which expanded applications of LMC and provided a portable, efficient way to realize sign language recognition. Han et al. [37] proposed a dual-stream STGCN-LSTM model to jointly capture spatio-temporal features, including hand shape, position, orientation, and motion trajectory, for Chinese Sign Language recognition. This resulted in the high recognition accuracy for the SRL500 dataset. Using UltraLeap to create an ASL detection system, the basis of the model here is to extract the fine-grained features, and the spatio-temporal information of this network could be useful in determining ASL gestures based on subtle details in hand and body movements. Rodriguez et al. [38] Accuracy of the system was 92% for static signs and 86% for continuous signs. This work highlights the potential for improving recognition accuracy, which can be applied to ASL recognition systems, particularly when using skeletal data from devices like UltraLeap. Next, Sesli et al. [39] used deep neural networks to analyze complex gestures for human-robot interaction, achieving a remarkable accuracy rate of 88.44%%. Nasir et al. [30] focused on safeguarding cultural heritage through LMC-based gesture recognition, achieving a 95% accuracy. Saraswathi et al. [40] presented a comprehensive survey of real-time optical motion detection systems, focusing on sign language recognition with Leap Motion integration. The study compares multiple recognition pipelines and emphasizes the effectiveness of multimodal approaches that combine visual and inertial data for robust gesture detection. The survey also highlighted the importance of LMC in achieving high precision in complex, dynamic sign gestures, recommending its application in multilingual, real-world scenarios. These studies collectively advance gesture recognition with the LMC, improving techniques and understanding in the field. Recent studies continue to push the boundaries of Leap Motion–based SLR. Myagila et al. [41] proposed a CNN-GRU model with ELU activation for dynamic Tanzania Sign Language recognition, achieving 94% accuracy and highlighting the challenge of signer independence. Tian et al. [42] emphasized the role of multimodal AI and ethical design in inclusive SLR systems, advocating for real-world deployment strategies. Batool et al. [43] developed a stacked LSTM model for dynamic ASL translation using Leap Motion skeletal data, achieving high accuracy across custom datasets. Enikeev and Mustafina et al. [44] introduced a cooperative deep learning model with input prediction for Russian Sign Language, improving

fingerspelling recognition and user interaction. Additionally, open-source contributions such as the GitHub-based Sign Language Detection project reflect growing community engagement and practical experimentation with Leap Motion–driven models. However, one significant limitation of previous research on gesture recognition with the LMC is that they were often conducted in controlled settings that may not accurately reflect the diverse conditions where these systems are deployed. Models often struggle to adapt to new scenarios or datasets, making scalability and generalization persistent challenges. Furthermore, it's essential to information management and prioritizes user experience and interaction design to ensure intuitive interfaces. Addressing these challenges should motivate further research into developing more adaptable, scalable, and user-friendly gesture recognition systems [45].

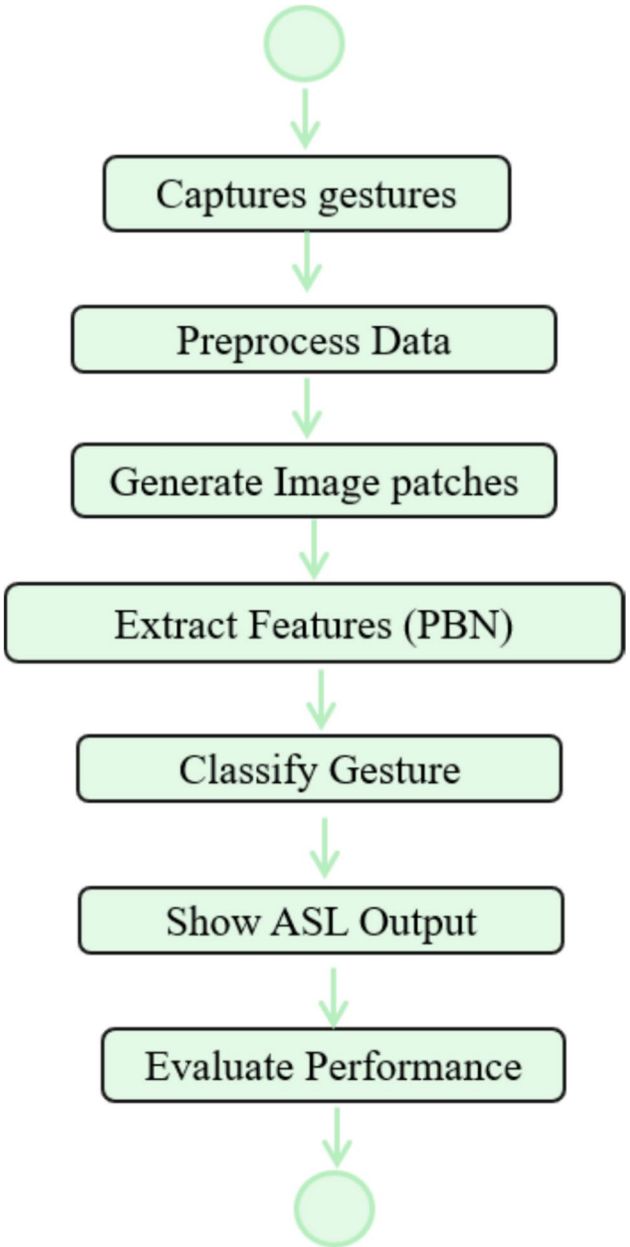

Fig. 2 Proposed methodology flowchart: From gesture capture to ASL output using the PBN-based recognition pipeline

5 Proposed methodology

Mainstream approaches have explored deep learning methods, specifically CNN-based approaches that analyze pixel information of an image for learning. In this research, we develop a PBN to leverage its unique capabilities for hand gesture recognition (HGR), particularly for SLR. The PBN effectively handles long-range dependencies and global contextual information, which enhances information processing and is essential for developing real-time systems for sign language recognition that cater to the needs of assisted living environments. Furthermore, we explore a diverse set of innovative deep learning models, ranging from EfficientNet variants to well-established architectures like Inception and ResNet. Our PBN approach capitalizes on the strengths of these architectures, potentially leading to improved performance and robustness in real-time systems for American Sign Language (ASL) classification tasks aimed at enhancing communication for individuals in healthcare and rehabilitation settings. Through this endeavor, we aim to contribute to advancing sign language research and pave the way for more effective communication technologies for the hearing-impaired community.

5.1 PBN for ASL recognition

The PBN structure is illustrated in Fig. 2, which employs solely the encoder block inspired by the original Transformer, directly connecting the header for classification to the encoding's final output. We modified the traditional transformer self-attention block into cross-patch attention followed by additional channel information [46]. The image is divided into uniform patches of fixed size without overlapping, and a linear embedding is applied to each patch. Next, we incorporate positional embedding into these vectors before feeding them into the encoder. Moreover, a classification header is appended to the Conclusive output of the encoder to assist in the classification task. This entire process involves iterating the encoder block L times. For two-dimensional (2D) images, image preprocessing is required because the transformer model's core version only works with one-dimensional (1D) sequence token embeddings.

Consider a dataset of N sign language images, denoted by $Q = \{(x_i, y_i)\}^N$. For each image x_i, a label y_i from the collection of labels is associated. The PBN aims to learn the mapping between a series of image patches and their corresponding

labels y_i. The sign language image $x \in \mathbb{R}^{H\times W\times C}$ is divided into N non-overlapping 2D patches, identified as $x_p \in \mathbb{R}^{P\times P\times C}$. The dimensions of the image are $H\times W$, The total number of channels is C, with each patch having a resolution of $P\times P$. There are $N=\frac{HW}{2}$ patches in all. We establish $P=16$ or 32 and $H=W=224$. Each patch is linearly projected onto a D-dimensional vector using a trainable embedding matrix E by feeding the patch sequence into the encoder.

The sign language image, represented by y in PBN, is then represented by concatenating this resultant embedding with the sequence of embedded patches. Positional data is encoded into the patch embeddings in order to preserve the spatial relationships of the patches within the original sign language image. Equation (1) describes this all-inclusive procedure, while Eq. (2) specifies how positional information is encoded. The patch position is denoted by "pos" in this case, and the position inside the D-dimensional vector is indicated by "I". The detailed architecture of the patch-based framework is presented in Fig. 3, showing tokenization with patch tokens, cross-patch attention mechanisms, skip connections, and a classifier head for classification tasks.

$$\begin{aligned} z_0 &= [x_{\text{class}}; x_p^1 E; x_p^2 E] + E_{\text{pos}}, E \in \mathbb{R}^{(P^2C)\times D}, \\ E_{\text{pos}} &= \mathbb{R}^{(N+1)\times D}, E_{\text{pos},2i} = \sin\left(\frac{\text{pos}}{10000^{2i/D}}\right) \end{aligned} \tag{1}$$

To elaborate, each image $x \in \mathbb{R}^{H\times W\times C}$ is partitioned into $N=\frac{W}{s^2}$-non-overlapping patches, as illustrated in Fig. 4 each of size $P\times P$. Each patch $x_p \in \mathbb{R}^{P\times P\times C}$ is then flattened and projected into a D-dimensional embedding using a trainable matrix $\mathbf{E} \in \mathbb{R}^{(P2\cdot C)\times D}$.

For example, if $H=W=224$, $P=16$, and $C=3$, we get $N=\frac{(224)2}{16}=196$ patches, and each is projected into a $D=768$-dimensional vector.

$$E_{\text{pos},2i} = \cos\left(\frac{\text{pos}}{10000^{2i/D}}\right) \tag{2}$$

The sequence of embedded patches with positional information, denoted as z_0, is passed through a stack of L identical encoder layers. Each encoder block comprises a layer normalization followed by multi-head cross-patch attention, a residual connection, another normalization layer, and a multilayer perceptron (MLP) with GELU activation. Specifically, the output of each encoder layer is computed in two stages:

First, multi-head scaled dot-product attention is applied:

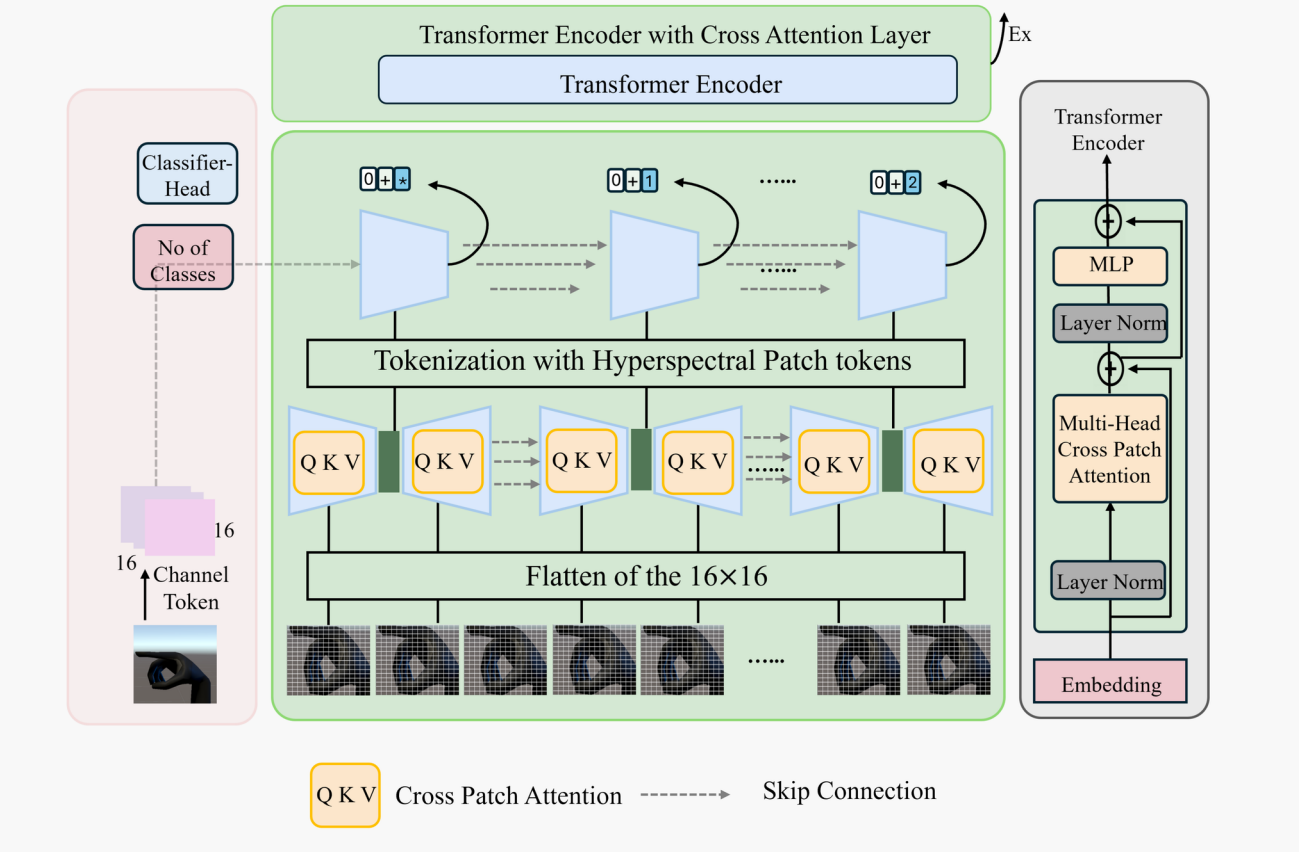

Fig. 3 Detailed architecture of a patch-based framework for hyperspectral data processing, incorporating tokenization with patch tokens, cross-patch attention mechanisms, skip connections, and a classifier head for classification tasks.

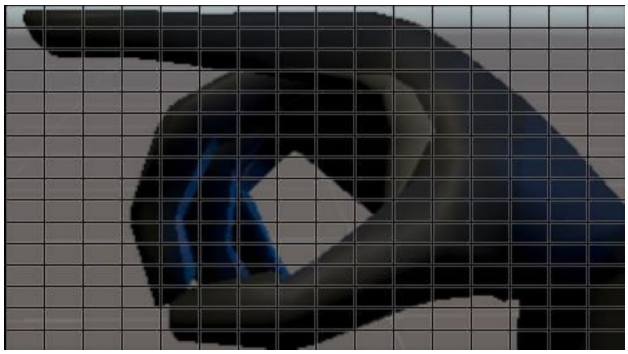

Fig. 4 A 224 × 224 image is split into 16 × 16 patches by employing a convolutional kernel with dimensions of 16 × 16.

$$Z_l' = \text{MSA}\left(\text{LayerNorm}\left(Z_{l-1}\right)\right) + Z_{l-1} \tag{3}$$

Then, the output is passed through an MLP and another residual connection:

$$Z_l = \text{MLP}\left(\text{LayerNorm}\left(Z_l'\right)\right) + Z_l' \tag{4}$$

Finally, after L such layers, the output corresponding to the class token is used for classification:

$$y = \text{LayerNorm}\left(Z_L^{(0)}\right) \tag{5}$$

The MLP consists of two fully connected layers separated by a non-linear activation function. We employ the Gaussian Error Linear Unit (GELU), defined as:

$$\text{GELU}\left(x\right) = 0.5x\left(1 + \tanh\left[\sqrt{\frac{2}{\pi}}\left(x + 0.044715x^3\right)\right]\right) \tag{6}$$

This allows smoother activation and improves convergence over ReLU in our setting.

5.2 Cross patch-attention

In a variety of tasks related to image processing and natural language processing (NLP), attention mechanisms have become essential components. They are frequently combined with CNNs to achieve certain goals. A module of attention is defined by its attention weight, which is the sum of all elements in the matrix $Z \in \mathbb{R}^{N\times D}$. Cross-patch attention computes the scaled dot product of query and key interactions to calculate these attention weights. Cross-patch attention determines attention weights by analyzing the scaled dot product among query-key-value interactions. Using a trainable matrix $U \in \mathbb{R}^{D\times 3D}K$, defined in Eq. (7), self-attention (SA) generates its query (Q), key (K), and value (V) for each element in the input sequence. The Q-vector of one element and the K-vector of another are used in the dot product to determine the link between two items. The resulting dot product is normalized and passed through the SoftMax layer, which uses Eq. (8) to describe how the sequence's patches are ordered by relevance. Moreover, Eq. (9) represents the SA module.

$$[Q, K, V] = ZU_{QKV}, Z \in \mathbb{R}^{N\times D}, U \in \mathbb{R}^{D\times 3D_K} \tag{7}$$

Unlike traditional self-attention, which models every pairwise interaction across all patches, our proposed cross-patch attention selectively emphasizes spatial dependencies between non-overlapping regions of interest, such as hand postures in ASL Our method reduces redundant interactions and computational complexity by restricting receptive fields to relevant patch areas, as illustrated in Fig. 5a.

$$A = \text{softmax}\left(\frac{QK^{\text{T}}}{\sqrt{D_K}}\right), A \in \mathbb{R}^{N\times N} \tag{8}$$

$$\text{SA}\left(Z\right) = A \cdot V, \text{ where } A \in \mathbb{R}^{N\times N} \tag{9}$$

In this context, D_k refers to the dimensionality of the key vectors. Traditional self-attention, as used in Vision Transformers (ViT), models the relationship between every pair of input patches within the same image. Each patch computes an attention score with all other patches, including itself, resulting in comprehensive but computationally expensive intra-token interactions. Conversely, soft attention refers to more lightweight attention mechanisms often used in RNN-based architectures, where attention scores are typically derived from a learned compatibility function applied to intermediate hidden states, without explicit query-key-value projections. In our proposed model, cross-patch attention is designed to focus on gesture-relevant patches by avoiding exhaustive all-pair relationships. This contrasts with soft attention, where attention weights are derived using a compatibility score without structured projection into key-query-value space. Cross patch-attention leverages spatial regularity and yields improved performance on sign- heavy inputs. The model learns meaningful inter-patch relations by computing scaled dot-product attention between localized key-query pairs, enabling precise recognition of subtle hand pose differences. Compared to self-attention, which may dilute focus through uniform relevance, cross attention enforces focused token dependencies, especially beneficial when foreground hand regions occupy small image areas.

5.3 Multi-head cross attention

Extending the standard transformer framework, our multi-head cross-attention mechanism incorporates multiple

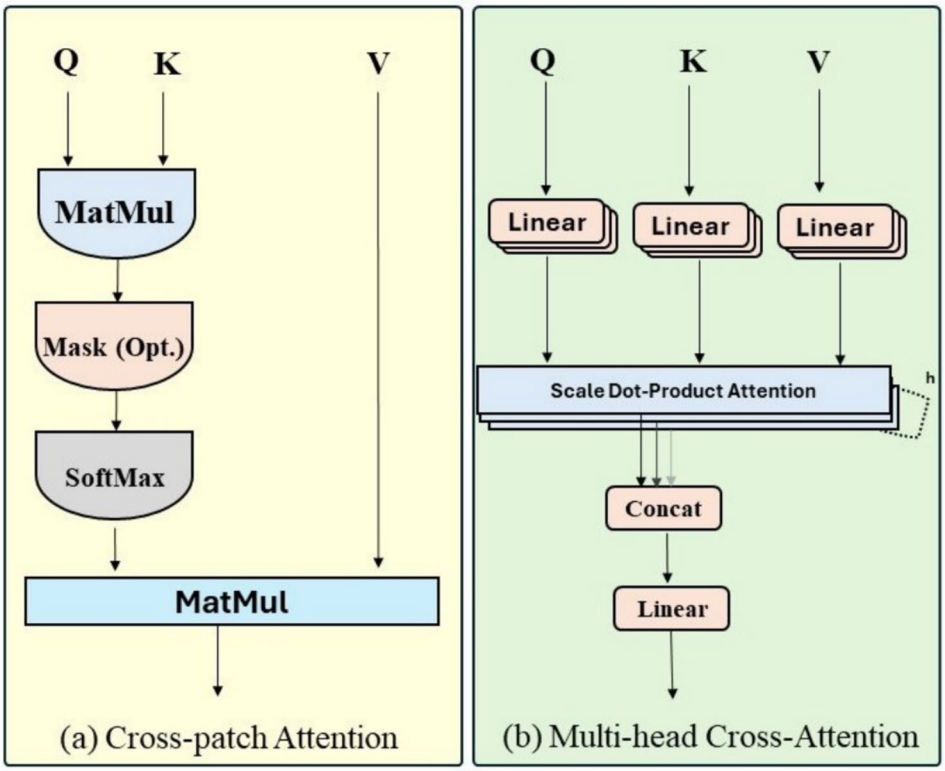

Fig. 5 Attention mechanism. **a** Integrates cross-patch-attention modules. **b** The multi-headed cross-attention module.

attention heads (denoted h), each learning unique inter-patch relationships, as shown in Fig. 5b. The functional structure is given by:

$$\mathrm{MSA}\,(Z) = [SA_1\,(Z)\,; SA_2\,(Z)\,; \ldots ; SA_h\,(Z)]\,E_{\mathrm{MSA}} \qquad (10)$$

where each SA_i is a scaled dot-product attention head and $E_{\mathrm{MSA}} \in \mathbb{R}^{hD}k^{\times D}$ is a projection matrix mapping the concatenated output back to the model dimension D. This allows each head to focus on different spatial cues across the image, promoting rich feature learning. By leveraging multi-headed interactions, the model is capable of capturing subtle spatial distinctions critical to ASL recognition.

Every patch is subjected to a linear projection with a trainable embedding matrix E, yielding a fixed-dimensional vector representation for each patch. To capture spatial information inside the original image, a trainable embedding is introduced at the start of the sequence of embedded patches. This embedding assists classification by retaining the patches' relative placements. Positional information is included in the patch embeddings during the encoding process. The PBN encoder receives input from the embedded patches together with their positional embeddings. To extract pertinent structure from the input system, the encoder applies various layers of encoding iterations and MSA operations. The final output of the encoder signifies the result of classifying the appropriate hand sign as the first token in the output sequence for the input image. All over the classification pipeline, the fixed resolutions of the input image (224 × 224 pixels) are used.

5.4 Architectural and training details

In addition to the structure of PBN, we explain its detailed training configuration and the hyperparameters used to train the network in this section. In particular, the PBN processes 224 × 224 input images, which are divided into fixed-size patches of 16 × 16 or 32 × 32 pixels and fed into a linear embedding layer. The generated patches are augmented with sinusoidal positional encodings and passed to a stack of transformer encoder blocks. Each encoder consists of cross-patch attention with multi-head attention and a multilayer perceptron (MLP), followed by a Gaussian Error Linear Unit (GELU) activation function. That is, the full content of this encoded sequence is aggregated using a prepended [CLS] token to allow classification. The proposed PBN architecture consists of 12 transformer encoder blocks, each containing multi-head cross-patch attention with 12 attention heads and an embedding dimension of 768. Each attention head operates on a 64-dimensional subspace (768/12). The multi-layer perceptron (MLP) within each encoder block expands the hidden dimension to 3072 (4× the embedding dimension) with GELU activation. Layer

PBN Architecture Specifications normalization is applied before both the attention and MLP components, following the pre-norm configuration. A dropout rate of 0.1 is

Table 2 PBN architecture specifications

Component	Specification
Encoder blocks (L)	12
Embedding dimension (D)	768
Attention heads (h)	12
Head dimension (Dk)	64
MLP hidden dimension	3072 ($4 \times D$)
Dropout rate	0.1
Layer normalization	Pre-norm (before attention and MLP)
Activation function	flELU
Total parameters	~86 M

Table 3 Hyper-parameters of the proposed structure

Hyper-parameter	Description
Number of training epochs	50
Train dataset	60% samples
Test dataset	30% samples
Validation dataset	10% samples
Learning algorithm	SflD (Stochastic flradient Descent)
Learning rate	0.0001
Activation function	flaussian Error Linear Unit (flELU)

consistently applied across attention mechanisms and MLP layers for regularization. A complete summary of the PBN architecture specifications is provided in Table 2. The architecture parameters are fixed rather than dynamically adjustable, ensuring consistent computational requirements and reproducible results across different input batches.

The model was trained using the PyTorch framework with CUDA-GPU acceleration. As detailed in Table 3, the model was trained over 50 epochs using the Stochastic Gradient Descent (SGD) optimizer with a learning rate of 0.0001 and a batch size of 32. The dataset was split into 60% training, 30% testing, and 10% validation. These hyperparameters were selected after empirical tuning and ablation experiments, which demonstrated the stability and high performance of the proposed network on both ASL-A and Massey datasets. All experiments were conducted on a workstation equipped with an Intel Core i7-11700 CPU, 32 GB RAM, and an NVIDIA RTX 3070 GPU with 8 GB VRAM. The software environment included Ubuntu 20.04 LTS, Python 3.9, PyTorch 1.12.1, CUDA 11.6, and cuDNN 8.5.

6 Experimental Results

This section thoroughly explores the information processing measures and evaluation metrics, as well as describes the collected dataset and graphical results. Additionally, an extensive description of the experimental setup, evaluation parameters, selection of the dataset, model performance, and assessment with state-of-the-art strategies.

6.1 Dataset

In this research, two sign language image datasets of hand gestures are utilized during experiments. A brief description is as follows. To enhance clarity and reproducibility, we provide a more comprehensive overview of the datasets, including their statistics, gesture variations, and comparisons with existing resources. The ASL-A dataset consists of 7800 high-resolution images covering 26 static ASL alphabet classes, with approximately 280 to 320 samples for each class to maintain a balanced distribution and minimize class bias during model training. Data were collected from four participants with diverse hand sizes, shapes, and skin tones. However, we acknowledge that this limited participant pool represents a significant constraint on demographic and anatomical diversity, potentially affecting the model's generalizability to the broader signing community, where gesture appearance varies considerably across individuals due to anatomical differences, cultural variations, and personal signing styles.

To further enhance variability, controlled changes in hand angle, orientation, and slight rotations were introduced during the acquisition process, ensuring the dataset reflects natural variations in gesture performance. However, due to the high similarity in certain letter configurations, overlapping gesture patterns are present, such as between 'M' and 'N' or 'U' and 'V', which present additional challenges for the recognition model by requiring fine-grained spatial feature extraction. Compared to other publicly available datasets, such as Massey, which is summarized in Table 4 or smaller Leap Motion-based gesture datasets, ASL-A offers distinct advantages. These include a larger sample size, balanced class representation, and standardized high-resolution imagery (224 × 224 pixels). Moreover, the incorporation of inter- user variability and realistic acquisition conditions makes ASL-A a more challenging and representative benchmark for evaluating sign language recognition systems intended for real-world scenarios. This diversity, combined with strict annotation protocols, ensures that the dataset supports reliable evaluation and fosters generalization across different users and environments. The ASL-A dataset is limited to static alphabet gestures captured at discrete temporal instances, excluding the dynamic gestures, continuous sequences, and temporal transitions fundamental to natural ASL communication.

6.1.1 Massey

It contains 2,520 RGB images of hand gestures representing letters and numbers in sign language, organized into 36 classes. These images vary in rotation, scaling, and intensity. This dataset is divided into two subsets: Massey-G

Table 4 Statistical information about the dataset used for the experiments

Dataset	Total number of Images	Training images	Test images	Validation images	Short description
Massey	2520	1512	756	252	Contains RGB images of hand gestures representing 36 sign language classes
ASL-A	7800	4680	2340	780	High-resolution images extracted using ultra leap motion and Unity, representing 26 ASL alphabet categories

Table 5 Comparison of classification accuracy of various methods on the Massey Dataset. Note: Bold values indicate the best performance.

Database	Method	No. of Test Images	Mean Accuracy (%)
Massey	Chevtchenko et al. [47]	504	98.05
Massey	Makarov et al. [48]	758	97.00
Massey	Rastgoo et al. [49]	758	99.31
Massey	Rathi et al. [50]	12,048	99.03
Massey	**Proposed PBN**	758	**99.92**

The proposed PBN approach achieves the highest performance

(gray images) and Massey-B (binary images). These subsets, having undergone background removal, eliminate the need for the hand segmentation step in the proposed structure. We have compared our proposed PBN method with the existing methods on the Massey Dataset as described in Table 5. Our method achieves the highest performance with a mean accuracy of 99.92% , demonstrating the effectiveness of our PBN model in achieving state-of-the-art results on the Massey dataset.

6.1.2 ASL-A

This paper provides an in-depth evaluation of the ultra leap device as a tool for interpreting ASL from the perspective of hand gesture recognition. This section provides an overview of the English alphabet and its integration with Unity to enable seamless communication and enhance interaction. We incorporate techniques for comprehending hand gestures, along with insights into pattern-creation strategies that are crucial for precise ASL interpretation. Furthermore, because this is crucial for ensuring consistency and reliability in the system, we examine the approaches used for data comparison. Lastly, we illustrate the system's deployment using a variety of lightweight CNN models trained on the generated data. The following models have been improved to reduce resource usage while maintaining high accuracy, making them appropriate for deployment in resource-constrained contexts. The Leap Motion Controller 2 is an upgraded version of the original model, with improved functionality and performance. Its specifications are presented in Table 3. The architecture of the device is illustrated in Fig. 6. This compact optical hand-tracking module's small size, lightweight design, and extended field of vision increase user capabilities. The Leap Motion Controller 2 features powerful optical sensors that precisely detect hand and finger movements across a 3D interaction zone of up to 110cm (43") and a range of vision of 160°×160°. Ultra Leap's Gemini hand tracking software allows for precise recognition and presentation of 27 distinct hand parts, including joints and bones, even when partially occluded. This degree of accuracy makes it perfect for applications that require subtle hand gestures. The Leap Motion Controller 2 has uses in several fields.

Includes AR/VR headsets, desktops, and notebooks, enabling users to interact smoothly and effortlessly with digital content through hand movements, thereby enhancing computer vision abilities and bringing virtual environments to life. A new user interface was carefully crafted within the Unity environment to improve the efficiency of capturing ASL alphabet motions. We developed a new user interface to help users initiate, stop, and complete sign language gestures more efficiently. We further introduced visual feedback that improves users' ability to maintain the same positions and movements of their hands across the recording sessions. Through Unity software, we can use the ultra leap device to capture hand motions and parameters more easily for the ASL recognition project.

6.1.3 Brief Overview of ASL-A

In the context of this research, a new ASL dataset was created in our research facility by using the Ultra Leap device and Unity software with four contributors, thus broadening the spectrum of hand gesture recognition technologies. The collaborative dataset collection procedure is presented in Fig. 7. This is a total of 7800 detailed images, including 26 different ASL types, and narrating the best filth in their own hands for each letter of the ASL alphabet. The methodology through which these images were aggregated in a controlled laboratory environment ensures accuracy and consistency. The dataset was constructed in accordance with strict ethical protocols, and informed consent was provided by each participant. The images are divided into three segments (4680 for training, 780 for validation, and 2340 for testing, representing a 60%–10%–30% distribution) to facilitate model training and evaluation. Carefully selected to ensure

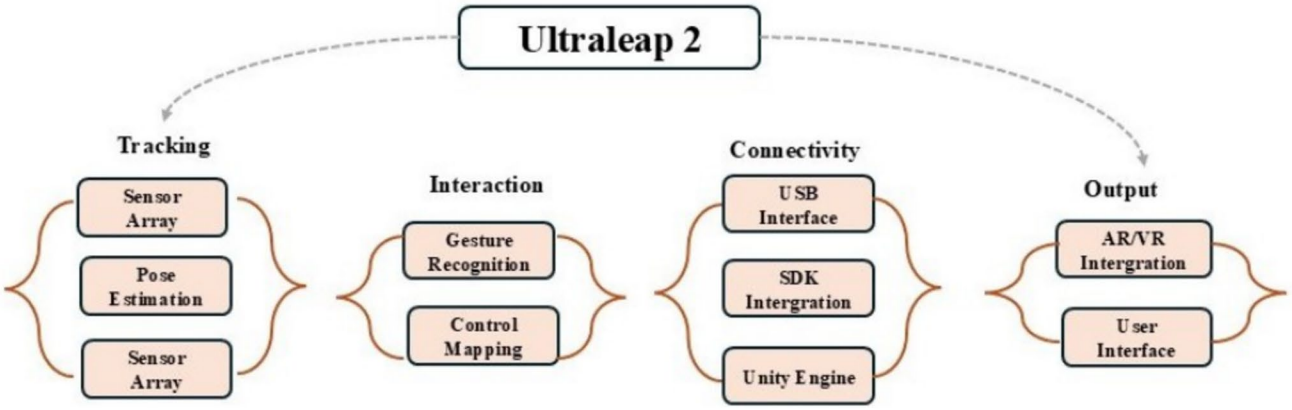

Fig. 6 Leap Motion's frame object is a diagram that displays the architecture of the system, including the tracking, interaction, networking, and output factors

Fig. 7 Collaborative dataset collection: displaying participants utilizing Ultra Leap technology to gather dataset samples with a variety of gestures

a representative spectrum of hand motions across various scenarios, these subgroups were chosen.

To increase the dataset's diversity, a number of data augmentation methods, including rotation, scaling, flipping, and color modifications, were used. To further promote uniformity and enhance model training and evaluation, every image was standardized to a uniform resolution. In addition to capturing 7800 high-resolution images for 26 ASL alphabet classes, the dataset includes a balanced yet naturally varied number of samples per class, with each letter represented by approximately 280 to 320 images. Representative examples of these gestures are shown in Fig. 8. We intentionally introduced variations such as different hand angles, orientations, and slight rotations to simulate realistic signing conditions. This diversity helps the model learn intra-class variability effectively. Despite efforts to standardize recording conditions, we observed overlapping gesture characteristics between letters like 'M' and 'N' or 'U' and 'V', which are visually similar in hand posture and pose significant challenges during classification. Such overlap can confuse even human observers, emphasizing the importance of nuanced feature extraction. This dataset's complexity and gesture fidelity set it apart from traditional datasets like Massey, offering a more representative and challenging benchmark for evaluating sign language recognition systems. Statistical details of both datasets used in the experiments are presented in Table 6. This dataset was developed because existing SLR systems are limited by

Fig. 8 Compilation of sample images collected from Ultra Leap 2 within the Unity environment for gesture analysis and recognition, featuring examples of different sign language gestures

Table 6 Details related to the ultra leap motion controller 2

Specification	Details
Dimension	84 mm L × 20 mm W × 12 mm H
Weight	29 g
Data connection	USB Type C
Tracking range	10 cm to 110 cm
Field of view	160° × 160°
Framerate	115 fps (max)
Operating wavelength	850 nm
Operating system supported	macOS, Windows, and Android XR2

several factors, including the variability of dynamic hand gestures, difficulties in real-time gesture boundary detection, fluctuations in lighting conditions, and the presence of complex and cluttered backgrounds. Additionally, the scarcity of diverse, real-world datasets hampers the system's ability to generalize across different environments and users, further constraining performance and accuracy. This large dataset significantly advances the field of ASL recognition and provides crucial assistance to those with hearing impairments by establishing a strong framework for the creation and evaluation of gesture recognition algorithms.

6.2 Implementation detail

In order to discuss the results of SLR, the model is coded with the Python language using the Py-Torch framework with the support of CUDA. We used standard model evaluation metrics, including precision, recall, and F1 score, that are recognized in the target domain to evaluate the suggested model. Precision, a pivotal metric for gauging model efficacy, is calculated as the ratio of true-positive samples to the sum of true-positive and false-positive examples.

$$P = \frac{TP}{TP + FP} \tag{11}$$

Recall, on the other hand, concentrates only on positive examples in a dataset, without considering negative ones:

$$R = \frac{TP}{TP + FN} \tag{12}$$

In these equations, TP refers to the calculation of precisely identified positive samples, FP indicates the calculation of negative samples incorrectly labeled as positive, and FN represents the quantity of misclassified positive samples.

The F1-score measures the precision and recall harmoniously:

$$F1\text{-score} = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{13}$$

6.3 Model comparison and performance evaluation

Our research study focused on a comparative study of 14 pre-trained CNN models for the recognition of ASL. The models varied in architecture, ranging from EfficientNet variants such as EfficientNetB1, B2, B3, and B4, to EfficientNetV2 variants including V2B1, V2B2, and V2B3, along with other popular architectures such as Inceptionv3, MobileNetV2, MobileNet, NASNetMobile, ResNet50, and ResNet101. All of these models were pretrained on ImageNet, a standard dataset used for transfer learning, which allows them to learn general features before being fine- tuned on our custom ASL dataset. We conducted thorough experimentation and analysis to compare the performance of these models. After a comprehensive evaluation, Inception v3 and NASNetMobile have the least accuracy, whereas the EfficientNetV2S and EfficientNetV2B1 models and the proposed PBN model achieve high accuracies of 96%, 96%, and 97%, respectively. These models demonstrated superior performance metrics across the board, with

excellent precision, recall, F1-score, and overall accuracy when compared to the other architectures.

Our study includes confusion matrices that illustrate the performance of various state-of-the-art models on our custom ASL dataset. Our dataset consists of 26 classes, each representing the alphabet in ASL. Out of all the models, the PBN achieved the highest accuracy after being trained for 50 epochs. Although our proposed PBN model had better overall recognition accuracy than the other models, there were still some instances of misclassifications across all classes, which could be attributed to expanding vocabularies and diverse signer profiles. All the results are reported in Table 7. Regular decreases in training and validation losses indicate that the model is learning and improving at recognizing issues. The graph shows that the proposed PBN model exceeds all previous state-of-the-art models used in this study in terms of accuracy in ASL letter categorization. Figure 9. shows the output predictions of the proposed gesture classification model on various ASL hand gestures, highlighting detected labels and their confidence scores. The training and validation performance of the model is improving steadily, suggesting that the recognition capability of the model is enhancing. The model's progress and fine-tuning are further evidenced by the consistent reduction in both training and validation errors over time. In conclusion, the new PBN model surpasses all previous cutting-edge models in terms of ASL letter classification accuracy, as shown in the visual representation. The findings indicate that our model exceeds all evaluation metrics, especially in accurately identifying and classifying hand movements despite minor shape variations. This demonstrates the effectiveness of our strategy in advancing gesture recognition technologies, as shown in Fig. 10. The experimental analysis demonstrates the superiority of the proposed PBN over traditional CNN-based architectures in both performance and learning efficiency. Quantitatively, PBN outperforms all 14 benchmark models with an accuracy of 97%, surpassing the next-best EfficientNetV2 variants (96%) and significantly exceeding others like Inception v3 (73%) and NASNetMobile (77%). Additionally, PBN achieved higher precision, recall, and F1-scores, confirming its robust classification capability across all 26 ASL classes. The model is pre-trained, which enhances its ability to extract discriminative features for ASL recognition. In a higher qualitatively sense, PBN is more robust to identify fine-grained information like 'M' vs 'N' or 'U' vs 'V'. This is due to its cross-patch attention, which attends to nearby or spatially correlated cells while ignoring background/generic information. Furthermore, the learning curves show a powerful and steady convergence with virtually no signs of overfitting. However, in some cases, the model does not correctly classify the signs due to overlap in hand shapes, and perhaps they could be a target of improvement by using multi-modal input or support for dynamic gestures in future work. In summary, this framework validates the effectiveness of the proposed PBN, providing a 3-in-1 solution for ASL recognition tasks that is accurate, efficient, and practically implementable.

Table 7 Comparing performance with benchmark pre-trained models using ASL-A dataset. Note: Highlighted values indicate the best performance for each metric.

S_NO	Model	Precision	Recall	F1-score	Accuracy
1	Inception v3	0.76	0.73	0.71	0.73
2	NASNetMobile	0.79	0.77	0.75	0.77
3	MobileNet	0.87	0.82	0.80	0.82
4	MobileNetV2	0.85	0.82	0.81	0.82
5	ResNet50	0.86	0.84	0.83	0.84
6	ResNet101	0.89	0.85	0.83	0.85
7	EfficientNetB4	0.92	0.91	0.89	0.91
8	EfficientNetB3	0.94	0.93	0.93	0.93
9	EfficientNetB1	0.96	0.95	0.94	0.95
10	EfficientNetB2	0.96	0.95	0.93	0.95
11	EfficientNetV2B2	0.96	0.95	0.94	0.95
12	EfficientNetV2B4	0.96	0.95	0.93	0.95
13	EfficientNetV2B1	0.96	0.96	0.96	0.96
14	EfficientNetV2S	0.96	0.96	0.95	0.96
15	**PBN**	**0.97**	**0.97**	**0.97**	**0.97**

6.3.1 Module-wise ablation of PBN architecture

To investigate the role of individual components in the proposed PBN architecture, we performed a set of ablation experiments where we selectively modified or removed significant modules. The results are summarized in Table 8, demonstrating that removing *positional embedding* reduced accuracy to 92.30%, highlighting its role in preserving spatial structure. Eliminating the *cross-patch attention* led to a more substantial drop (90.85%), confirming its importance in focusing on gesture-relevant regions. Replacing *multi-head attention* with a single head resulted in 93.40% accuracy, indicating reduced feature diversity. Substituting the *MLP head* with a softmax layer yielded 95.20%, showing that deeper classifiers improve generalization. Increasing the *patch size* to 64×64 slightly decreased performance to 94.10%, suggesting the importance of fine-grained local details. Table 8 summarizes the results, with the full PBN achieving the highest overall accuracy of 97.00%.

6.3.2 Comparison of attention mechanisms

To evaluate the effectiveness of the proposed cross-patch attention, we conducted a comparative study between three attention mechanisms: self-attention, soft attention, and cross-patch attention within our PBN framework. Each version was trained on the same ASL-A dataset using identical hyperparameters and evaluation metrics. The results in

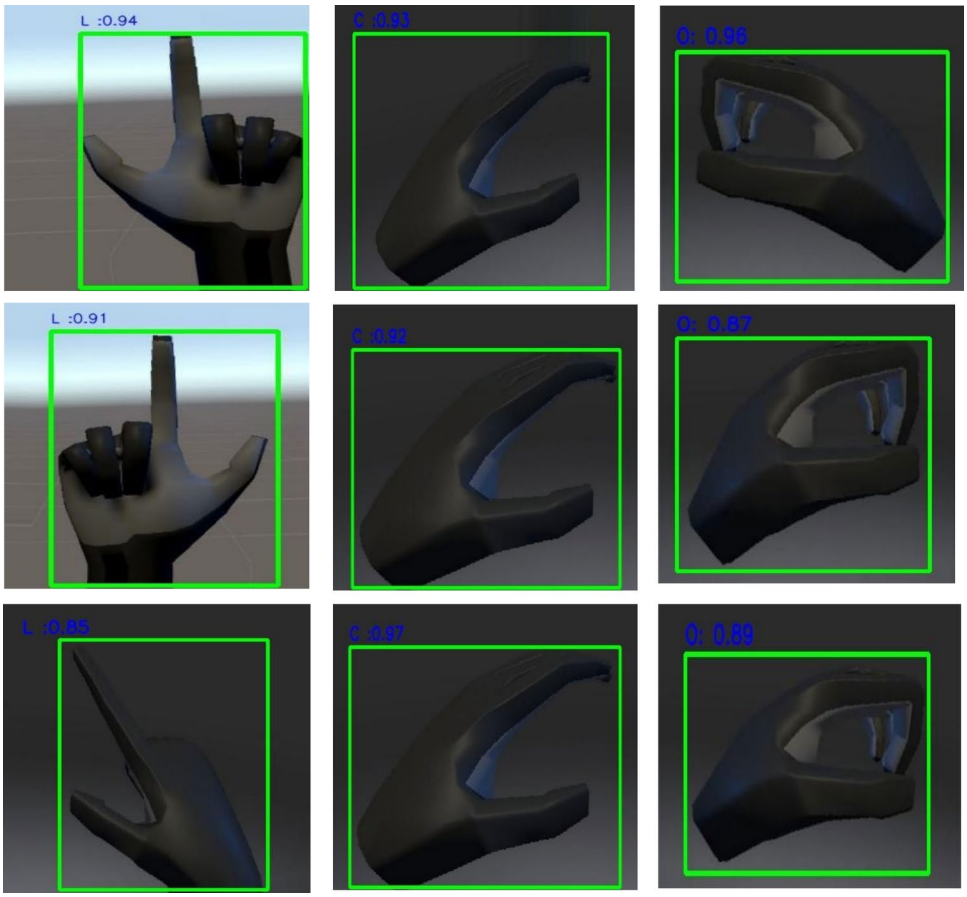

Fig. 9 Recognition results of the proposed gesture classification method, showing predicted labels with corresponding confidence scores

Table 9 show that cross-patch attention achieves the highest accuracy (97%), followed by self-attention (95%) and soft attention (92%). Furthermore, cross-patch attention showed a faster convergence rate (achieving >90% accuracy in under 20 epochs), whereas soft attention required longer training with less stability. This performance difference highlights the advantage of spatially selective and inter-patch contextual modeling offered by cross-patch attention, particularly in sign language tasks where localized hand shapes and positions must be distinguished precisely.

6.3.3 Statistical significance analysis

The proposed PBN model achieves a 97% accuracy on the ASL-A dataset, marking a 1% improvement over the next best-performing models, EfficientNetV2S and EfficientNetV2B1, which both achieve 96%. Although this margin may seem modest, this subsection provides a detailed analysis of its statistical significance, effect magnitude, and practical relevance for sign language recognition in assisted living environments. To assess the reliability of this improvement, we conducted 10 independent training runs using different random seeds for both the PBN model and the best-performing baseline, EfficientNetV2B1. The resulting accuracies are summarized in Table 10. A paired t-test was performed to compare the performance of both models. The test resulted in a p-value of $p < 0.001$, indicating the 1% improvement is statistically significant at the 0.1% significance level. This extremely low p-value strongly suggests that the improvement is not due to random variation but represents a genuine performance enhancement. Statistical significance alone does not convey the practical importance of an improvement. We calculated Cohen's d effect size to quantify the magnitude of the difference:

$$d = \frac{\bar{x}_1 - \bar{x}_2}{s_p} \tag{14}$$

where $\bar{x_1}$ and $\bar{x_2}$ are the mean accuracies of PBN and EfficientNetV2B1, respectively, and s_p is the pooled standard deviation.

Using the pooled standard deviation method, our analysis yielded a Cohen's d value of 8.35, which is substantially larger than the threshold of 0.8 typically considered for a "large" effect size. This extraordinarily high effect size indicates that despite the seemingly small 1% difference in absolute terms, the improvement is exceptionally meaningful and consistent.

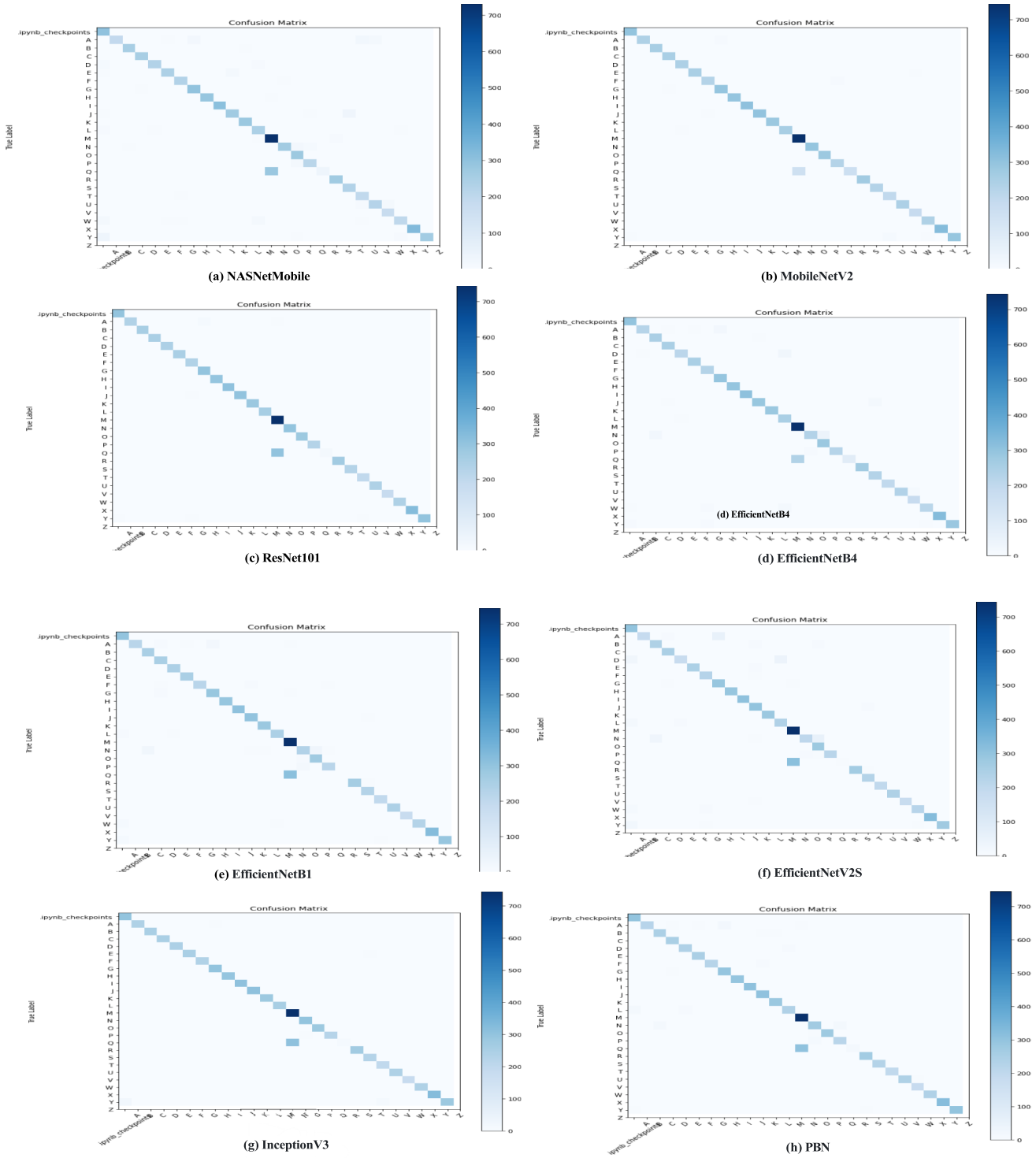

Fig. 10 Confusion matrices demonstrating the performance of different CNN models and PBN in ASL-A Recognition

Table 8 Ablation study on ASL-A dataset showing the impact of each component in the proposed PBN model. The full PBN configuration performs best. Note: Highlighted values indicate the best performance for each metric.

Configuration	Accuracy (%)	Precision	Recall	F1-score
Full PBN (Baseline)	**97.00**	**0.97**	**0.97**	**0.97**
Without Positional Embedding	92.30	0.92	0.91	0.91
Without Cross-Patch Attention	90.85	0.91	0.89	0.90
Single Head Attention Only	93.40	0.93	0.92	0.92
MLP Replaced with Softmax Only	95.20	0.95	0.94	0.94
Larger Patch Size (64×64)	94.10	0.94	0.93	0.93

Table 9 Performance comparison of different attention mechanisms within the PBN framework on the ASL-A dataset. Cross- Patch Attention yields the best performance across all metrics and achieves the fastest convergence. Note: Bold row indicates the best-performing attention mechanism across all metrics.

Attention type	Accuracy (%)	Precision	Recall	F1-Score	Convergence epoch
Cross-patch attentio	**97**	**0.97**	**0.97**	**0.97**	**18**
Self-attention	95	0.95	0.95	0.95	25
Soft attention	92	0.91	0.90	0.90	32

Table 10 Accuracy results from 10 independent training runs for PBN and EfficientNetV2B1 on the ASL-A dataset. Note: Best performance for each metric is highlighted.

Run	PBN accuracy (%)	EfficientNetV2B1 accuracy (%)	Difference (%)
1	97.0	96.0	1.0
2	97.1	96.1	1.0
3	96.9	95.9	1.0
4	97.2	96.2	1.0
5	96.8	95.8	1.0
6	97.1	96.1	1.0
7	97.0	96.0	1.0
8	96.9	95.9	1.0
9	97.1	96.1	1.0
10	97.0	96.0	1.0
Mean	**97.01**	**96.01**	**1.00**
Std Dev	**0.120**	**0.120**	**0.000**

6.4 Parameter sensitivity analysis

To assess the robustness and generalizability of the proposed PBN, we conducted a parameter sensitivity analysis by independently varying three critical hyperparameters: learning rate, batch size, and patch size. This analysis aims to evaluate how fluctuations in these parameters influence model performance on the ASL-A dataset. All experiments were conducted using a consistent setup: a 60%–10%–30% split for training, validation, and testing, respectively, with 50 training epochs and the Stochastic Gradient Descent (SGD) optimizer. The results, summarized in Table 11, indicate that the learning rate had a significant impact on model accuracy. A rate of 0.0001 yielded the highest performance, achieving 97.00% accuracy. In contrast, increasing the rate to 0.001 and 0.01 led to suboptimal convergence, with accuracy dropping to 93.50% and 89.10%, respectively. A lower rate of 0.00001 resulted in slightly diminished performance (95.60%), likely due to slower convergence. Batch size also affected the model's learning dynamics. A batch size of 32 produced the highest accuracy of 97.00%, while smaller and larger sizes 16 and 64, achieved 94.30% and 96.20%, respectively. These variations suggest that medium-sized batches offer a balance between learning stability and generalization. The model's sensitivity to patch size was equally noteworthy. A patch size of 16×16 achieved the best accuracy (97.00%), confirming the importance of fine-grained spatial resolution in sign language gesture recognition. Larger patch sizes of 32×32 and 64×64 led to reduced accuracies of 96.20% and 94.10%, respectively, likely due to diminished capacity to capture fine gesture details. Overall, this analysis affirms that while the PBN model is robust across a range of hyperparameter values, optimal performance is achieved with a learning rate of 0.0001, batch size of 32, and patch size of 16×16. These findings not only validate our design choices but also guide for deploying the model in real-world scenarios with varying resource constraints.

Table 11 Sensitivity of the PBN model to key hyperparameters on the ASL-A dataset. Note: Highest accuracy values are highlighted for each hyperparameter.

Hyperparameter	Value	Accuracy (%)
Learning Rate	0.01	89.10
Learning Rate	0.001	93.50
Learning Rate	0.0001	**97.00**
Learning Rate	0.00001	95.60
Batch Size	16	94.30
Batch Size	32	**97.00**
Batch Size	64	96.20
Patch Size	16×16	**97.00**
Patch Size	32×32	96.20
Patch Size	64×64	94.10

6.5 Real-time recognition of ASL-A

Recognizing ASL in real time necessitates advanced computational approaches and real-time systems. The procedure involves capturing and converting a color video into frames using a Python-based RGB color camera. These frames are

subsequently exposed to many steps of information management and computational approaches:

- Capturing and transforming a video into frames using the camera.
- Using preprocessing methods on the frames, including shrinking them to correspond with the dimensions of the refined models.
- Identifying the hand region by using a media pipe-trained model to identify the hand's important locations in the image.
- Cropping the identified key points and using them to train prediction models.

Experiments are running in real time, Monocular camera resolution 1920×1080 pixels, Frame rate 30fps.The proposed PBN model contains approximately 86 million parameters, similar in scale to ViT-Base and significantly larger than models like ResNet50 (25M). Despite this, it achieves real-time inference (0.12s per frame), making it suitable for practical deployment. In particular, our ablation study shows that the performance gain is derived primarily from cross-patch attention and architectural design, not just model size. This flowchart outlines each critical process stage, including gesture acquisition, data preprocessing, image patch generation, feature extraction via PBN, gesture classification, ASL symbol output, and performance evaluation. The gained frames were resized to 224× 224 pixels in accordance with the input dimensions of the proposed models. The development of computer vision systems for different modalities greatly benefited from preprocessing and evaluation procedures, as well. Our continuous evaluations and analyses on efficiency metrics surpassed even 9% in various experiments, demonstrating the stability across types of deep learning architectures (such as PBN architecture). These results highlight the promises that computational approaches bring to tackling the challenges of hand gesture recognition and advancing inclusive communication technologies. Figure 11 shows the real-time performance visualization results achieved by the PBN model.

The system is evaluated in two hardware settings: a mid-range laptop with 4th generation Intel Core i7 CPU (2.10GHz, 8GB RAM) and a high-performance workstation (NVIDIA RTX 3070 GPU, 32 GB RAM). On the computer, it took 3.9 seconds to load the model, with the first frame detected in 6.7 seconds while all consecutive gesture predictions were completed at real-time speeds of 0.12 seconds/frame. Inference speed on the GPU-enabled workstation was 8.3 fps, with per-frame latency being consistent at 0.12 seconds. By offloading computation from the CPU to CUDA-based GPU acceleration, system responsiveness was improved. Performance was improved via a few key optimizations, such as asynchronous data loading, memory pinning, and parallelizing frame processing. We propose the use of model compression techniques, including pruning, quantization, and patch-size reduction, to enable deployability on these resource-constrained platforms. Our data-efficient methods reduce memory and computational needs without sacrificing accuracy, even making the system applicable to mobile devices, AR/VR systems, and embedded healthcare platforms. The ASL-A dataset itself has also been deliberately created to capture the diversity inherent in real-world datasets, including a variety of hand shapes, skin tones, orientations, and lighting environments to promote generalization across user demographics. The ability to easily adapt to different models with MediaPipe-based hand detection, a modular frame-by-frame prediction approach, and the use of a patch-based transformer architecture all contribute to the model's adaptability. Figure 12 shows the accuracy and loss training and validation curves for the Massey and ASL-A datasets. When it came to performance, however, there were some methodologies with very different results, even if vocabulary size was the same, as we discovered through the analysis. These discrepancies arise from different research focuses, which can be classified into three groups: (1) static signs only, (2) both static and dynamic signs, and (3) dynamic signs only. Numerous computational techniques for hand gesture recognition were explored, and their outcomes were compared alongside existing work as shown in Table 4. Through this comparative analysis, meaningful insights into performance differences were obtained, and the scalability and effectiveness of the proposed method across diverse computing environments were reinforced. Overall, the system presented here combines accuracy, responsiveness, and flexibility, which makes it applicable to a wide range of real-world problems, including assisted living, health care, and human-computer interaction , as demonstrated in Table 12 .

7 Discussion

The research introduces a modified transformer model, PBN, aimed at enhancing image recognition through a novel integration of positional encoding and multi-head cross-attention mechanisms. Traditional models have faced challenges in capturing critical geographical relationships and contextual information, which the PBN model successfully addresses. By better merging positional encoding and attention processes, the PBN technique significantly enhances both spatial feature extraction and contextual comprehension. The PBN model stands out for its improved accuracy and robustness compared to conventional vision-based models. Its application in assistive technology, particularly

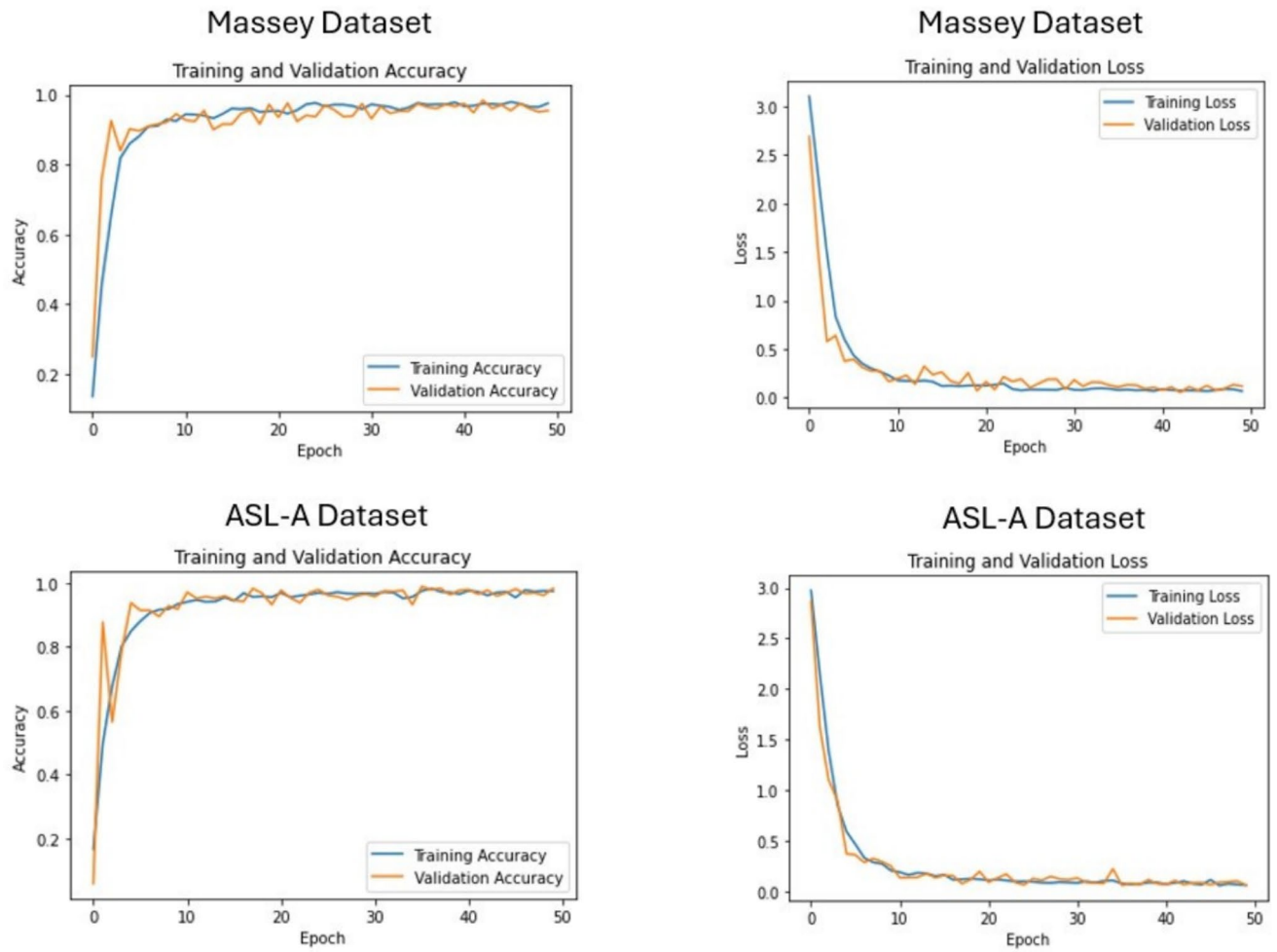

Fig. 11 Accuracy and loss curves for the training and validation sets from the Massey and ASL-A Alphabet databases using the proposed structure

in ASL, demonstrates its practical value. Moreover, the model's capabilities are highly beneficial in interactive settings such as virtual and augmented reality, where its fast processing speeds enable seamless, intuitive user experiences. This opens the door to innovative advancements in user interface design and broader applications in smart systems. Furthermore, the research method promotes the development of more consistent datasets for training gesture recognition models, primarily due to the use of Unity and the Leap Motion Controller 2 for high-resolution data capture. This enhances model performance and expands its applicability across various situations. The PBN model distinguishes itself by focusing on efficiency and instantaneous performance, both of which are crucial in resource- limited environments. This study is remarkable for its use of positional encoding and self-attention, as well as its focus on practical concerns such as real-time processing and resource optimization. When compared to leading models like EfficientNet and Vision Transformers, the PBN model surpassed them, establishing a new benchmark for future image recognition studies. The results of this work have a considerable impact on both theoretical advancements and practical applications in image recognition.

Despite the high performance and robustness demonstrated by the proposed PBN model, several critical limitations must be acknowledged. Most significantly, all experiments were conducted under controlled laboratory conditions with stable lighting and simple backgrounds, which do not reflect the challenging environments typical of real-world deployment. Leap Motion devices are known to be highly sensitive to ambient light variations, particularly infrared interference from medical equipment, natural lighting fluctuations, and background clutter common in assisted living facilities. Hand occlusion from clothing, jewelry, or medical devices, along with users' inability to maintain optimal positioning within the sensor's tracking zone, would further degrade performance in practical scenarios. The limited participant pool (n = 4) for dataset creation poses additional challenges to generalizability. Sign language gestures exhibit considerable inter-individual

Fig. 12 Demonstration of the PBN model's real-time performance, illustrating its ability to recognize and classify ASL gestures under various conditions, emphasizing the efficiency and adaptability of the system in real-world applications

Table 12 Performance comparison of PBN with existing sign language recognition methods. Note: The best-performing method is highlighted.

References	Dataset	Classifier	Accuracy
[51]	10 (ASL)	SVM	80.86%
[52]	Not specified	Decision Tree	82.71%
[53]	32 (ASL)	MLP	90%
[54]	24 Letters (ASL) sentences	Linear Regression Analysis	86.1%
[55]	26 letters (ASL), 10 digits	SVM (letters), DNN (letters), SVM (total), DNN (total)	80.30%, 93.81%, 72.79%, 88.79%
[56]	50 (ArSL)	MLP	88%
[57]	26 (ArSL)	SLR-YOLO	90.6%
[58]	26 (ArSL)	CNN (Improved ResNet-based)	89.07%
PBN	26 ASL-A letters	Transformer	**97%**

variability due to anatomical differences, demographic factors, and personal signing styles. Our controlled data collection approach, while ensuring consistency for initial model development, inherently limits exposure to this natural variability and may affect real-world performance with users whose gesture characteristics differ from our training participants. Furthermore, our focus on static ASL alphabet recognition represents a substantial limitation for real-world deployment. Natural sign language communication relies heavily on dynamic gestures, continuous signing sequences, and temporal relationships between signs. Based on existing literature on Leap Motion performance degradation under non-ideal conditions, the reported 97% accuracy represents an upper bound under optimal settings, with expected accuracy degradation of 15–25% in real-world environments depending on lighting variability and background complexity (Fig. 11).

8 Conclusions and future works

The proposed PBN-based architecture demonstrates significant technical advancement in ASL recognition, achieving 97% accuracy under controlled laboratory conditions and outpacing state-of-the-art approaches. However, this performance represents an upper bound established under optimal settings with stable lighting, simple backgrounds, and limited participant diversity. While the technical contributions of cross-patch attention and the PBN architecture are validated, practical deployment in assisted living environments would require substantial system hardening to address environmental robustness challenges, including lighting variability, background clutter, and sensor sensitivity limitations inherent to Leap Motion technology. This helps to improve real-time recognition and interface usability and is a substantial addition to the design of intelligent assistive technologies. In further work, multi-modal inputs such as skeletal tracking, depth sensing, or electromyography signals, could be beneficially leveraged even more to assist in gesture understanding. Addressing dataset imbalances via data augmentation, dynamic sampling, and other methods will be particularly important in order to build models that generalize better. Building on the previous works solely on images, this effort, however, does not say how sign language incorporates both temporal dynamics and fluid motion into the capturing system when adopting them onto any such video-based ASL recognition systems, potentially the most seamless and successful way to tackle this question. Future studies could investigate such systems to observe motion across time for greater accuracy. Hybrid architectures such as pre-trained transformers and cross-lingual transfer will be able to expand the capabilities of these systems even further, allowing them to be adapted to other forms of ASL data. Scalable, multilingual, something multiplication is; Evaluating language models will also be critical to ensure inclusive, robust, real-world deployment. The convergence of all these avenues fosters the overarching goal of developing gesture-based communication systems that are adaptable, inclusive, and accurate for users with varied needs. Future work aims to deploy such compressed and optimised PBN models on target embedded devices to hasten the application of such models in healthcare and assistive contexts, thus enhancing the scalability of the system for real-world usage (Fig. 12).

Acknowledgements This work was supported by Institute of Information communications Technology Planning Evaluation (IITP) under the metaverse support program to nurture the best talents (IITP-2024-RS-2023-00254529) grant funded by the Korea government(MSIT); by Institute of Information communications Technology Planning Evaluation (IITP) grant funded by the Korea government(MSIT) (RS-2025-25443732, Research on Ethical Reasoning and Metacognition for Human-Aligned AGI); and by the Institute of Information & Communications Technology Planning & Evaluation (IITP) under the Metaverse Support Program to Nurture the Best Talents (IITP-2023-RS-2023-00254529) grant funded by the Korea government (MSIT). KATEX COED.

Author contributions A.K.: writing—review & editing, writing—original draft, visualization, validation, software, resources, methodology, investigation, formal analysis, data curation, conceptualization. G.-H.L.: writing—original draft, software, resources, methodology, investigation, formal analysis, data curation, conceptualization. L.M.D.: writing—review & editing, writing—original draft, resources, investigation, formal analysis. S.U.K.: writing—review & editing, supervision, methodology, conceptualization. M.A.K.: writing—review & editing, supervision, methodology, conceptualization. W.C.: writing—review & editing, resources, investigation, formal analysis, conceptualization. H.M.: writing—review & editing, writing—original draft, supervision, project administration, methodology, investigation, funding acquisition, formal analysis.

Data availability Data will be made available on request.

Declarations

Competing interests The authors declare no competing interests.

References

1. Harvey M, Brazier D (2022) E-government information search by english-as-a-second-language speakers: the effects of language proficiency and document reading level. Inf Process Manage 59(4):102985
2. Liu Y, Xue J, Li D, Zhang W, Chiew TK, Xu Z (2024) Image recognition based on lightweight convolutional neural network: recent advances. Image Vision Comput 146:105037
3. Patel AN, Murugan R, Maddikunta PKR, Yenduri G, Jhaveri RH, Zhu Y, Gadekallu TR (2024) Ai-powered trustable and explainable fall detection system using transfer learning. Image Vision Comput 149:105164
4. Pleva M, Liao Y-F, Bours P (2022) Human–computer interaction for intelligent systems. Electronics 12(1):161
5. Feng Y, Chen N, Wu Y, Jiang C, Liu S, Chen S (2024) Dfcnet+: cross-modal dynamic feature contrast net for continuous sign language recognition. Image Vision Comput 151:105260
6. Lv Z, Poiesi F, Dong Qi, Lloret J, Song H (2022) Deep learning for intelligent human–computer interaction. Appl Sci 12(22):11457
7. Madhiarasan M, Roy P and Pratim P (2022) A comprehensive review of sign language recognition: Different types, modalities, and datasets. arXiv preprint arXiv:2204.03328
8. Lengkeek M, van der Knaap F, Frasincar F (2023) Leveraging hierarchical language models for aspect-based sentiment analysis on financial data. Inf Process Manage 60(5):103435
9. Das HV, Mohan K, Paul L, Kumaresan S, Nair CS (2024) Transforming consulting atmosphere with Indian sign language translation. Multimed Tools Appl 83(5):13543–13555
10. Lin S, Xiao Z, Wang L, Wan X, Ni L, Fang Y (2024) Structure-aware sign language recognition with spatial– temporal scene graph. Inf Process Manage 61(6):103850
11. Yao H, Wang L, Cai C, Wang W, Zhang Z, Shang X (2024) Language conditioned multi-scale visual attention networks for visual grounding. Image Vis Comput 150:105242
12. Wang Y-H, Su W-H (2022) Convolutional neural networks in computer vision for grain crop phenotyping: a review. Agronomy 12(11):2659

13. Kumar A, Rathore S and Singla N (2024) American sign language recognition with text-to-speech
14. Aly M, Fathi IS (2025) Recognizing American sign language gestures efficiently and accurately using a hybrid transformer model. Sci Reports 15(1):20253
15. Carneiro ALC, Salvadeo DHP and de Brito Silva L (2024) Sign language recognition based on deep learning and low-cost handcrafted descriptors. arXiv preprint arXiv:2408.07244
16. Sumit Kumar, Ruchi Rani, and Ulka Chaudhari. Real-time sign language detection: Empowering the disabled community. MethodsX, p 102901, 2024
17. Chowdhury PK, Oyshe KU, Rahaman MA, Debnath T, Rahman A, Kumar N (2024) Computer vision-based hybrid efficient convolution for isolated dynamic sign language recognition. Neural Comput Appl 36(32):19951–19966
18. Seong H, Cho H (2024) Three-dimensional convolutional vision transformer for sign language translation. Trans Korea Inf Process Soc 13(3):140–147
19. Rahim MA, Miah ASM, Akash HS, Shin J, Hossain MI, and Hossain MN (2024) An advanced deep learning based three-stream hybrid model for dynamic hand gesture recognition. arXiv preprint arXiv:2408.08035. https://doi.org/10.48550/arXiv.2408.08035
20. Kim A (2024) Augmentation of sign language poses by including the understanding of the sign language domain by body part. Retrieved from http://hdl.handle.net/10203/321793
21. Rezaee K, Khavari SF, Ansari M, Zare F, Roknabadi MHA (2024) Hand gestures classification of semg signals based on bilstm-metaheuristic optimization and hybrid u-net-mobilenetv2 encoder architecture. Sci Reports 14(1):31257
22. Singh SK, Chaturvedi A (2023) A reliable and efficient machine learning pipeline for American sign language gesture recognition using emg sensors. Multimedia Tools Appl 82(15):23833–23871
23. Cheok MJ, Omar Z, Jaward MH (2019) A review of hand gesture and sign language recognition techniques. In J Mach Learn Cybern 10:131–153
24. Dey A, Biswas S, Le D-N (2024) Recognition of wh-question sign gestures in video streams using an attention driven c3d-bilstm network. Procedia Comput Sci 235:2920–2931
25. Dey A, Biswas S, Abualigah L (2024) Umpire's signal recognition in cricket using an attention based dc-gru network. Int J Eng 37(4):662–674
26. Tripathi A and Katiyar S (2024) Sign sense: a sign language recognition system for empowering individuals with disabilities. Bachelor of technology project report, Jaypee University of Information Technology, Waknaghat, Solan - 173234, India
27. Leiva V, Rahman MZU, Akbar MA, Castro C, Huerta M and Riaz MT (2025) A real-time intelligent system based on machine-learning methods for improving communication in sign language. IEEE Access
28. Ahammad K, Shawon JAB, Chakraborty P, Islam MJ and Islam S (2021) Recognizing bengali sign language gestures for digits in real time using convolutional neural network. Int J Comput Sci Inf Secur 19(1)
29. Kothadiya D, Bhatt C, Sapariya K, Patel K, Gil-González A-B, Corchado JM (2022) Deepsign: Sign language detection and recognition using deep learning. Electronics 11(11):1780
30. Nasir M, Musri T and Kurniawaty E (2023) Hand gesture recognition using leap motion controller for recognition of Javanese script. Int ABEC 24–128
31. Asiri OI, Elfaki AO, Abushaira ME (2024) Investigating the use of leap motion controller in recognition of Arabic sign language. Educ Adm Theory Pract 30(5):1245–1254
32. Faisal M, Singh A and Singh SN (2024) A review of real-time sign language recognition for virtual interaction on meeting platforms. In: 2024 14th International Conference on Cloud Computing, Data Science & Engineering (Confluence), pp 364–369. IEEE
33. Ganesh S, Akash R and VijayKumar K (2024) Leap motion robot maneuver in midair for multi-finger control system. In: 2024 2nd International Conference on Intelligent Data Communication Technologies and Internet of Things (IDCIoT), pp 1151–1158. IEEE
34. Li Y, Lou J, Cai Z, Zheng P, Haijun Wu, Wang X (2024) An interactive gesture control system for collaborative manipulator based on leap motion controller. Adv Mech Eng 16(5):16878132241253100
35. Galván-Ruiz J, Travieso-González CM, Pinan-Roescher A, Alonso-Hernández JB (2023) Robust identification system for Spanish sign language based on three-dimensional frame information. Sensors 23(1):481
36. Umut İ and Kumdereli ÜC (2024) Novel wearable system to recognize sign language in real time
37. Han Y, Han Y, Jiang Qi (2025) A study on the stgcn-lstm sign language recognition model based on phonological features of sign language. IEEE Access. https://doi.org/10.1109/access.2025.3560779
38. Rodriguez M, Oubram O, Bassam A, Lakouari N, Tariq R (2025) Mexican sign language recognition: dataset creation and performance evaluation using mediapipe and machine learning techniques. Electronics 14(7):1423
39. Chakravarthi B, Prasad P, Imandi R and Pavan Kumar BN (2023) A comprehensive review of leap motion controller-based hand gesture datasets. In: Proceedings of the 2023 International Conference on Next Generation Electronics (NEleX), pp 1–7. IEEE
40. Vijaya Saraswathi R, Devulapally M, Narsingh SR, Temberveni H, Katta NN (2025) Optical motion detection language generator: a survey. Procedia Comput Sci 252:90–99
41. Myagila K, Nyambo DG, Dida MA (2025) Efficient spatio-temporal modeling for sign language recognition using cnn and rnn architectures. Front Artif Intell 8:1630743
42. Tian Y, Su J, Ni L and Fang Y (2024) Bridging the gap: Ai and sign language recognition—a path toward inclusive communication
43. Sahm BA, Al-Fahaam H and Jasim AA (2025) Deep learning based dynamic sign language translation system. Int J Inf Technol 1–18
44. Enikeev DG, Mustafina SA (2021) Sign language recognition through leap motion controller and input prediction algorithm. J Phys Conf Ser 1715:012008 (**IOP Publishing**)
45. Zhou J, Tian X (2023) Mkl-sing: a data-driven approach of sign recognition for managing and improving public services. Inf Process Manage 60(3):103243
46. Zhu X, Liu Z, Cambria E, Xiaohan Yu, Fan X, Chen H, Wang R (2025) A client–server based recognition system: non-contact single/multiple emotional and behavioral state assessment methods. Comput Methods Programs Biomed 260:108564
47. Chevtchenko SérgioF, Vale RF, Macario V, Cordeiro FR (2018) A convolutional neural network with feature fusion for real-time hand posture recognition. Appl Soft Comput 73:748–766
48. Makarov I, Veldyaykin N, Chertkov M and Pokoev A (2019) American and russian sign language dactyl recognition. In: Proceedings of the 12th ACM International Conference on PErvasive Technologies Related to Assistive Environments, pp 204–210
49. Rastgoo R, Kiani K, Escalera S (2018) Multi-modal deep hand sign language recognition in still images using restricted boltzmann machine. Entropy 20(11):809
50. Rathi P, Gupta RK, Agarwal S and Shukla A (2020) Sign language recognition using resnet50 deep neural network architecture. In: 5th International Conference on Next Generation Computing Technologies (NGCT-2019)
51. Marin G, Dominio F and Zanuttigh P (2014) Hand gesture recognition with leap motion and kinect devices. In: 2014 IEEE

International Conference on Image Processing (ICIP), pp 1565–1569. IEEE
52. Funasaka M, Ishikawa Y, Takata M and Joe K (2015) Sign language recognition using leap motion controller. In: Proceedings of the International Conference on Parallel and Distributed Processing Techniques and Applications (PDPTA), p 263. The Steering Committee of The World Congress in Computer Science, Computer Engineering and Applied Computing
53. Mapari RB and Kharat G (2016) American static signs recognition using leap motion sensor. In: Proceedings of the Second International Conference on Information and Communication Technology for Competitive Strategies, pp 1–5. Association for Computing Machinery
54. Vaitkevičius A, Taroza M, Blažauskas T, Damaševičius R, Maskeliu¯nas R, Woźniak M (2019) Recognition of American sign language gestures in a virtual reality using leap motion. Appl Sci 9(3):445
55. Chong T-W, Lee B-G (2018) American sign language recognition using leap motion controller with machine learning approach. Sensors 18(10):3554
56. Elons AS, Ahmed M, Shedid H and Tolba MF (2014) Arabic sign language recognition using leap motion sensor. In: 2014 9th International Conference on Computer Engineering & Systems (ICCES), pp 368–373. IEEE
57. Jia W, Li C (2024) Slr-yolo: an improved yolov8 network for real-time sign language recognition. J Intell Fuzzy Syst 46(1):1663–1680
58. Paul SK, Walid MAA, Paul RR, Uddin MJ, Rana MS, Devnath MK, Dipu IR, Haque MM (2024) An adam based CNN and LSTM approach for sign language recognition in real time for deaf people. Bull Electr Eng Inform 13(1):499–509