Journal of Electronics, Electromedical Engineering, and Medical Informatics
Multidisciplinary: Rapid Review: Open Access Journal **Vol. 7, No. 2, April 2025, pp: 318-329; eISSN: 2656-8632**

RESEARCH ARTICLE **OPEN ACCESS**

Manuscript received January 05, 2025; revised February 20, 2025; date of publication February 24, 2025
Digital Object Identifier (**DOI**): https://doi.org/10.35882/jeeemi.v7i2.685

How to cite: Hemant Kumar, Rishabh Sachan, Mamta Tiwari, Amit Kumar Katiyar, Namita Awasthi, Pushpa Mamoria and Ramnayn Mishra, "Hybrid Sign Language Recognition Framework Leveraging MobileNetV3, Mult-Head Self Attention and LightGBM", Journal of Electronics, Electromedical Engineering, and Medical Informatics, vol. 7, no. 2, pp. 318-329, April 2025.

Hybrid Sign Language Recognition Framework Leveraging MobileNetV3, Mult-Head Self Attention and LightGBM

Hemant Kumar[1*], Rishabh Sachan[2], Mamta Tiwari[3], Amit Kumar Katiyar[4], Namita Awasthi[5], Pushpa Mamoria[3] and Ramnayn Mishra[1]

[1] Department of Information Technology, Chhatrapati Shahu Ji Maharaj University, Kanpur, India.
[2] Department of Computer Science and Engineering (AI), KIET Group of Institutions, Ghaziabad, India.
[3] Department of Computer Applications, Chhatrapati Shahu Ji Maharaj University, Kanpur, India.
[4] Department of Electronics and Communication Engineering, Chhatrapati Shahu Ji Maharaj University, Kanpur, India.
[5] Department of Computer Science & Engineering, Allenhouse Institute of Technology, Kanpur, India.

Corresponding author: Hemant Kumar (e-mail: hemantime@gmail.com).

ABSTRACT Sign-language recognition (SLR) plays a pivotal role in enhancing communication accessibility and fostering the inclusion of deaf communities. Despite significant advancements in SLR systems, challenges such as variability in sign language gestures, the need for real-time processing, and the complexity of capturing spatiotemporal dependencies remain unresolved. This study aims to address these limitations by proposing an advanced framework that integrates deep learning and machine learning techniques to optimize sign language recognition systems, with a focus on the Indian Sign Language (ISL) dataset. The framework leverages MobileNetV3 for feature extraction, which is selected after rigorous evaluation against VGG16, ResNet50, and EfficientNet-B0. MobileNetV3 demonstrates superior accuracy and efficiency, making it optimal for this task. To enhance the model's ability to capture complex dependencies and contextual information, multi-head self-attention (MHSA) was incorporated. This process enriches the extracted features, enabling a better understanding of sign language gestures. Finally, LightGBM, a gradient-boosting algorithm that is efficient for large-scale datasets, was employed for classification. The proposed framework achieved remarkable results, with a test accuracy of 98.42%, precision of 98.19%, recall of 98.81%, and an F1-score of 98.15%. The integration of MobileNetV3, MHSA, and LightGBM offers a robust and adaptable solution that outperforms the existing methods, demonstrating its potential for real-world deployment. In conclusion, this study advances precise and accessible communication technologies for deaf individuals, contributing to more inclusive and effective human-computer interaction systems. The proposed framework represents a significant step forward in SLR research by addressing the challenges of variability, real-time processing, and spatiotemporal dependency. Future work will expand the dataset to include more diverse gestures and environmental conditions and explore cross-lingual adaptations to enhance the model's applicability and impact.

INDEX TERMS Sign Language Recognition, Gesture Recognition, MobileNetV3, Multi-head Self-Attention (MHSA), LightGBM, Indian Sign Language (ISL).

I. INTRODUCTION

SLR remains an essential domain in computer vision and machine learning, aiming to enhance communication accessibility and inclusion for deaf individuals [1]. Developing efficient recognition systems is crucial for minimizing communication barriers and facilitating seamless interactions between deaf communities and a broader society [2]. Sign languages inherently comprise complex components, including manual gestures, facial expressions, and body movements, each of which contains significant semantic and syntactic information that necessitates advanced analytical

methods for accurate interpretation. Traditional modes of communication for Deaf individuals typically depend on human interpreters or direct engagement with other sign language users. However, these methods are impractical in contexts where remote communication or automated systems are required [3]. Automated systems capable of instantly recognizing and translating sign languages can significantly enhance accessibility and foster inclusivity [4].

Despite these advancements, building robust sign-language recognition systems poses several challenges. Variations in signing style, speed, and context among individuals can affect the consistency of sign language gestures. Additionally, regional and dialectical differences introduce further complexity, requiring models that generalize effectively across a wide array of signing styles, while maintaining high accuracy [5]. Another challenge pertains to the dynamic nature of sign language gestures, which encompass manual motions, facial expressions, and body positions that require precise recording and interpretation. Accurately capturing these elements requires models that effectively integrate both spatial and temporal information [6]. Advanced models must address the chronological progression of gestures, while accurately representing the spatial features of hand movements and facial expressions.

Data availability and quality are crucial factors in advancing sign language recognition systems. Developing high-quality, labelled datasets for training and testing machine learning models presents a significant challenge, as it requires substantial effort in data collection and annotation, which can be resource intensive [7]. The scarcity of comprehensive and diverse datasets limits the development of models capable of accurately recognizing different sign languages and scenarios, thereby underscoring the need for ongoing efforts in data acquisition and annotation. Real-time processing is crucial for sign language recognition systems to be practical. Systems designed for real-time translation or communication assistance must operate with low latency, which adds further constraints to the model design and implementation. Balancing accuracy and computational efficiency is essential to ensure that the models perform effectively within the required time frames [8]. Consequently, research in this domain is increasingly focusing on creating models that maintain accuracy and efficiency while optimally managing computational resources.

Sign language recognition (SLR) has emerged as a critical domain within the realms of computer vision and ML, exhibiting notable advancements while also facing significant challenges. A multi-tier framework that amalgamates CNNs with LSTM networks achieved a recognition precision of 98.8% on established benchmark datasets, thereby accentuating the efficacy of these computational structures [9]. An alternative methodology utilizes CNNs in conjunction with image processing techniques, such as the Histogram of Oriented Gradients, to facilitate precise, real-time gesture detection, thereby contributing to inclusivity and accessibility initiatives [10]. Despite these advancements, challenges such as the necessity for real-time processing and the absence of standardized representations of sign language continue to persist [11].

The complexity of models significantly influences performance, as evidenced by the inflated 3D model, which exhibits enhanced word recognition from video frames, indicating a correlation between increased complexity and improved accuracy [1]. The YOLOv5 architecture was implemented to achieve robust real-time sign language interpretation, with mean Average Precision (mAP) values fluctuating between 92% and 99%, demonstrating potential applicability within dynamic settings. Although advancements are apparent, additional investigations are warranted to refine generalization and real-time functionalities, thereby fostering broader societal inclusivity in individuals with hearing impairments [12].

Developments in attention-based methodologies for SLR have further enhanced gesture recognition precision. The Intra-inter Gloss Attention model utilizes localized self-attention to mitigate complexity and noise, achieving a competitive word error rate of 20.4% [13]. Hybrid CNN-LSTM frameworks augmented with attention mechanisms have demonstrated efficacy, achieving an average accuracy of 84.65% on the WLASL dataset, indicative of their efficiency and potential for further enhancement [14]. Models that concentrate on hand shapes and motion trajectories, such as the Top-Down Attention model, have surpassed state-of-the-art methodologies on extensive datasets [15]. Nevertheless, challenges pertaining to real-time applications persist, as illustrated by the 3D-CNN system that integrates an attention mechanism, achieving an accuracy of 98.49% [16].

To provide a more comprehensive rationale for the selection of MobileNetV3, MHSA, and LightGBM, we emphasize their combined advantages in sign language contexts: MobileNetV3 exhibits lightweight and efficient properties suitable for on-device or real-time applications; MHSA incorporates the capacity to capture spatial and context-specific dependencies crucial in gesture interpretation; and LightGBM offers robust and scalable classification with rapid convergence.

This study proposes a novel approach that combines MobileNetV3 [17] with MHSA [18] and LightGBM [19] to enhance sign language recognition. MobileNetV3 serves as the feature extractor, optimizing the computational efficiency and performance, as demonstrated in prior studies. MHSA enhances the model's capacity to capture intricate relationships within the extracted features, whereas LightGBM integrates advanced methodologies to improve classification accuracy and model robustness. The core contributions of this study can be summarized as follows:

1) Development of an efficient feature extraction pipeline utilizing MobileNetV3, focusing on maximizing the accuracy of sign language recognition.

2) The MHSA was utilized to improve the feature representation and capture intricate dependencies within the data.
3) The optimized LightGBM classifier was used to enhance the classification accuracy and ensure the reliability of the recognition system.

The remainder of this paper is organized as follows. Section 2 describes the methodology, MobileNetV3 architecture, MHSA integration, and use of LightGBM for classification. Section 3 details the experimental setup and dataset. Section 4 discusses the results and analyses. Finally, Section 5 concludes the paper.

II. MATERIAL AND METHODS

This section describes the methodologies employed to develop an effective sign-language recognition system. The proposed framework integrates MobileNetV3 [17] for feature extraction, multihead self-attention [18] mechanisms to enhance feature representations, and a LightGBM classifier to achieve precise classification [19]. The architecture of the proposed SLR system comprises four primary components

1) Data Preprocessing and Augmentation: The dataset undergoes a series of preprocessing steps, including cleaning, normalization, and augmentation, applied to the images to enhance the model's performance.
2) Feature Extraction using MobileNetV3: MobileNetV3 is employed to extract robust and discriminative features from preprocessed images, leveraging its efficient architecture tailored for mobile and edge devices [17].
3) Feature Enhancement using Multihead Self-Attention: The extracted features are subsequently refined through multihead self-attention mechanisms, which capture complex interdependencies and contextual information within the feature space [18].
4) Classification with LightGBM: Finally, the enhanced features are classified using a LightGBM classifier, which contributes to an improved accuracy and robustness in the recognition process [19].

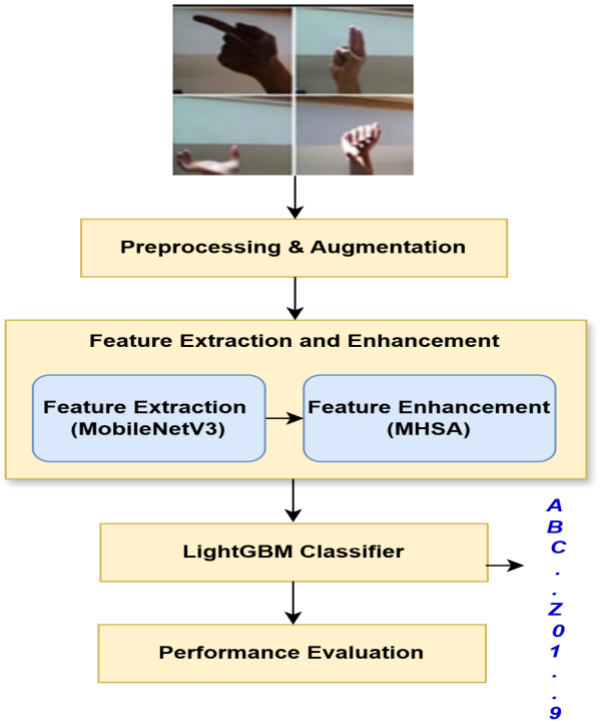

FIGURE 1. Hybrid Sign Language Recognition Model.

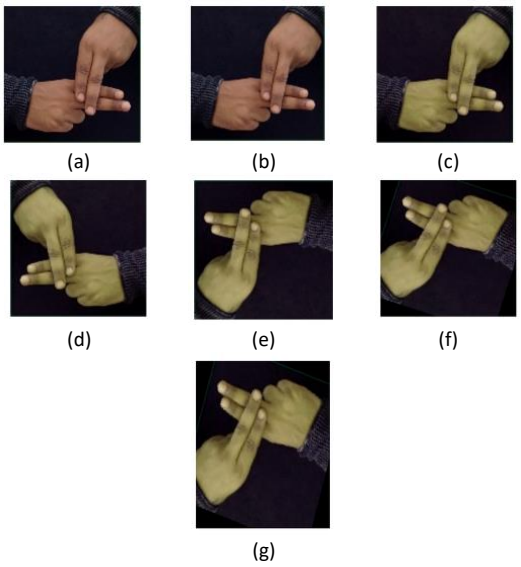

FIGURE 2. (a) the original image, (b) resizing, (c) color jitter, (d) random horizontal flip, (e) random vertical flip, (f) random rotation, (g) conversion to a tensor.

FIGURE 1 illustrates the architecture of the proposed system. The input images were processed using MobileNetV3 to extract pertinent features. These features were passed through a multihead self-attention module to incorporate contextual information. The refined features were then fed into the LightGBM classifier for the final classification of sign language gestures.

A. PREPROCESSING

Preprocessing techniques were applied to the sign-language image dataset to ensure consistency and enhance the performance of the recognition system. The preprocessing pipeline included image scaling, normalization, and augmentation. Initially, the images were uniformly scaled to comply with the input specifications of MobileNetV3, specifically resizing them to 224 × 224 pixels. Following scaling, the pixel values were normalized to facilitate efficient training [20].

In our data augmentation pipeline, images were first uniformly resized to 224 × 224 pixels to ensure consistent input dimensions and facilitate batching. We subsequently applied ColorJitter to introduce slight variations in brightness, contrast, saturation, and hue, which aided the model in learning robust color-invariant features. Next, random

horizontal and vertical flips were implemented to introduce spatial diversity by mirroring the image, while random rotation up to 35° was employed to mitigate the sensitivity to object orientation. Each image was converted into a PyTorch tensor to standardize the data format, after which normalization was applied with the mean and standard deviation values. These data augmentation techniques collectively expand the effective size of the training set, reduce overfitting, and enhance the generalization capability of the model. FIGURE 2 illustrates the sequential data augmentation steps applied to an ISL image.

B. MOBILENETV3 ARCHITECTURE

MobileNetV3 [17] was optimized for both mobile and edge devices. Building on the foundations of MobileNetV2, MobileNetV3 incorporates novel architectural elements that enhance performance without significantly compromising computational efficiency. The MobileNetV3 architecture is shown in FIGURE 3.

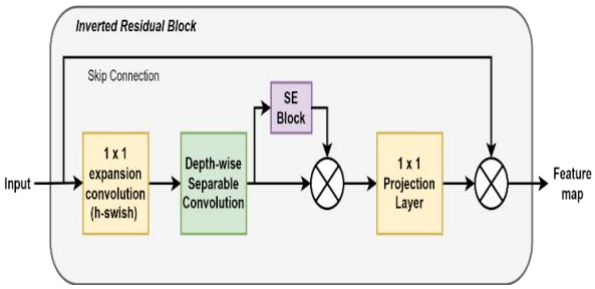

FIGURE 3. MobileNetV3 Architecture.

FIGURE 4(a) illustrates the depth-wise separable convolution, while FIGURE 4(b) demonstrates the functionality of squeeze-and-excitation (SE) blocks, which are integral to the MobileNetV3 architecture.

1) INVERTED RESIDUAL BLOCKS

The inverted residual blocks are fundamental to the MobileNetV3 architecture. These blocks employ a bottleneck structure in which the input is first expanded into a higher-dimensional space, processed through depth-wise separable convolutions, and subsequently projected back to a lower-dimensional space. This approach contrasts with traditional residual blocks, which directly process the input. The utilization of inverted residuals facilitates more efficient feature extraction, enabling the capture of intricate patterns without a substantial increase in the computational overhead. Additionally, skip connections within these blocks enhance computational efficiency by allowing the direct addition of inputs to outputs.

2) DEPTH-WISE SEPARABLE CONVOLUTIONS

Depth-wise separable convolutions are employed to reduce the computational costs and model complexity by decomposing standard convolutions into depthwise and pointwise operations.

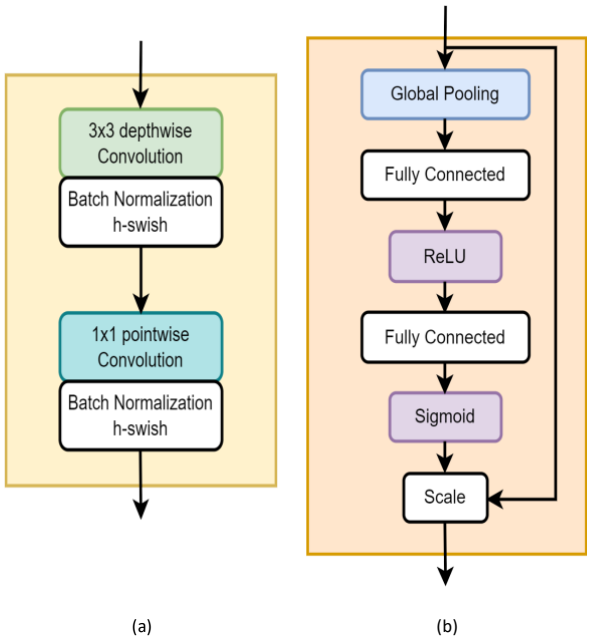

FIGURE 4. (a) Depthwise Separable Convolution, (b) Squeeze-and-Excitation (SE) Blocks

Depth-wise Convolution: This operation involves applying a single convolutional filter to each input channel independently. If X is the input tensor with dimensions (H, W, C) (height, width, channels), and K is a is a depthwise convolutional kernel with dimensions (k, k, C), then the depthwise convolution process is as follows Eq. (1) [17]

$$Y_{i,j,k} = \sum_{m,n} X_{i+m,j+n,k} \cdot K_{m,n,k} \quad (1)$$

Pointwise Convolution: Following depthwise convolution, a 1×1 convolution is applied across all channels. If X is the tensor resulting from the depthwise layer with shape (H', W', C), and K is a pointwise kernel with shape $(1, 1, C, C')$, the pointwise convolution operation is as follows Eq. (2) [17]

$$Y_{i,j,k'} = \sum_{c} X_{i,j,c} \cdot K_{1,1,c,k'} \quad (2)$$

where Y is the final output tensor of the shape (H', W', C').

3) SQUEEZE-AND-EXCITATION (SE) BLOCKS

SE [23] blocks dynamically modulate channel-specific feature responses, thereby enhancing the representational capacity of the network. For a feature map X with C channels, the SE block performs two primary operations.

Squeeze: Aggregates spatial information into a channel descriptor z by performing global average pooling is as follows Eq. (3) [23]

$$z_c = \frac{1}{H \cdot W} \sum_{i=1}^{H} \sum_{j=1}^{W} X_{i,j,c} \quad (3)$$

Excitation: Utilizes a fully connected neural network to model channel-wise dependencies, producing a scaling vector s is as follows Eq. (4) [23]

$$s_c = \sigma(W_2 \cdot \delta(W_1 \cdot z + b_1) + b_2) \quad (4)$$

where δ is ReLU and σ is sigmoid activation function, and W_1, W_2, b_1 and b_2 are learned parameters.

C. MULTI-HEAD SELF ATTENTION (MHSA)

MHSA enables the model to attend to different segments of the input feature map concurrently, thereby capturing a comprehensive range of contextual information [18]. MHSA captures long-range dependencies and intricate relationships within the feature space, which are critical for distinguishing highly similar gestures that vary in subtle spatial details. The features extracted from the previous steps were passed to the MHSA layer to enhance the features. Extracted feature tensor X with shape (N, T, D), where N is the batch size, T is the sequence length, and D is the dimensionality of each feature vector. The input tensor is projected into query Q, key K, and value V matrices using learned weight matrices is represented by Eq. (5) [18]

$$Q = XW_Q, \quad K = XW_K, \quad V = XW_V \tag{5}$$

where W_Q, W_K , and W_V are weight matrices with shapes $(D, d_q), (D, d_k)$, and (D, d_v), respectively, and d_k and d_v are the dimensions of the key and value vectors. The attention output using the query, key, and value vectors is given by Eq. (6) [18]

$$Attention(Q, K, V) = Softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{6}$$

where $\sqrt{d_k}$ serves as the scaling factor, and Softmax normalizes the scores. The outputs from multiple attention heads are concatenated within the MHSA layer, and the amalgamating features from the diverse representation subspaces is represented by Eq. (7) [18]

$$MultiHead(Q, K, V) = Concat(h_1, h_2, \dots, h_h)W_O \tag{7}$$

where each head is $h_i = Attention\left(QW_{Q_i}, KW_{K_i}, VW_{V_i}\right)$ and W_O is the output projection matrix. In our study, we used four attention heads with eight layers, and a dropout rate of 0.2. The ability of the self-attention mechanism to evaluate the significance of diverse image regions was harnessed by this structural design, facilitating a refined analysis of the input data. Through the integration and synthesis of multiple attention head outputs, the model discerns more elaborate and subtle correlations within the information. Subsequent processing via feedforward networks and normalization layers further refines these representations, ensuring a comprehensive understanding of the input sequence. This thorough approach ultimately yields more accurate final predictions by enabling the model to develop a nuanced understanding of the data [24].

D. LIGHTGBM

LightGBM [19] is a gradient-boosting framework renowned for its efficiency and scalability in handling large-scale datasets. LightGBM distinguishes itself by its ability to train models more rapidly and with lower memory consumption than traditional gradient boosting methods. The framework incorporates several innovative techniques, including leafwise tree growth, Exclusive Feature Bundling (EFB), and Gradient-based One-Side Sampling (GOSS), to enhance both performance and accuracy. LightGBM constructs an ensemble of decision trees in a stage-wise manner to minimize a specified loss function. Each subsequent tree was added to the model to correct the errors made by the preceding trees, with the final prediction being an aggregation of the outputs from all individual trees. In LightGBM, model at the t-th iteration is as follows Eq. (8) [19].

$$F_t(X) = F_{t-1}(X) + \eta h_t(X) \tag{8}$$

where $F_{t-1}(X)$ is the model from the previous iteration, $h_t(X)$ is the new tree added at iteration t, and η is the learning rate. LightGBM employs a leaf-wise growth strategy, as opposed to the traditional level-wise approach. This strategy selects the leaf with the maximum loss reduction for splitting, thereby resulting in deeper and potentially more accurate trees. Leaf-wise growth involves selecting the leaf l^* that maximizes the gain ΔL, represented by Eq. (9) [21]

$$l^* = arg\max_{l \in L} \Delta L_l \tag{9}$$

where ΔL_l is the reduction in loss by splitting leaf *l*. A notable feature of LightGBM is GOSS, enhances the training efficiency by retaining all instances with large gradients while performing random sampling on instances with small gradients. This approach preserves the accuracy of the information gain estimation while reducing the number of data instances to be processed. To efficiently manage datasets with a vast number of features, LightGBM employs EFB, consolidates mutually exclusive features, and consolidates those that do not take non-zero values simultaneously into single features. This technique significantly reduces the dimensionality of the dataset without compromising the essential information. The objective function in LightGBM incorporates regularization to control model complexity, thereby preventing overfitting. The objective function is expressed by Eq. (10) [19]

$$L = \sum_{i=1}^{n} l(y_i, \hat{y}_i) + \sum_{t=1}^{T} \Omega(h_t) \tag{10}$$

where $l(y_i, \hat{y}_i)$ is the loss function and $\Omega(h_t)$ is the regularization term for tree h_t. The regularization term is given by Eq. (11) [22]

$$\Omega(h) = \gamma T + \frac{1}{2}\lambda \sum_{j=1}^{T} w_j^2 \tag{11}$$

where γ is the penalty for adding a new leaf, T is the number of leaves in the tree, λ is the L2 regularization term on leaf weights, and w_j are the leaf weights [22]. During training, the LightGBM employs a second-order Taylor expansion to approximate the loss function, facilitating more accurate and efficient optimization. The optimization objective at iteration t is represented by Eq. (12) [19]

$$L^{(t)} \approx \sum_{i=1}^{n} \left[l\left(y_i, \hat{y}_i^{(t-1)}\right) + g_i h_t(x_i) + \frac{1}{2} h_i h_t^2(x_i) \right] + \Omega(h_t) \tag{12}$$

where g_i and h_i are gradients of the loss function with respect to the predictions. We conducted a stratified 5-fold cross-validation to optimize the key LightGBM hyperparameters (learning rate, max_depth, subsampling) in the ranges $\eta \in \{0.001, 0.01\}$, $\max\_depth\{4 \ldots 10\}$ and $subsample \in [0.5,1.0]$. LightGBM exhibits expeditious training and robust performance, leveraging leaf-wise tree growth and sophisticated sampling techniques (GOSS and EFB). In our comparative evaluation, LightGBM outperformed other gradient boosting classifiers (CatBoost and XGBoost) and traditional models (Random Forest, SVM) [19, 22].

III. EXPERIMENTAL SETUPS

A. DATASETS

The dataset utilized in this study comprises a total 42745 images representing 35 distinct categories of Indian Sign Language (ISL) [25]. These categories include numerals 1-9 and letters A-Z, covering a comprehensive range of gestures used in the ISL. Each class contains 1200 images, with slight variations in the number of images for certain classes. Specifically, classes C, I, and O contained 1447, 1379, and 1429 images, respectively. TABLE 1 provides a summary of the dataset. This dataset provides a clear and concise overview of its composition, preprocessing steps, and splitting criteria, ensuring transparency and reproducibility.

TABLE 1
Description of ISL dataset

Attribute	Details
Total Images	42745
Classes	35 (Numerals 1-9 and Letters A-Z)
Images per class	~1200 (C: 1447, I: 1379, O: 1429
Train Set	34196 (80% of total dataset)
Validation Set	8749 20% of total dataset)

B. IMPLEMENTATION DETAILS

The proposed pipeline was implemented on a Linux-based platform equipped with 16 GB of RAM and an 8GB NVIDIA RTX 4060 GPU. We used Python 3.9 with PyTorch (v1.12) for the deep learning modules and LightGBM (v3.3) for gradient boosting. The training process leveraged the Indian Sign Language (ISL) dataset and employed MobileNetV3 for feature extraction because of its optimal balance between computational efficiency and accuracy. To further refine the feature representations, an MHSA layer with four attention heads and a dropout rate of 0.2 was integrated into the MobileNetV3 architecture. This integration enables the model to capture complex dependencies and contextual information, thereby improving the quality of the extracted features prior to classification. The Adam optimizer was employed with a learning rate of 0.001 and batch size of 32 for both MobileNetV3 and the MHSA modules.

The model begins with an initial convolution followed by batch normalization and HardSwish activation, and then proceeds through a series of inverted residual blocks. Each inverted residual block progressively refines the spatial dimensions and channel depth, which are reflected in the changing output shapes. An attention module appears near the end, applying MHSA over a feature space of size [1, 960, …], presumably enabling the network to focus on important regions. Finally, the model applies adaptive average pooling, followed by several linear activation layers. The total number of parameters is approximately 8.51 million, all of which are trainable. TABLE 2 provides a summary of the proposed model.

TABLE 2
Summary of the proposed model

Layer (type: depth - idx)	Output Shape	Param #
MobileNetV3WithAttention	[1, 960]	--
Sequential: 1 - 1	[1, 960, 7, 7]	--
Conv2dNormActivation: 2 -1	[1, 16, 112, 112]	--
Conv2d: 3 - 1	[1, 16, 112, 112]	432
BatchNorm2d: 3 - 2	[1, 16, 112, 112]	32
Hardswish: 3 - 3	[1, 16, 112, 112]	--
InvertedResidual: 2 - 2	[1, 16, 112, 112]	--
Sequential: 3 - 4	[1, 16, 112, 112]	464
InvertedResidual : 2 - 3	[1, 24, 56, 56]	--
Sequential: 3 - 5	[1, 24, 56, 56]	3, 440
InvertedResidual : 2 - 4	[1, 24, 56, 56]	--
Sequential: 3 - 6	[1, 24, 56, 56]	4, 440
InvertedResidual : 2 - 5	[1, 24, 56, 28]	--
Sequential: 3 - 7	[1, 24, 56, 28]	10, 328
InvertedResidual : 2 - 6	[1, 24, 56, 28]	--
Sequential: 3 - 8	[1, 24, 56, 28]	20, 992
InvertedResidual : 2 - 7	[1, 24, 56, 28]	--
Sequential: 3 - 9	[1, 24, 56, 28]	20, 992
InvertedResidual : 2 - 8	[1, 80, 14, 14]	--
Sequential: 3 - 10	[1, 80, 14, 14]	32, 080
InvertedResidual : 2 - 9	[1, 80, 14, 14]	--
Sequential: 3 - 12	[1, 80, 14, 14]	34, 760
InvertedResidual : 2 - 10	[1, 80, 14, 14]	--
Sequential: 3 - 12	[1, 80, 14, 14]	31, 992
InvertedResidual : 2 - 11	[1, 80, 14, 14]	--
Sequential: 3 - 13	[1, 80, 14, 14]	31, 992
InvertedResidual : 2 - 12	[1, 112, 14, 14]	--
Sequential: 3 - 14	[1, 112, 14, 14]	214, 424
InvertedResidual : 2 - 13	[1, 112, 14, 14]	--
Sequential: 3 - 15	[1, 112, 14, 14]	386, 120
InvertedResidual : 2 - 14	[1, 160, 7, 7]	--
Sequential: 3 - 16	[1, 160, 7, 7]	429, 224
InvertedResidual : 2 - 15	[1, 160, 7, 7]	--
Sequential: 3 - 17	[1, 160, 7, 7]	797, 360
InvertedResidual : 2 - 16	[1, 160, 7, 7]	--
Sequential: 3 - 18	[1, 160, 7, 7]	797, 360
Conv2dNormActivation: 2 -17	[1, 160, 7, 7]	--
Conv2d: 3 - 19	[1, 160, 7, 7]	153, 600
BatchNorm2d: 3 - 20	[1, 160, 7, 7]	1, 920
Hardswish: 3 - 21	[1, 160, 7, 7]	--
AdaptiveAvgPool2d: 1 - 2	[1, 160, 1, 1]	--
AttentionRefinement: 1 - 3	[1, 1, 960]	--
MultiheadAttention: 2 - 18	[1, 1, 960]	3, 690, 240
LayerNorm: 2 - 19	[1, 1, 960]	1, 920
Sequential: 2 - 20	[1, 1, 960]	--
Linear: 3 - 22	[1, 1, 960]	922, 560
ReLU: 3 - 23	[1, 1, 960]	--
Dropout: 3 - 24	[1, 1, 960]	--
Linear: 3 - 25	[1, 1, 960]	922, 560
LayerNorm: 2 - 21	[1, 1, 960]	1, 920
Total Params: 8, 511, 152 **Trainable Params**: 8, 511, 152 Non - trainable Params: 0 Total mult-adds (M): 215, 96		
Input size (MB): 0.60 Forward/backward pass size (MB): 70.47 Params size (MB): 19.28 Estimated Total Size (MB): 90.35		

For classification, LightGBM was selected owing to its efficiency and robust performance in gradient-boosting scenarios. The LightGBM model was fine-tuned using GridSearchCV, optimizing key hyperparameters, such as a learning rate of 0.001, maximum tree depth of 7, and subsample ratio of 0.8 in 100 epochs. An early stopping criterion was applied after ten rounds of no improvement in the validation set to prevent overfitting. This configuration effectively integrates MobileNetV3's feature extraction capabilities with MHSA and LightGBM's classification functionality, culminating in a robust sign language recognition system. TABLE 3 summarizes the hyperparameters used in the proposed framework, including those of the MobileNetV3, MHSA, and LightGBM.

TABLE 3
Hyperparameters for MobileNetV3, MHSA, and LightGBM

Components	Hyperparameter	Value
MobileNetV3	Input Resolution	224 x 224 pixels
	Learning Rate	0.001
	Batch Size	32
	Optimizer	Adam
MHSA	Number of Attention Heads	4
	Number of layers	8
	Dropout Rate	0.2
LightGBM	Learning Rate	0.001
	Maximum Depth	7
	Sabample Ratio	0.8
	Early Stopping Rounds	10
	Number of epochs	100

C. EVALUATION METRICS

We evaluated our model using the accuracy, precision, recall, and F1-score. Additionally, we computed 95% confidence intervals for each metric in five randomized trials (Eq. (13) to Eq. (16)).

$$\text{Accurcy} = \frac{TP+TN}{TP+TN+FP+FN} \quad (13)$$

$$\text{Precision} = \frac{TP}{TP+FP} \quad (14)$$

$$\text{Recall} = \frac{TP}{TP+FN} \quad (15)$$

$$\text{F1} - \text{score} = \frac{2 \cdot TP}{2 \cdot TP+FP+FN} \quad (16)$$

where TP represents true positives, TN stands for true negatives, FP indicates false positives, and FN indicates false negatives.

IV. RESULTS

The performance of the proposed hybrid sign language recognition model was evaluated using the ISL dataset. First, we conducted experiments to identify the most suitable deep learning model for feature extraction within our sign language recognition system. The evaluated models included VGG16, VGG19, ResNet50, InceptionNet, EfficientNet-B0, and MobileNetV3, and the results are presented in TABLE 4.

TABLE 4 compares the performance of various DL models for feature extraction on the ISL dataset. Among the tested models, MobileNetV3 exhibited the highest testing accuracy of 96.11%, with a training accuracy of 96.50%. This high accuracy, along with a precision of 96.00%, recall of 96.20%, and F1-score of 96.10%, suggests that MobileNetV3 effectively captured relevant features with minimal overfitting. These metrics highlight MobileNetV3's strength in extracting the discriminative features necessary for accurate sign language recognition. Its superior performance compared to other models, such as EfficientNet-B0 (95.56% testing accuracy) and ResNet50 (94.15% testing accuracy), validates its selection as the most suitable feature extractor for further experimentation.

TABLE 4
Performance of Various Deep Learning Models for Feature Extraction on the ISL Dataset

Model	Accuracy		Precision	Recall	F1-score
	Train	Test			
VGG16	94.11 ±1.2	93.23 ±1.1	92.10	93.0	92.55
VGG19	94.45 ±1.3	93.67 ±1.2	93.20	93.50	93.35
ResNet50	95.22 ±1.0	94.15 ±0.8	94.10	94.30	94.20
InceptionNet	95.10 ±1.1	94.12 ±1.0	93.84	94.01	93.92
EfficientNet-B0	96.11 ±0.9	95.56 ±0.8	95.53	95.71	95.61
MobileNetV3	96.53 ±0.7	96.11 ±0.6	96.00	96.21	96.11

After selecting MobileNetV3 as the optimal extractor, an experiment was conducted to determine the most effective machine learning classifier for sign language recognition using the extracted features. The classifiers tested included a Gradient Boosting Machine (GBM), XGBoost (XGB), LightGBM (LGBM), CatBoost (CAT), Support Vector Machine (SVM), and Random Forest (RF). The performance of each classifier was evaluated without parameter optimization to identify the classifier that maximized the recognition performance when paired with MobileNetV3.

TABLE 5
Performance of Various Classifiers on Features Extracted by MobileNetV3

Model	Accuracy		Precision	Recall	F1-score
	Training	Testing			
GBM	97.21 ±0.7	96.84 ±0.6	96.51	96.74	96.62
CAT	97.54 ±0.6	97.12 ±0.5	97.00	97.13	97.06
LGBM	**98.12 ±0.4**	**97.81 ±0.4**	**97.71**	**97.83**	**97.77**
XGB	97.69 ±0.7	97.45 ±0.5	97.30	97.40	97.35
RF	96.95 ±0.6	96.78 ±0.6	96.66	96.84	96.75
SVM	96.79 ±0.7	96.66 ±0.7	96.44	96.58	96.51

TABLE 5 presents the performance of various classifiers applied to the features extracted by MobileNetV3. LightGBM achieved the highest testing accuracy of 97.81% with a

training accuracy of 98.12%. Additionally, the LightGBM's precision of 97.71%, recall of 97.83%, and F1-score of 97.77% demonstrated its ability to provide a balanced and reliable classification. These results indicate that LightGBM is particularly effective in leveraging the features provided by MobileNetV3, surpassing other classifiers, such as XGBoost (97.45% testing accuracy) and CatBoost (97.12% testing accuracy). The balanced performance across all metrics, particularly the precision and recall values, confirms LightGBM's capability to minimize both false positives and false negatives, making it the most effective classifier for SLR.

Following the selection of MobileNetV3 as the feature extractor and LightGBM as the classifier, an additional experiment was conducted where Multi-head Self-Attention (MHSA) was integrated to enhance the extracted features. By incorporating MHSA, the model captured more complex dependencies within the data, thereby improving the overall performance when combined with the LightGBM. The results of the enhanced setup are listed in TABLE 6.

TABLE 6
Performance of MobileNetV3 + MHSA with LightGBM Classifier.

Model	Accuracy		Precision	Recall	F1-score
	Training	Testing			
Mobile NetV3 + LGBM	98.12 ±0.4	97.81 ±0.4	97.71	97.83	97.77
Mobile NetV3 + MHSA + LGBM	**99.54 ±0.3**	**98.42 ±0.4**	**98.19**	**98.81**	**98.15**

V. DISCUSSION

The classification reports of proposed model are presented in TABLE 7. Based on the classification report, the model performs exceptionally well in recognizing Indian Sign Language (ISL) gestures, achieving an overall macro-average precision, recall, and F1-score of 0.98. The high performance across all metrics suggests that the model is highly accurate, precise, and robust in classifying 35 sign language gestures. The precision and recall values remained consistently high across all classes, indicating that the model produced very few false positives and false negatives. Some classes, such as Class 6 (0.94 precision, 0.92 recall, 0.93 F1-score) and Class 14 (0.88 precision, 0.90 recall, 0.89 F1-score), showed relatively lower scores than others. This indicated a slightly higher misclassification rate for these classes. Several classes achieved 100% precision and recall, meaning that the model never misclassified them. TABLE 8 presents a confusion matrix for the proposed hybrid model. The majority of the classes exhibited strong diagonal dominance, but common errors arose in gestures with very similar hand configurations or under poor lighting conditions. The proposed sign language recognition framework, integrating MobileNetV3, MHSA, and LightGBM, demonstrated state-of-the-art performance compared to existing methods. As shown in TABLE 9, our proposed model achieved an impressive accuracy of 99.54 %, surpassing methods such as HOG + CABM-based CNN [26] with 99.22 %, and an Attention-based Hybrid CNN [27] with 97.67 %. This high accuracy highlights the effectiveness of the model in minimizing the classification errors across diverse sign gestures. Additionally, the proposed method achieves a precision of 98.19 %, which is slightly lower than SE-YOLOv5x [8] with 98.9 %, and maintains a strong balance between precision and recall, ensuring reliable classification. Notably, a recall of 98.81 % outperforms models such as SE-YOLOv5x (96.5 %) and Hybrid CNN-LSTM [14] with 87.4 %), indicating that the proposed method effectively reduces false negatives and accurately identifies relevant sign gestures. Furthermore, the F1-score of 98.15 % surpasses that of Hybrid CNN-LSTM by 84.4 % and is comparable to the Attention-based Hybrid CNN by 97.42 %, demonstrating robustness in handling imbalanced datasets.

TABLE 7
Classification reports of proposed models

class	precision	recall	f1	support
0	1.00	0.96	0.98	227
1	0.98	1.00	0.99	244
2	1.00	0.95	0.97	241
3	0.97	0.98	0.97	239
4	1.00	1.00	1.00	233
5	1.00	1.00	1.00	230
6	0.94	0.92	0.93	275
7	0.92	0.96	0.94	237
8	1.00	1.00	1.00	239
9	1.00	1.00	1.00	236
10	1.00	1.00	1.00	267
11	0.99	0.97	0.98	252
12	0.90	1.00	0.95	235
13	1.00	1.00	1.00	240
14	0.88	0.90	0.89	220
15	1.00	1.00	1.00	265
16	1.00	1.00	1.00	267
17	0.99	1.00	0.99	221
18	0.98	0.99	0.98	256
19	1.00	1.00	1.00	233
20	1.00	1.00	1.00	244
21	1.00	1.00	1.00	238
22	0.92	0.88	0.90	256
23	1.00	1.00	1.00	222
24	1.00	1.00	1.00	242
25	1.00	1.00	1.00	256
26	0.95	0.92	0.93	253
27	1.00	1.00	1.00	229
28	1.00	1.00	1.00	223
29	1.00	1.00	1.00	229
30	1.00	1.00	1.00	213
31	0.99	0.99	0.99	227
32	1.00	1.00	1.00	222
33	1.00	1.00	1.00	254
34	1.00	1.00	1.00	235
				8400
macro average	0.98	0.98	0.98	
weighted average	0.98	0.98	0.98	

Overall, the results confirm that the proposed MobileNetV3, Multi-Head Self-Attention, and LightGBM frameworks effectively capture both spatial and temporal dependencies in sign language recognition, leading to improved feature extraction and classification. While SE-YOLOv5x achieves a slightly higher precision, the proposed

method offers a more balanced performance across all metrics, making it highly suitable for real-world applications where both precision and recall are crucial.

TABLE 8
Confusion matrix of proposed models

	0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**	**17**	**18**	**19**	**20**	**21**	**22**	**23**	**24**	**25**	**26**	**27**	**28**	**29**	**30**	**31**	**32**	**33**	**34**
0	**227**	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	1	**244**	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
2	0	0	**241**	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
3	0	1	0	**239**	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4	0	0	0	0	**233**	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	**230**	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6	2	0	0	0	2	0	**275**	0	0	0	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
7	0	0	0	0	0	1	0	**237**	0	0	0	0	0	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
8	0	0	0	0	0	0	0	0	**239**	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	0	**236**	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0	0	0	**267**	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
11	0	0	0	0	0	1	0	0	0	0	0	**252**	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1
12	0	0	0	0	0	0	0	0	1	0	0	0	**235**	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
13	0	0	0	0	0	0	0	0	0	0	0	0	0	**240**	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
14	0	1	0	0	0	2	0	0	0	0	0	0	0	1	**220**	0	2	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	2
15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	**265**	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	**267**	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
17	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	**221**	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
18	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	**256**	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
19	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	**233**	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
20	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	**244**	0	0	0	0	0	0	0	0	0	0	0	0	0	0
21	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	**238**	0	0	0	0	0	0	0	0	0	0	0	0	0
22	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3	0	0	0	**256**	0	0	3	0	0	0	0	0	0	1	0	0
23	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	**222**	0	0	0	0	0	0	0	0	0	0	0
24	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	**242**	0	0	0	0	0	0	0	0	0	0
25	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	**256**	0	0	0	0	0	0	0	0	0
26	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	0	0	0	0	0	0	0	0	0	**253**	0	0	0	0	0	0	0	0
27	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	**229**	0	0	0	0	0	0	0
28	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	**223**	0	0	0	0	0	0
29	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	**229**	0	0	0	0	0
30	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	**213**	0	0	0	0
31	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	**227**	0	1	0
32	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	**222**	0	0
33	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	**254**	0
34	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	**235**

TABLE 9
Performance of MobileNetV3 + MHSA with LightGBM Classifier.

Model	Accuracy	Precision	Recall	F1-score
SE-YOLOv5x [8]	-	**98.9**	96.5	-
Hybrid CNN-LSTM [14]	84.65	86.8	87.4	84.4
HOG + CABM based CNN [26]	99.22	-	-	-
Attention based Hybrid CNN [27]	97.67	97.47	97.35	97.42
Proposed Method	**99.54**	98.19	**98.81**	**98.15**

Our results imply that near-real-time performance is achievable on a consumer GPU, suggesting the potential for on-device deployment on mobile or embedded hardware. From the perspective of deaf communities, real-world usability depends on both accuracy and responsiveness. Achieving 98.42% accuracy with minimal latency is a promising step toward practical applications, such as live sign language translation in public services, educational settings, and healthcare interactions.

Despite its overall strong performance, our dataset does not cover all regional ISL dialects or extensive demographic variations. Additionally, certain environmental conditions, extreme lighting changes, and cluttered backgrounds were underrepresented, which may account for some errors. A more diverse dataset could further bolster generalizability.

Although we have shown strong accuracy, the exploration of 3D or temporal attention models can capture dynamic gestures more effectively. Cross-lingual adaptations are another natural extension, as deaf communities worldwide use distinct sign languages. Incorporating body posture or facial cues as well as real-world longitudinal testing would also provide deeper insights into the model's stability and robustness over time.

VI. CONCLUSION

This paper presented a hybrid SLR framework integrating MobileNetV3, Multi-Head Self-Attention (MHSA), and LightGBM. Experimental evaluations on the ISL dataset confirmed MobileNetV3's effectiveness as a feature extractor, achieving a testing accuracy of 96.11%. LightGBM emerged as the best classifier, achieving 97.81% accuracy when used directly on MobileNetV3 features. The incorporation of MHSA further improved the performance, reaching a testing accuracy of 98.42%. In addition to the performance metrics, we performed statistical significance tests and provided confidence intervals, confirming the reliability of our results. Regarding practical implications, the near real-time performance of our approach is suitable for on-device deployment, addressing real-world constraints often encountered by deaf communities. Despite these advancements, these limitations persist. Our dataset, although sizable, does not cover all demographic or environmental variations. We also observed certain misclassifications under the extreme lighting conditions. Moreover, although our approach performed consistently over multiple runs, long-term or longitudinal studies would further validate performance stability under varying conditions. Future work will explore transformer-based 3D CNNs for enhanced spatiotemporal modeling, extend recognition to multiple sign languages, and integrate facial expressions, body poses, and depth sensors to reduce gesture ambiguity. Additionally, longitudinal and on-device testing will assess the model stability across time, environments, and hardware platforms, ensuring real-world applicability.

CONFLICTS OF INTEREST

The authors and co-authors declare that they have no conflicts of interest.

FUNDING INFORMATION

This study received no funding for this research article.

REFERENCES

[1] M. Mahyoub, F. Natalia, S. Sudirman, and J. Mustafina, "Sign Language Recognition using Deep Learning," *2021 14th International Conference on Developments in eSystems Engineering (DeSE)*, pp. 184–189, Jan. 2023, doi: 10.1109/dese58274.2023.10100055.

[2] Koller, O. (2020). Quantitative Survey of the State of the Art in Sign Language Recognition. *arXiv (CornellUniversity)*. https://doi.org/10.48550/arxiv.2008.09918

[3] Padden, C., & Humphries, T. (2009). *Inside Deaf Culture*. https://doi.org/10.2307/j.ctvjz83v3

[4] Stokoe, W. C. (2004). Sign Language Structure: An Outline of the Visual Communication Systems of the American Deaf. *The Journal of Deaf Studies and Deaf Education*, *10*(1), 3–37. https://doi.org/10.1093/deafed/eni001

[5] Cooper, H., Holt, B., & Bowden, R. (2011). Sign Language Recognition. In *Springer eBooks* (pp. 539–562). https://doi.org/10.1007/978-0-85729-997-0_27

[6] Cui, R., Liu, H., & Zhang, C. (2019). A Deep Neural Framework for Continuous Sign Language Recognition by Iterative Training. *IEEE Transactions on Multimedia*, *21*(7), 1880–1891. https://doi.org/10.1109/tmm.2018.2889563

[7] Bragg, D., Koller, O., Bellard, M., Berke, L., Boudreault, P., Braffort, A., Caselli, N., Huenerfauth, M., Kacorri, H., Verhoef, T., Vogler, C., & Morris, M. R. (2019). Sign Language Recognition, Generation, and Translation. In *Proceedings of the 21st International ACM SIGACCESS Conference on Computers and Accessibility* (pp. 16–31). https://doi.org/10.1145/3308561.3353774

[8] Attia, N. F., Ahmed, M. T. F. S., & Alshewimy, M. A. (2023). Efficient deep learning models based on tension techniques for sign language recognition. *Intelligent Systems With Applications*, *20*, 200284. https://doi.org/10.1016/j.iswa.2023.200284

[9] Kumar, C. M. N., Vanitha, A., Lavanya, N. Y., Lekhana, N. C., Tasmiya, R., & Nisarga, L. D. (2024). Deep learning-based recognition of sign language. *Second International Conference on Data Science and Information System*, 1–6. https://doi.org/10.1109/icdsis61070.2024.10594011

[10] Kothadiya, D., Bhatt, C., Sapariya, K., Patel, K., Gil-González, A., & Corchado, J. M. (2022). Deepsign: Sign Language Detection and Recognition Using Deep Learning. *Electronics*, *11*(11), 1780. https://doi.org/10.3390/electronics11111780

[11] Ashrafi, A., Mokhnachev, V. S., & Harlamenkov, A. E. (2024). Improving Sign Language Recognition with Machine Learning and Artificial Intelligence. *2022 4th International Youth Conference on*

Radio Electronics, Electrical and Power Engineering (REEPE), 1–6. https://doi.org/10.1109/reepe60449.2024.10479844

[12] Rajasekhar, N., Yadav, M. G., Vedantam, C., Pellakuru, K., & Navapete, C. (2023). Sign Language Recognition using Machine Learning Algorithm. In *International Conference on Sustainable Computing and Smart Systems (ICSCSS)* (Vol. 9, pp. 303–306). https://doi.org/10.1109/icscss57650.2023.10169820

[13] Ranjbar, H., & Taheri, A. (2024). Continuous Sign Language Recognition Using Intra-inter Gloss Attention. *arXiv (Cornell University)*. https://doi.org/10.48550/arxiv.2406.18333

[14] Kumari, D., & Anand, R. S. (2024). Isolated Video-Based Sign Language Recognition Using a Hybrid CNN-LSTM Framework Based on Attention Mechanism. *Electronics*, *13*(7), 1229. https://doi.org/10.3390/electronics13071229

[15] Sarhan, N., Wilms, C., Closius, V., Brefeld, U., & Frintrop, S. (2023). Hands in Focus: Sign Language Recognition Via Top-Down Attention. *2022 IEEE International Conference on Image Processing (ICIP)*, 2555–2559. https://doi.org/10.1109/icip49359.2023.10222729

[16] Ma, Y., Xu, T., & Kim, K. (2022). A Digital Sign Language Recognition based on a 3D-CNN System with an Attention Mechanism. *2022 IEEE International Conference on Consumer Electronics-Asia (ICCE-Asia)*, 1–4. https://doi.org/10.1109/icce-asia57006.2022.9954810

[17] Howard, A., Sandler, M., Chen, B., Wang, W., Chen, L., Tan, M., Chu, G., Vasudevan, V., Zhu, Y., Pang, R., Adam, H., & Le, Q. (2019). Searching for MobileNetV3. *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, 1314–1324. https://doi.org/10.1109/iccv.2019.00140

[18] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., & Polosukhin, I. (2017). Attention is All you Need. *arXiv (Cornell University)*, *30*, 5998–6008. https://arxiv.org/pdf/1706.03762v5

[19] Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., Ye, Q., & Liu, T. (2017). LightGBM: A Highly Efficient Gradient Boosting Decision Tree. In *31st International Conference on Neural Information Processing Systems*. https://hal.science/hal-03953007

[20] LeCun, Y. A., Bottou, L., Orr, G. B., & Müller, K. (2012). Efficient BackProp. In *Lecture notes in computer science* (pp. 9–48). https://doi.org/10.1007/978-3-642-35289-8_3

[21] Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2017). ImageNet classification with deep convolutional neural networks. *Communications of the ACM*, *60*(6), 84–90. https://doi.org/10.1145/3065386

[22] Chen, T., & Guestrin, C. (2016). XGBoost: A Scalable Tree Boosting System. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 785–794). https://doi.org/10.1145/2939672.2939785

[23] Hu, J., Shen, L., & Sun, G. (2018). Squeeze-and-Excitation Networks. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*. https://doi.org/10.1109/cvpr.2018.00745

[24] Kumar, H., Dwivedi, A., Mishra, A. K., Shukla, A. K., Sharma, B. K., Agarwal, R., & Kumar, S. (2024). Transformer-based decoder of melanoma classification using hand-crafted texture feature fusion and Gray Wolf Optimization algorithm. *MethodsX*, *13*, 102839. https://doi.org/10.1016/j.mex.2024.102839

[25] Indian *Sign Language (ISL)*. (2021, June 4). Kaggle. https://www.kaggle.com/datasets/prathumarikeri/indian-sign-language-isl

[26] D. Kumari and R. S. Anand, "Fusion of Attention-Based Convolution Neural Network and HOG features for static sign language recognition," *Applied Sciences*, vol. 13, no. 21, p. 11993, Nov. 2023, doi: 10.3390/app132111993.

[27] S. Biswas, R. Saw, A. Nandy, and A. K. Naskar, "Attention-enabled hybrid convolutional neural network for enhancing human–robot collaboration through hand gesture recognition," *Computers & Electrical Engineering*, vol. 123, p. 110020, Dec. 2024, doi: 10.1016/j.compeleceng.2024.110020.

AUTHOR'S BIOGRAPHY

Hemant Kumar is an Assistant Professor in the Department of Information Technology, School of Engineering and Technology (UIET), Chhatrapati Shahu Ji Maharaj University Kanpur, with over 10 years of academic and research experience. His research interests include Artificial Intelligence, Machine Learning, Image Processing, and Data Science. He has authored numerous research papers in reputed journals, international conferences, and book chapters. He holds an M.Tech. from Devi Ahilya Vishwavidyalaya and an MCA from Uttar Pradesh Technical University. He is currently pursuing his Ph.D. from the Harcourt Butler Technical University, Kanpur. Beyond his academic engagement, he is a lifetime member of the Association for Computing Machinery (ACM) and actively contributes to the research community as a reviewer for SCI-indexed journals. His dedication to advancing knowledge in his field earned him recognition among peers and researchers.

Rishabh Sachan is an accomplished professional in the fields of Artificial Intelligence and Data Science. He holds an M.Tech from the prestigious Indian Institute of Technology (IIT) Jodhpur, where he specialized in the Department of School of Artificial Intelligence with a focus on Data Science. With nearly four years of experience working in multinational corporations, Rishabh has collaborated with renowned organizations such as Cognizant, partnering with clients like TCS, Samsung Research, and the World Bank Group. His diverse professional background has endowed him with a robust skill set and a profound understanding of industry practices. Currently, Rishabh serves as an Assistant Professor at KIET, Ghaziabad, where he is dedicated to imparting his knowledge and expertise to the next generation of professionals in Artificial Intelligence and Data Science. Through his academic contributions, he continues to shape the future of these dynamic fields**.**

Dr. Mamta Tiwari is an Assistant Professor at the School of Engineering and Technology, Chhatrapati Shahu Ji Maharaj University, Kanpur, with over 22 years of experience in teaching and research. She holds an MCA, M.Tech., and Ph.D., with expertise in programming languages such as C and Python, and core subjects including Operating Systems, Databases, Data Mining, and Data Science. She has published 10 research papers in reputed international and national

journals, Scopus, and UGC Care-listed journals, along with a book chapter. Actively involved in academic and professional development, she has served as a member of organizing committees in international and national conferences, attended 21 FDPs and webinars, and has mentored over 250 BCA and MCA students' projects. Her commitment to research, technical education, and student mentorship makes her a distinguished academic in her field.

Amit Kumar Katiyar is an Assistant Professor at the Department of Electronics and Communication Engineering, School of Engineering and Technology (UIET), Chhatrapati Shahu Ji Maharaj University Kanpur, with over 15 years of experience in academia and research. He is currently pursuing his Ph.D. in Electronics and Communication Engineering. He obtained his M.Tech. in Electronics and Communication Engineering in 2011 from Harcourt Butler Technical University (HBTI), Kanpur. His research interests include optical networks, digital filters, and communication systems. He holds seven patents in various areas of electronic and communication engineering, and has published numerous research papers in journals and conferences. His extensive experience and significant contributions to both research and teaching have established him as a prominent figure in his field.

Namita Awasthi is an Assistant Professor in the Department of Computer Science and Engineering at Allenhouse Institute of Technology, Kanpur, with over eight years of academic and research experience. She has made significant contributions to Artificial Intelligence, Machine Learning, and Image Processing, authoring and co-authoring numerous research papers published in Scopus-indexed and international journals. She holds an M.Tech. from Kanpur Institute of Technology, affiliated with Dr. A.P.J. Abdul Kalam Technical University, and an MCA from Amity School of Computer Sciences, Noida, affiliated with Uttar Pradesh Technical University. Her research focuses on developing advanced computational techniques to enhance AI applications, bridging theoretical advancements with real-world solutions. Through her relentless efforts, she continues to inspire future technologists, foster innovation, and shape the evolution of AI-driven solutions in the field of computer science.

Dr. Pushpa Mamoria has over 20 years of academic and administrative experience and is currently an Associate Professor in the Department of Computer Applications, School of Engineering and Technology, Chhatrapati Shahu Ji Maharaj University Kanpur. She was previously associated with IIIT Allahabad between 2005 and 2007. She has held various key roles, including Head of the Department, Proctorial Board Member, Convener of the Alumni Association, Hostel Warden, and Program Officer for NSS. Her research interests include Digital Image Processing, Artificial Intelligence (AI), Machine Learning (ML), Fuzzy Logic, Soft Computing, IoT, Big Data, Data Science, Indian Knowledge System, and Social Studies. She has published numerous research papers in renowned international and national journals, conferences, and symposia. Dr. Mamoria earned her Ph.D. in Computer Science from Babasaheb Bhimrao Ambedkar University, Lucknow (2018), an M.Tech. from Devi Ahilya Vishwavidyalaya, Indore, and a B.E. in Computer Science and Engineering from Shri G. S. Institute of Technology and Science, Indore (SGSITS). She supervised several M. Tech., MCA, and B. Tech. theses, and are currently guiding MCA, BCA, and Ph.D. students.

Mr. Ramnayan Mishra is an Assistant Professor in the Department of Information Technology, School of Engineering and Technology (UIET), CSJM University, Kanpur, with over 10 years of academic and research experience. He is a dedicated academic and researcher in the field of Computer Science and Engineering with expertise in Blockchain, Machine Learning, and Image Processing. He holds an M.Tech. in Information Technology and B.Tech. in Computer Science and Engineering. He is currently pursuing his Ph.D. from the Chhatrapati Shahu Ji Maharaj University, Kanpur, where he also serves as a faculty member. He has authored several research papers in reputed international and national journals, and conferences. Active engagement in academic mentorship and research contributes to innovative developments in computing. His dedication to scholarly research, technical excellence, and student guidance made him a distinguished figure in his field.