International Journal of Engineering Trends and Technology
ISSN: 2231-5381/ https://doi.org/10.14445/22315381/IJETT-V74I4P114

Volume 74 Issue 4, 180-195, April 2026

Original Article

A Hybrid Deep Learning Framework for Indian Sign Language Recognition from Images and Videos

S. Jayalakshmi[1], S.P. Balamurugan[2*]

[1,2]*Department of Computer and Information Science, Annamalai University, Tamil Nadu, India.*

[*]*Corresponding Author : spbcdm@gmail.com*

Received: 23 June 2025 Revised: 11 February 2026 Accepted: 28 February 2026 Published: 29 April 2026

Abstract *- This paper proposes a comprehensive deep learning approach for Indian Sign Language (ISL) recognition based on static image and dynamic video modalities. Based on a common use of spatial and temporal complexities in sign language gestures, this paper proposes two types of specialist hybrids. For static gestures, a CNN-Transformer model is introduced, which effectively combines multiple types of convolutional networks with Transformer models to capture the image features depending on global contextual relationships. A CNN-LSTM model based on the ConvLSTM2D layer is then used to jointly learn both spatial and temporal relations between sequences of video frames, responsible for generating dynamic sign sequences. Full-fledged repository containing the data, which is a balanced set of 36 ISL signs for each modality, along with uniform preprocessing steps like grayscale normalization, Gaussian smoothing, resizing, and sequence padding to maintain homogeneity in inputs. To guarantee computational efficiency while preserving gesture clarity, frames are temporally sampled in the video pipeline. Dynamic class weighting is applied during training to balance the classes, while early stopping and learning rate schedulers ensure that convergence is optimal without leading to overfitting. The models are tested and trained, and five-fold cross-validation is used for statistical strength and generalization. These metrics are calculated, and furthermore, the model's performance is plotted using confusion matrices, ROC curves, and accuracy-loss curves. The CNN-Transformer model has excellent image-based classification performance, and the CNN-LSTM model has very good motion-based temporal features capturing capability. This dual-modality system shows a successful merger of spatial and spatiotemporal deep neural network architectures for real-time, accurate ISL recognition. This system can easily be incorporated into assistive communication tools (such as sign-to-speech translators), educational resources, and mobile applications, thus facilitating inclusion for the hearing and speech-impaired communities.*

Keywords *- Indian Sign Language recognition, CNN-Transformer model, CNN-LSTM Model, Static Image, Dynamic Video, Spatial and Temporal Features, Five-Fold Cross-Validation.*

1. Introduction

In every deaf and hard-of-hearing community worldwide, sign language is the first and most important means of communication. The Sign Language of India is very important for communication by the hearing-impaired in India. Note that despite the highest priority of getting things right, accessibility requirements in general have been scarcely met as trained interpreters are not readily available, apart from being inconveniently expensive to reach assistive technologies that only become more common with every decade, thus leading to an effective let-out for preventing free-flowing interaction between both the hearing impaired and mainstream populace. There is an increasing macro trend in a wider migration, such as forced or voluntary by political reasons, civil violence, and an increase in refugees, which affects accessibility for many types of persons with various needs who cannot engage and participate in social processes. Sign language encompasses varying hand and finger movements, facial expressions, and dynamics of motion in time; therefore, automatically recognizing this modality of language is a challenging task. Additionally, variations in lighting conditions, background, signing style, and speed of execution present further challenges for recognition. As such, the preliminary research works are predominantly dependent on machine learning methodologies that apply engineered characteristics like Histogram of Oriented Gradients (HOG), Local Binary Patterns, and optical flow descriptors. The approaches based on these have not been able to achieve significant success; they rely heavily on human-engineered features that restrict generalization across different conditions used during signing [1].

Moreover, the discrete convolution and pooling operations allow CNN architectures to extract features hierarchically and gradually learn objects in the images at different semantic scales. Models based on CNNs have

yielded state-of-the-art results for recognizing static gestures over ISL data [1, 2]. However, since these CNN-based models only take spatial representations into account, they do not solve for temporal factors when working on continuous sign language videos or gesture recognition models [1]. Accordingly, a new effective hybrid model is introduced, which combines CNN and sequence deep learning algorithms (Recurrent Neural Networks (RNN) and long short-term memory (LSTM)) to identify dynamic hand gestures on top of ISL data [2, 3]. Although these CNN-LSTM models are good at recognizing the corresponding dynamic hand gestures, it is still a bit difficult for them to deal with long-distance contextual dependencies, and on the other hand, global feature relationship coordination. However, brainy approaches based on Transformer architectures, which lean on self-attention strategies, have proven to be favorable for managing long-distance contextual relations without relying on any recurrent makeup. The use of transformer architecture for sign language recognition has proven fruitful results in managing complex spatial dependencies, as its main objective is to attend to all features equally well [4]. Nevertheless, existing solutions based on the Transformer architecture are mainly regarded as being independent or continuous hybrid approaches without suitable handling of both static and dynamic sign recognition requirements independently or collectively. Furthermore, most of the research work is focused on static and dynamic video recognition separately, without any global solution to both problems in an efficient way.

1.1. Research Gap and Motivation

A review of the current literature reveals a lack of a dual modality, task-specific deep learning framework capable of handling both static sign images and dynamic sign videos through architectures tailored to the characteristics of each modality. Existing approaches have either relied on CNN-based methods, which suffer from not being able to cope with sign videos due to limited temporal modeling, or they have employed general hybrid techniques without leveraging attention-based models for spatial context learning and temporal memory modeling within the realm of sign videos.

1.2. Proposed Contribution and Novelty

This paper puts forward a hybrid deep learning approach for ISL recognition in response to such shortcomings, which includes two models:

(i) a CNN-Transformer model (HDL-CTM) works for the static sign image recognition, and
(ii) a CNN-LSTM model (HDL-CLM) for Dynamic Sign Video Recognition

The local feature extraction capacity of CNNs and the global context learning power of the Transformer attention mechanism are not contradictory but complementary, as they provide a more accurate representation of intricate hand shapes and spatial relations in static visual contents [4]. The CNN–LSTM model, on the other hand, is a combination of convolutional layers and LSTM layers that can be used for extracting spatial features in addition to temporal modeling using sequences [3]. In contrast to these works, we propose an architecture that adopts a task-specific hybrid design where each modality is ensured to learn features in the best possible way instead of trying to fit one architecture for both. Extensive experiments on publicly available ISL image and video datasets validate the effectiveness of the proposed framework. This is achieved through commonly used evaluation metrics such as accuracy, precision, recall, F1-score, confusion matrices, and ROC curves that are based on the findings of [5]. Results with state-of-the-art models further confirm the efficacy of our proposed framework. Thus, this paper contributes a strong and efficient framework for recognition of ISL, which has evident application potential with respect to the assisted communication systems.

2. Related Works

Ahmed Mateen Buttar et al. [6] have developed a hybrid deep learning model with dynamic and static sign language gesture recognition. In this work, the authors present a new computational architecture comprising CNNs and LSTM networks to capture spatial-temporal interactions underlying sign language interactions. Through incorporating computational consideration regarding static hand poses and motion templates, it overcomes the limitations of single modality systems. Prachi Pramod Waghmare et al. To give access to an under-explored territory we are focused on in this paper, introduce a brand new dual-task framework that integrates the tasks to be solved: sign language recognition and video generation. The authors propose a deep learning-based model that recognizes ISL gestures as well, but in addition to recognition, also generates a sequence of videos that align with the recognized gesture, thus linking recognition with interpretability. Using CNNs to extract features from audio and using a generative approach to the video output implemented, it is conceivable that these methods could help create real-time communication tools for deaf people. Demonstrating promising results for recognition accuracy and visual quality in synthesized videos, with implications for applications such as education and assistive technology. Deep Spatiotemporal Learning System – Md Azher Uddin et al. This paper deals with the specific issue of sign language recognition in videos, named Sign Language Recognition (SLR) [2, 7]. The proposed system learns in this way and adopts CNNs and LSTMs to individually extract spatial features from sign language image frames and temporal features using the sequences of sign language videos. This is crucial in understanding the gesture patterns of a signer, as it helps the model understand time-dependent sign language. The results of an experimental analysis show a high degree of accuracy, which demonstrates the model's ability to adapt to changing sign language and other environmental dynamics. Jay Joshi et al. [4] demonstrate a contemporary deep learning framework, grounded in the Transformer architecture, that

could be beneficial to solve the sign language recognition task. The study investigates the ability of self-attention transformer models to learn informative spatial dependencies and temporal relations between sign language gestures, which are typically difficult to model with standard CNNs or RNNs. Class classification improves significantly by performing a classification task based on various types of sign language image classification data. A multimodal neural representation will contribute to continuous Sign Language Recognition (SLR) [8]. The SignVTCL model is built upon a visual-textual contrastive learning paradigm, in which signs can be aligned to text that represents what the person signed, allowing for more detailed semantic analysis. This approach achieves higher accuracy in real-time, continuous scenarios by leveraging spatial and temporal video features of signs along with the linguistic aspects.

Maher Jebali et al. proposed a highly efficient sign language recognition system based on both manual and non-manual elements in [9]. The authors further note that because sign languages have, in most of their presentations, been treated as conventionally based on both subtle non-manual elements and hand signs, this has led to the need for them to be incorporated in any final model. This deep learning scheme works great as it is able to combine all the modalities simultaneously with a multi-stream CNN model. It is done for the improved semantic comprehension as well as the precision of the ultimate identification system. Sunusi Bala Abdullahi et al.62595] propose a novel deep learning scheme for efficient recognition of 3D sign language signs in [10]. It has been shown that such deep models can be effective in modeling both spatial and temporal dynamics due to the transformation of the sequence into the frequency domain. Thus, the spectral method shows clearly in the reduction of more focus on sparse to observe the compact as tentative observations. Given how much experimentation had been done, it is no surprise that the model embodies a broad set of improvements. "Liqing Gao et al. in [11] present a new method based on this novel cross-modal knowledge distillation framework, which improves the efficiency of continuous sign language recognition." This technique involves semantics knowledge transfer from a pretrained high-capacity textual or multimodal teacher model and a low-capacity, visual-only student model. The latter network is able to learn better temporal patterns and context of a time series of signs, improving the recognition rate without giving up its computational efficiency. Sharvani Srivastava et al. in [12] proposed a deep learning-based system consisting of a light-weight and computation-efficient approach for recognizing continuous signs using the MediaPipe Holistic model. This process works effectively for the extraction of images, hand, body, and face coordinates from the frames, resulting in the model being able to learn accurate spatial-temporal differences corresponding with specific sign gestures. These features are used in a deep-learning model to develop a system that recognizes the continuous signs efficiently and accurately. Abdesselem Dakhli et al. However, [9] proposed a complete model based on deep learning that is able to combine sign language's manual and non-manual components seamlessly in the task of sign language recognition. The aforementioned research highlights how non-manuals aid in the description of intricacies within sign language and, as a response, suggests a multi-stream approach that is applied prior to integration and culminating in prediction between manualism and non-manual components.

Anudyuti Ghorai et al. [13] addressed a new method for ISL recognition based on the use of spatial transformation networks and network deconvolution. The approach is based on the premise that spatial invariance not only improves redundancy (of features) but also increases the recognition performance of hand gestures. To this end, the proposed model architecture enables the model to learn important functions deep inside for selective attention without loss of generalizability, with higher interpretive and efficient performance since it can be realized by network deconvolution. Zaid Saad Bilal et al. In contrast, [14] presents a new deep learning method that is novel in the field of Arabic Sign Language recognition. This research results from the growing need for communication aids for people with disabilities who speak Arabic. It presents an innovative approach incorporating CNNs and sophisticated preprocessing methods to extract efficient spatial characteristics from sign gestures, including methods for data augmentation, and optimizing for both improved model performance and efficiency. Edwin Shalom Soji et al. [15] acknowledge the importance of a robust machine learning model for predicting and correctly classifying human-performed ISL gestures.3 By integrating optimized approaches for feature extraction and selection, the existing machine learning models are improved to allow more sophisticated classification for different sign types. In addition, the proposed solution is efficient and can be integrated with a high degree of computational efficiency, allowing real-time usage and deployment. Hussain A et al. An overview of trends relevant to ISL gestures and signs manifestations shows a propensity for various technology convergences that were introduced in [16].

The article discusses some of the developments that have been performed using deep learning algorithms and their influence on ISL signs and sentences interpretation, particularly considering machine learning algorithm advancements. Moreover, at the same time that these methods ensure better classification results, it also introduces a need for including multimodal data like gestures, faces, and motion. Yanqiong Zhang et al. [17] provide an overview of the recent work done in applying deep learning techniques to sign language recognition. In this review, the authors systematically explore many types of deep models for static and dynamic gesture recognition, from simple, widely known models such as CNNs and RNNs to more complex ones like

LSTMs or even Transformers and/or hybrid ones. In particular, this depicts the importance of recent developments in spatiotemporal modeling, attention mechanisms, and multimodal fusion towards achieving class-leading accuracy and scalability for state-of-the-art sign recognition systems. Fatma M. Najib [18] suggests an adaptive sign language recognition system that is capable of integrating multiple sign languages in a single machine learning framework.

By focusing on cross-linguistic interpretation of gestures, this work fills a much-needed gap in sign language processing and thus makes the system adaptable to users with various linguistic backgrounds. This approach applies newer techniques of feature extraction and classification for different sign languages, such as American, British, and Arabic sign languages. Boháček and Hruš [19] proposed a Transformer-based framework for word-level sign language recognition based on pose representations instead of working directly with the raw images. They demonstrated that the model effectively exploits self-attention over sequences of sign poses to handle long-range dependencies and contextual relations, achieving enhanced recognition rates for isolated sign words. Kumari et al. [20] proposed a hybrid architecture that integrated CNN with Attention and LSTM, respectively, for isolated sign language recognition in video. The system performs better in the recognition of dynamic sign gestures as spatial features are extracted using a CNN, and temporal dependency is modeled by an attention-enhanced LSTM. Zhao et al. [21] introduced a self-supervised learning framework which enforces spatial–temporal consistency for sign language recognition. Achieving this decreases the dependency on large labeled datasets by learning robust representations from unlabeled data and a scalable model for generalization across multiple signing conditions. Chaudhari et al. [22] draw inspiration from biologically plausible neural computation and investigate spiking neural networks for sign language recognition.

This method yields energy-efficient gesture recognition with comparable accuracy, thus positioning neuromorphic models as a viable alternative to real-time and low-power sign language solutions. Other recent research has investigated multimodal and cross-lingual sign language recognition settings for improving semantic understanding and generalization. These approaches use visual information along with other modalities: hand poses, keypoints, facial expressions, or even body landmarks and text representations to model both manual and non-manual components of signing fully. Various studies have shown that incorporating pose or text embeddings with visual features through attention mechanisms and contrastive learning approaches yields better recognition performance. Meanwhile, cross-lingual and multilingual sign language recognition models are proposed to leverage knowledge transfer across different sign languages by learning a joint representation. In return, such methods break free of any modal specificity in performance (being language-agnostic) but incur higher computational and multimodal annotation complexity. In contrast, we propose a task-specific visual-only architecture for the recognition of Indian Sign Language while being extensible to multimodal and cross-lingual modalities in subsequent works.

3. The Proposed Model

This paper presents a hybrid deep learning framework for identifying Indian Sign Language (ISL), using both static images and video sequences. A Hybrid Deep Learning CNN-Transformer Model (HDL-CTM) is used for image classification to obtain local and global spatial features. To learn temporal features from the sequence of gestures for video-based recognition, a Hybrid Deep Learning CNN-LSTM Model (HDL-CLM) is used. The two models go through the same preprocessing techniques for being pre-learned alongside class balancing strategies. The block diagram of the approaches proposed is shown in Figure 1.

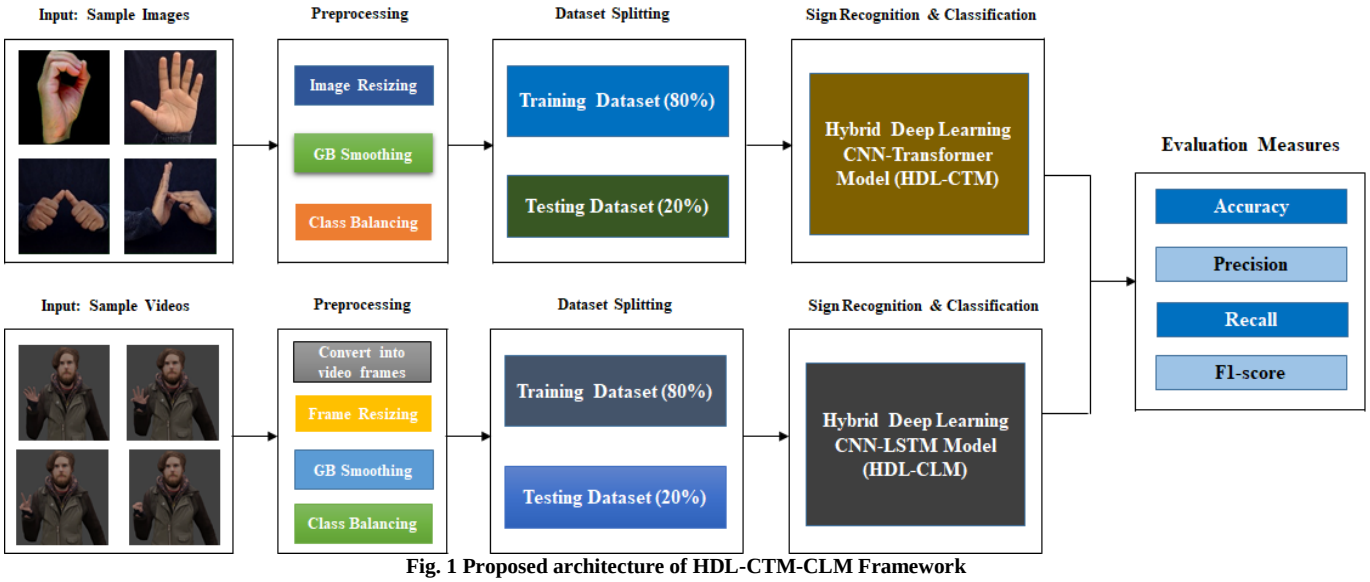


Fig. 1 Proposed architecture of HDL-CTM-CLM Framework

3.1. Image Preprocessing

To maintain the reliability, consistency, and robustness of the proposed deep learning models for Indian Sign Language classification on both image and video data, we follow a systematic preprocessing pipeline. These processes standardize input dimensions, amplify useful image characteristics, and limit superfluous noise that could degrade model performance. Sticker data-splitting, transformation key operations, grayscale transformation resizing, Gaussian blurring, normalization, and temporal frame extraction. The collective contributions of these preprocessing tasks ensure that the crude input data is well-conditioned for stable feature learning, which prevents the CNN-Transformer and CNN-LSTM models from chasing downliers or noise in gesture patterns and provides a pathway to high classification accuracy.

3.1.1. Grayscale Conversion

RGB images are converted to grayscale in order to reduce complexity and focus on the shape and texture.

Consider an RGB image $I(x, y)$ defined in terms of its red, green, and blue components:

$$I(x, y) = [R(x, y), G(x, y), B(x, y)] \quad (1)$$

The grayscale image $G(x, y)$ is computed as:

$$G(x, y) = 0.2989.R(x, y) + 0.5870.G(x, y) + 0.1140.B(x, y) \quad (2)$$

3.1.2. Image Resizing

The images are resized to a constant dimension, which helps in normalizing the input size of a deep learning model.

Given an input image $I \epsilon R^{H*W}$, the resized image becomes:

$$I_{resized} \epsilon R^{64*64} \quad (3)$$

Resizing helps us with the similar input size of all samples.

3.1.3. Gaussian Blurring

A Gaussian filter is applied to decrease high-frequency noise.

The Gaussian kernel $G(x, y)$ is defined as:

$$G(x, y) = \frac{1}{2\pi\sigma^2} . e^{-\frac{x^2+y^2}{2\sigma^2}} \quad (4)$$

The blurred image I_b is obtained by:

$$I_b(x, y) = \sum_{i=-k}^{k} \sum_{j=-k}^{k} I(x + i, y + j).G(i, j) \quad (5)$$

3.1.4. Normalization

Normalize the pixel values to [0,1] for faster convergence and more stable training:

Let the grayscale image pixel be $p \epsilon [0,255]$, then,

$$p_{norm} = \frac{p}{255} \quad (6)$$

This ensures that the input tensor is in a numerically stable range for neural networks.

3.1.5. Frame Extraction (For video)

A set number of frames per video is sampled to compose a temporal sequence.

Let V be a video containing N frames, the frame sampling interval is:

$$\Delta = \left\lfloor \frac{N}{T} \right\rfloor \quad (7)$$

Where T is the number of desired frames.

Selected frames:

$$F_t = V[t.\Delta] \quad for\ t = 0,1,2, \ldots . T - 1 \quad (8)$$

3.1.6. Sequence Padding

If a short video has fewer than enough sampled frames, we do not add to make it consistent:

Let $F = \{f_1, f_2, \ldots, f_k\}, with\ k < T$. Then pad with:

$$F' = \{f_1, f_2, \ldots, f_k, 0, \ldots, 0\},\ such\ that\ |F'| = T \quad (9)$$

These preprocessing operations maintain the data well prepared for HDL-CTM (image) and HDL-CLM (video) models, homogeneous in nature, with minimum noise, and easy feature extraction.

3.1.7. Class Balancing

A natural dataset is made up of classes, which can be imbalanced since some occur more than others. It causes an imbalance that skews a model's learning such that the model learns really well for the majority classes but struggles with minority ones. In response, the training is conducted with class weighting: minority classes are given a higher weight and majority ones a lower. This, in turn, motivates the model to assign higher importance to less prevalent classes while increasing fairness and overall classification accuracy.

Let the dataset contain N total samples distributed across C classes, with n_c samples in class c. The class weight, w_c for class c, is calculated using:

$$w_c = \frac{N}{C.n_c} \quad (10)$$

Where, w_c is the weight for class c, N is the total number of samples in the dataset, C is the total number of classes, and n_c is the number of samples in class c. These are then used as weights in the loss function at training time, making it a lot more penalizing when the model makes an error on minority class examples. In this way, all classes contribute equally to the learning process, no matter how prevalent they are in the training set.

3.2. Dataset Splitting

The dataset is systematically partitioned into training and test sets for evaluating the performance and generalization ability of the designed deep learning models. In this work, the data was split in an 80:20 ratio, where 80% of the total data was used to train the model and the other 20% is used for testing. This makes sure that the model sees the structures in most of the data, and on those examples that it has not seen, we can check how well it is predicting. Moreover, a video classification uses 5-fold cross-validation to achieve better robustness and lower bias in the evaluation. The data set is split into 5 different subsets in each fold, and the model trains and validates 5 times using 1 of the subsets as validation and the others to train. This method, however, provides a rough estimate of model accuracy and guards against overfitting, while producing comparable predictions with random data partitions.

3.3. Sign Image Recognition and Classification using HDL-CTM

The proposed approach for static sign image recognition is through the use of a hybrid deep learning model that takes advantage of Convolutional Neural Networks for spatial feature extraction and Transformer layers to contextualize these features. This helps the model to learn global as well as local visual patterns from the hand gesture images.

3.3.1. CNN-Based Feature Extraction

Convolutional layers have the same application to extract local spatial details such as edges, curves, and hand shape textures. Multiple convolutional and pooling layers are stacked in a way so that the hierarchy of features is learned, capturing higher-order moments.

$$X_l = f(W_l * X_{l-1} + b_l \quad (11)$$

Where f is typically a ReLU activation function, W_l and b_l are the weights and bias for layer l, and $*$ denotes the convolution operation. The model obtains a feature map that retains its most relevant spatial characteristics for the input gesture.

3.3.2. Flattening and Positional Encoding

In the 2D feature map generated by our CNN, we flatten it into a sequence of tokens to be processed by the Transformer. Transformers are position-invariant, so positional encodings are added to keep spatial order.

$$Z_0 = [x_1 + p_1, x_2 + p_2, \dots, x_n + p_n] \quad (12)$$

Where, x_i is the vectorized i^{th} patch, and p_i is its corresponding positional encoding. Now this converts the spatial information into a token sequence that is compatible with the Transformer's attention mechanism.

3.3.3. Transformer Encoder for Global Attention

The sequence of tokens is processed using Multi-Head Self-Attention and Feed-Forward Networks in the transformer encoder. This gives the model a holistic context, like how the individual parts of the hand are related to each other in an entire image.

Self-Attention:

$$Attention(Q.K,V) = softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)V \quad (13)$$

Multi-Head Attention:

$$MHSA(Z) = [head_1; \dots; head_h]W^o \quad (14)$$

Feed-Forward Network:

$$FFN(x) = ReLU(xW_1 + b_1)W_2 + b_2 \quad (15)$$

These operations assist the model in perceiving intricate, non-local dependencies within the gesture.

3.3.4. Classification Head

The output of Transformer encoding is then forwarded to a classification head comprising a fully connected layer with softmax activation that outputs class probabilities.

$$\hat{y} = softmax(W_{cls}Z_{out} + b_{cls} \quad (16)$$

Where, Z_{out} is the final encoded sequence, and $\hat{y}\ \epsilon\ R^C$ is the predicted probability for each class.

3.3.5. Loss Function

During training, the categorical cross-entropy loss is minimized between true labels and predicted probabilities.

$$L = -\sum_{i=1}^{C} y_i \log(\hat{y}_i) \quad (17)$$

Where, y_i is the true label in one-hot encoding, and $\hat{y}_i$ is the predicted probability for class i.

3.3.6. ADAM Optimization

The Adam (Adaptive Moment Estimation) optimizer is one of the popular stochastic optimizers used in deep learning, well known for its efficiency and a unique feature of adaptive learning rate. Adam combines the strengths of two other extensions: momentum and RMSProp. Adam keeps decaying averages of the gradients exponentially (first moment) and

squared gradients (second moment), which are then used to update each parameter's learning rate. Given a loss function $L(\theta)$ with parameters θ, the gradient at timestep t is $g_t = \nabla_\theta L(\theta_t)$. The optimizer updates the first moment. m_t and the second moment v_t as:

$$m_t = \beta_1 m_{t-1} + (1-\beta_1) g_t \quad (18)$$

$$v_t = \beta_2 v_{t-1} + (1-\beta_2) g_t^2 \quad (19)$$

These are bias-corrected as:

$$\hat{m}_t = \frac{m_t}{1-{\beta_1}^t} \quad and \quad \hat{v}_t = \frac{v_t}{1-{\beta_2}^t} \quad (20)$$

Finally, the parameter update rule is:

$$\theta_{t+1} = \theta_t - \alpha . \frac{\hat{m}_t}{\sqrt{\hat{v}_t}+\epsilon} \quad (21)$$

Where α is the learning rate, $\beta_1 \, and \, \beta_2$ are exponential decay rates, and ϵ is a small constant for numerical stability. Adam's adaptability is particularly useful for sparse gradients and noisy data, making it very popular for training deep neural networks.

Table 1. Layer-wise architecture of the HDL-CTM technique

Layer Name	Type	Output Shape	Description
Input	Input Layer	(64, 64, 1)	Grayscale image input
Conv1	Conv2D + ReLU	(64, 64, 32)	3x3 filters, 32 channels
Pool1	MaxPooling2D	(32, 32, 32)	2x2 pool size
Conv2	Conv2D + ReLU	(32, 32, 64)	3x3 filters, 64 channels
Pool2	MaxPooling2D	(16, 16, 64)	2x2 pool size
Reshape	Flatten to Patches	(256, 64)	Split into 256 tokens of dim 64
PosEncoding	Positional Encoding	(256, 64)	Add spatial position info
TransformerEncoder	Multi-Head attention + FFN	(256, 64)	2 Transformer encoder layers
GlobalAvgPool	GlobalAveragePooling1D	(64)	Pooled token embeddings
Dropout	Dropout (0.3)	(64)	Prevent overfitting
Dense	Fully Connected + Softmax	(36)	Output class probabilities

3.4. Sign Video Recognition and Classification using HDL-CLM

Dynamic sign language recognition involves recognizing spatio-temporal attributes. More specifically, a hybrid CNN-LSTM model can combine features of both types in a single architecture using ConvLSTM2D units; Generalized versions of Long Short-Term Memory (LSTM) that use the Convolution layer directly with spatiotemporal sequences.

3.4.1. Convolutional Feature Extraction (ConvLSTM2D)

The basic LSTM is extended by ConvLSTM2D because in ConvLSTM2D, convolution is used in the gates. This enables the model to preserve spatial context (such as CNNs) and follow temporal evolution (like LSTMs). All the ConvLSTM gates are used to output the memory and filter out features of interest for both time and space.

$$i_t = \sigma(W_{xi} * X_t + W_{hi} * H_{t-1} + b_i \quad (22)$$

$$f_t = \sigma(W_{xf} * X_t + W_{hf} * H_{t-1} + b_f \quad (23)$$

$$\tilde{C}_t = tanh(W_{xc} * X_t + W_{hc} * H_{t-1} + b_c \quad (24)$$

$$C_t = f_t \cdot \, C_{t-1} + i_t \cdot \tilde{C}_t \quad (25)$$

$$o_t = \sigma(W_{xo} * X_t + W_{ho} * H_{t-1} + b_o \quad (26)$$

$$H_t = o_t \, \cdot \tanh(C_t \quad (27)$$

Where, $*$ is the convolution operator, $\cdot$ is element-wise multiplication, σ is the sigmoid function, and H_t, C_t are hidden and cell states at time t.

3.4.2. Temporal Pooling

The temporal pooling combines the feature outputs across all time steps to produce a compact, fixed-length representation of the dynamic gesture sequence. This enables the model to keep only a few of the most informative spatiotemporal features for classification.

$$f = \frac{1}{T}\sum_{t=1}^{T} H_t \quad (28)$$

3.4.3. Fully Connected Layer and Classification

The pooled features vector is then passed through a dense layer, followed by the softmax activation function. That provides a probability distribution of all available ISL classes.

$$\hat{y} = softmax(Wf + b) \quad (29)$$

Where, $\hat{y} \, \epsilon \, R^C$ is the predicted class probabilities, and C is the number of sign classes.

3.4.4. Loss Function and Optimization

Similar to HDL-CTM, the HDL-CLM also utilizes minimization of the classification error loss function and the ADAM optimizer for adaptive updates to its weights.

Table 2. Layer-wise architecture of the HDL-CLM technique

Layer Name	Type	Output Shape	Description
Input	Input Layer	(T, 64, 64, 1)	Sequence of T grayscale frames (resized to 64x64)
ConvLSTM1	ConvLSTM2D	(T, 64, 64, 32)	3x3 filters, 32 channels, return_sequences=True
Dropout1	Dropout	(T, 64, 64, 32)	Dropout rate = 0.25
ConvLSTM2	ConvLSTM2D	(T, 64, 64, 64)	3x3 filters, 64 channels, return_sequences=False
Dropout2	Dropout	(64, 64, 64)	Dropout rate = 0.25
Flatten	Flatten	(262144)	Flatten spatiotemporal output
Dense1	Dense + ReLU	(128)	Fully connected layer with 128 neurons
Dropout3	Dropout	(128)	Dropout rate = 0.5
Output	Dense + Softmax	(36)	Class probabilities for 36 ISL signs

The following algorithm describes the complete proposed framework.

Algorithm: Hybrid CNN-Transformer & CNN-LSTM for ISL Recognition

Input: ISL image and video dataset, Labels, Epochs, Learning rate, Batch size

Output: Trained CNN-Transformer and CNN-LSTM models, Predictions

1. Initialize Models
   a. Define CNN layers for spatial feature extraction in both models.
   b. For image model: Add Transformer layers for global context understanding.
   c. For video model: Add ConvLSTM2D layers for spatiotemporal feature extraction.
   d. Add fully connected output layers with Softmax for classification.

2. Preprocess Data
   a. Convert images and video frames to grayscale.
   b. Resize all inputs to 64x64 resolution.
   c. Normalize pixel values to range [0, 1].
   d. Extract T frames from videos and pad if necessary.
   e. Encode labels and split the dataset.

3. Train Models
   FOR epoch = 1 to E:

      FOR each batch in image data:

         a. Forward Pass through CNN-Transformer.
         b. Compute cross-entropy loss.
         c. Update weights using Adam optimizer.

      END FOR

      FOR each batch in video data:

         a. Forward Pass through CNN-LSTM.
         b. Compute cross-entropy loss.
         c. Update weights using Adam optimizer.

      END FOR

   END FOR

4. Test Models
   a. Predict labels using trained CNN-Transformer and CNN-LSTM models.
   b. Decode predictions to original categorical labels.

5. Evaluate Models
   a. Compute accuracy, precision, recall, and F1-score.
   b. Generate confusion matrix, ROC curve, and accuracy-loss plots.

6. Output
   a. Trained CNN-Transformer and CNN-LSTM models.
   b. Class predictions for input ISL images and videos.

4. Results and Discussion

The two proposed methods, HDL-CTM and HDL-CLM, were tested on the publicly available datasets from Kaggle Contests. HDL-CTM is validated on a sign image dataset [23], and HDL-CLM is tested on a sign video dataset [24]. The image dataset contains 3,600 images of ISL divided into 36 classes of signs, ranging from digits (the numbers '0' to '9') and alphabet (the characters 'A' to 'Z'). With 100 images from each class for uniformity, to evaluate the performance Experimental Testing: An 80:20 ratio format of the image dataset was used for the training and test set. That is, 2,880 images (80%) were utilized to train the model and to obtain discriminative features, and 720 images (20%) were employed to verify the ability of generalization of the model. The video dataset consists of a total of 720 videos selected from the same 36 classes, divided evenly with 20 samples for each class. The training was done with 5 folds to make the model stronger and more precise. We present the distribution of images and video across the 36 sign classes in Table 3, along with some sample images and video frames per class in Figures 2 and 3.

Table 3. Dataset details

Class Names	Class Label	No. of Samples
ISL Images		
0 to 9	0 to 9	1000 (Each class – 100)

'A' to 'Z'	10 to 35	2600 (Each class – 100)
Total		3600
ISL Videos		
0 to 9	0 to 9	200 (Each class – 20)
'A' to 'Z'	10 to 35	520 (Each class – 20)
Total		720

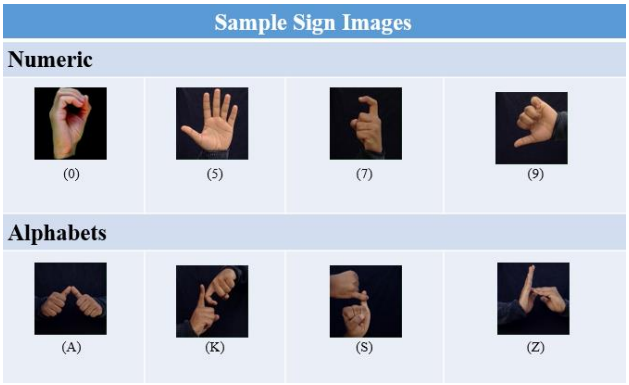


Fig. 2 Sample sign images for different classes

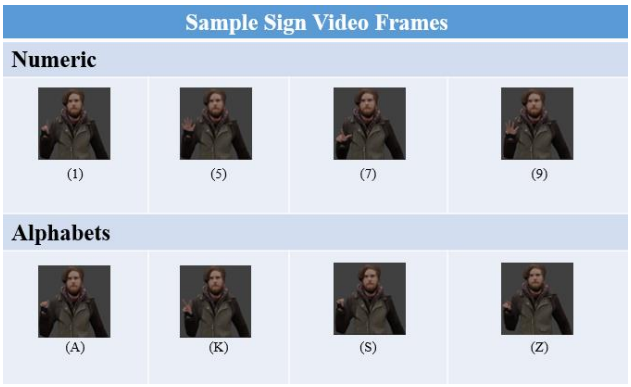


Fig. 3 Sample sign video frames for different classes

Implementation details of HDL-CTM and the HDL-CLM techniques are provided in Table 4.

Table 4. Simulation Variables

Hyperparameter	HDL-CTM	HDL-CLM
Optimizer	Adam	Adam
Learning Rate	0.001	0.001
Batch Size	64	8
Epochs	50	50
Dropout Rate	0.3	0.25, 0.5
Activation Function	ReLU, Softmax	ReLU, Softmax
Transformer Layers	2	—
Attention Heads	4	—
LSTM Units	—	64 per layer
Convolution Filter Size	3×3	3×3
Frame Count (Video)	—	30 frames
Input Size	64×64	64×64
Validation	80:20	5-fold

In Figures 4 and 5, the model presents the graphical user interface used to perform sign image prediction and video prediction using the HDL-CTM and HDL-CLM methods, respectively. The interfaces provide users a way to select, iterate, and engage with sign inputs interactively whilst receiving feedback on both predictions in real-time, as well as visually and performance-dependent.

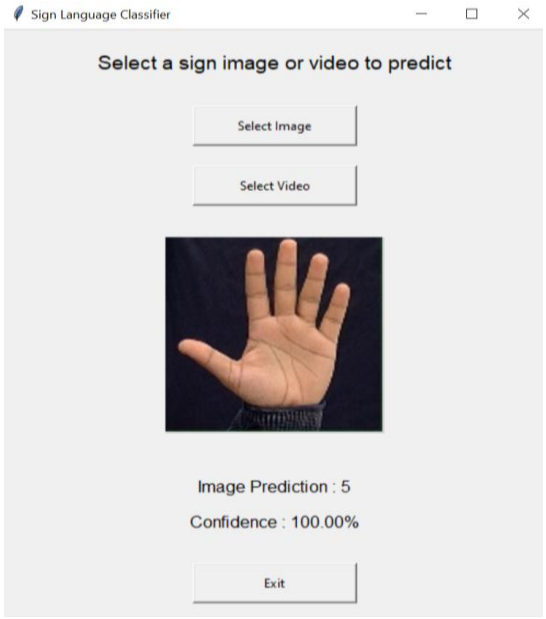


Fig. 4 GUI for Sign Image Prediction using HDL-CTM

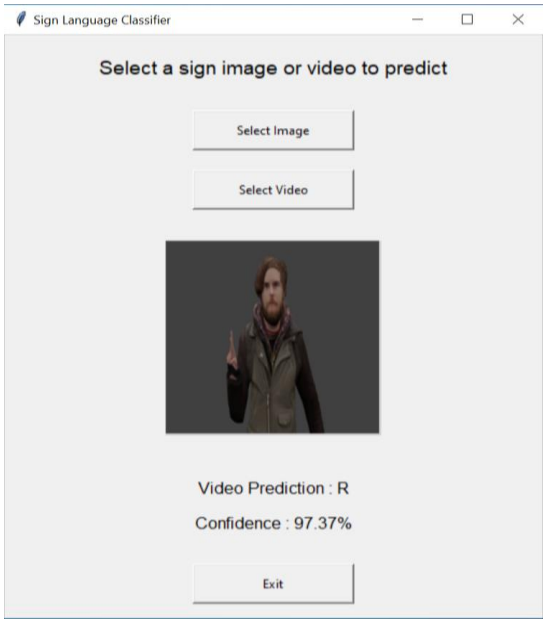
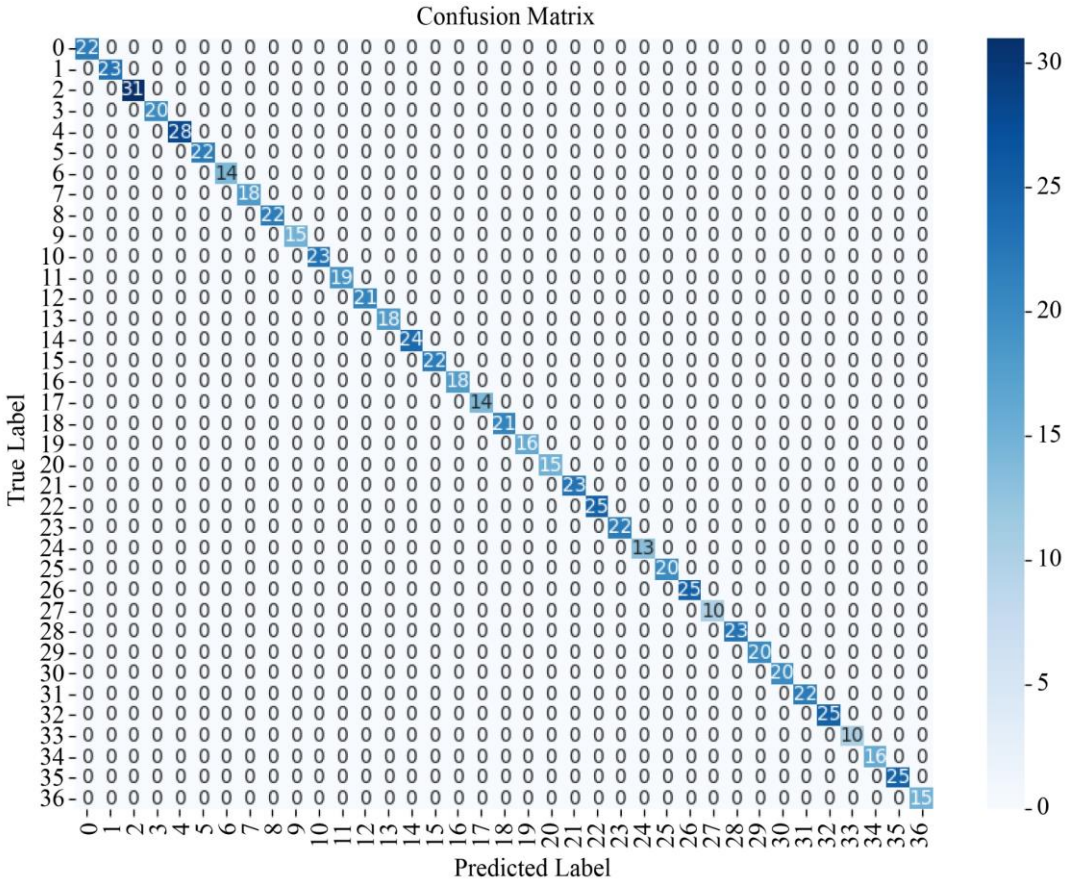


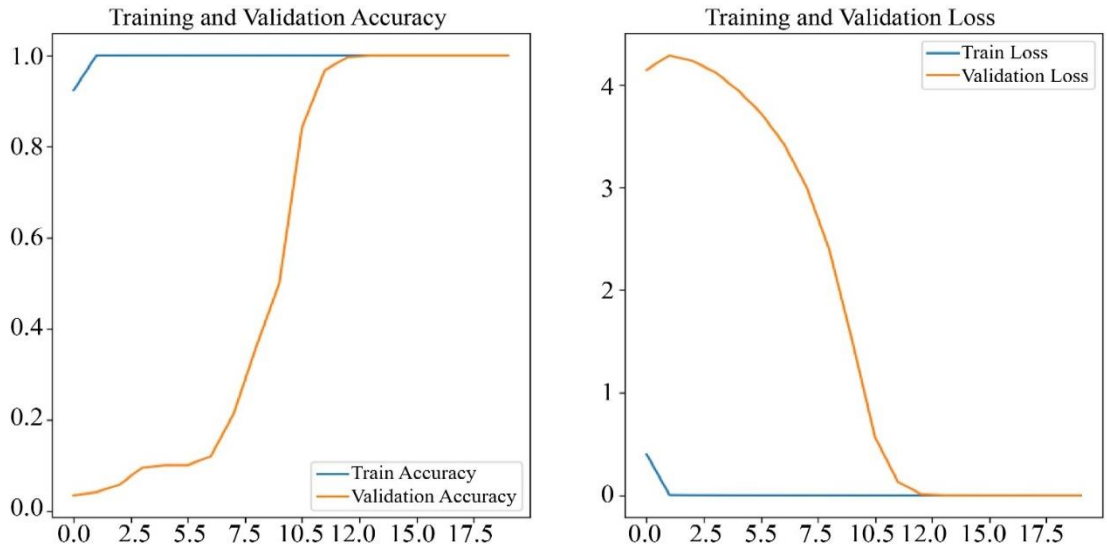
Fig. 5 GUI for Sign Video Prediction using HDL-CLM

Overall evaluation of the HDL-CTM approach is shown in Figure 6. Figure 6(a) represents the confusion matrix for all thirty-six classes, where all subclasses were identified with needed minuscule misclassifications. The accuracy and loss

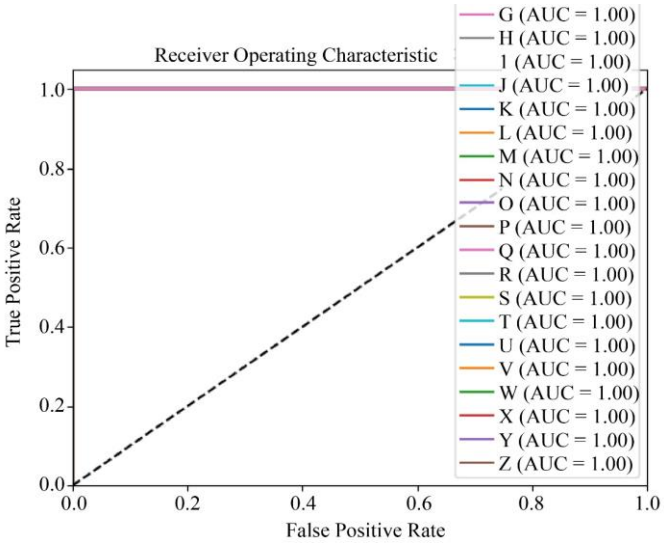
curves of the HDL-CTM model, shown in Figure 6(b), show there is high accuracy with low loss during training and validation, clearly indicating good training. Additionally, as shown in Figure 6(c), the ROC curves further support high true positive rates of several classes, indicating the stability of the model. Together, these test results confirm the predictive certainty and classification ability of the HDL-CTM framework.



(a)



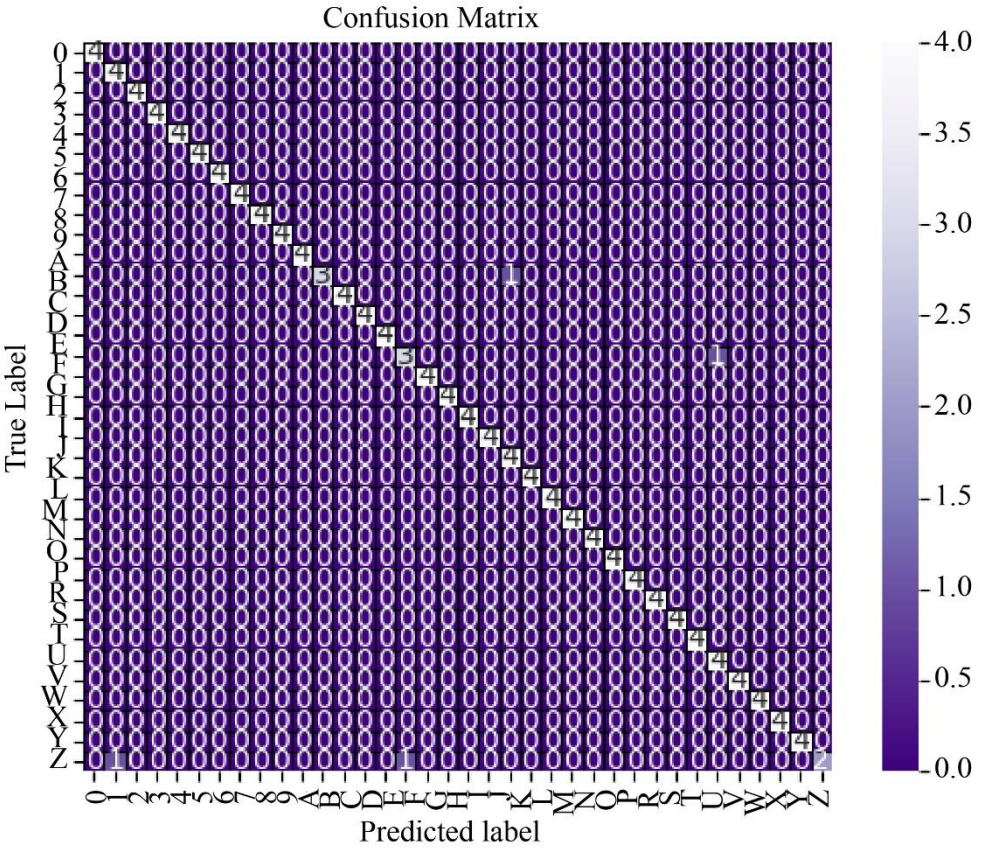
(b)



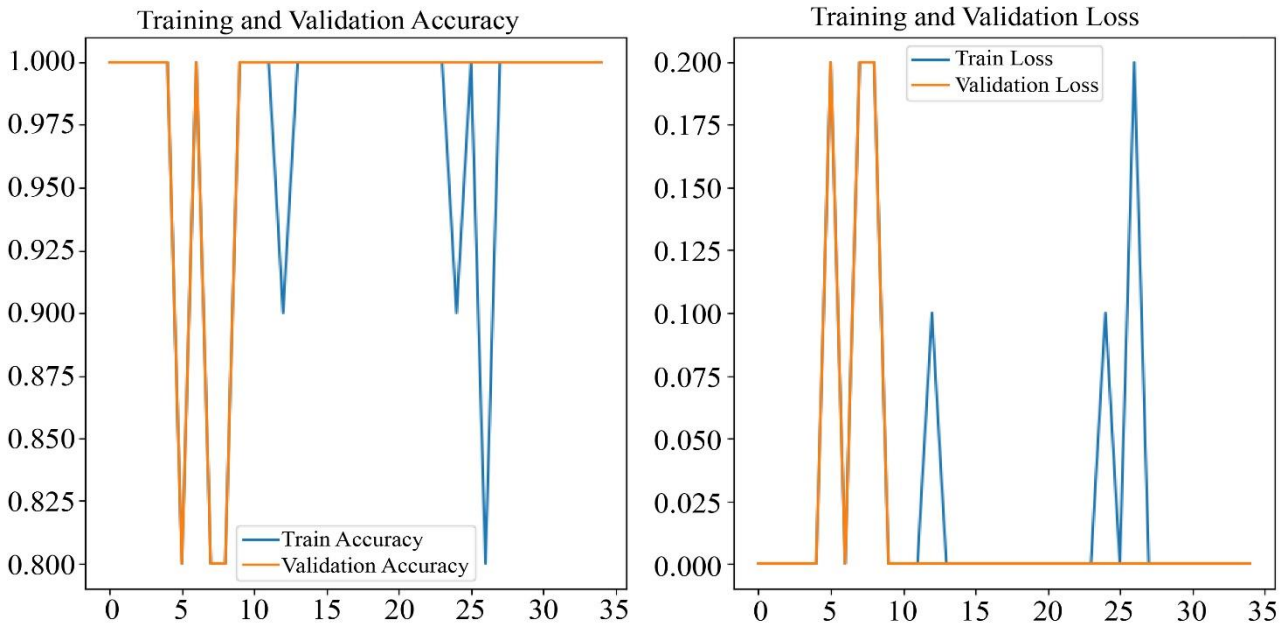
(c)

Fig. 6 Classification analysis of HDL-CTM approach (a) Confusion matrix, (b) Accuracy-Loss curve, and (c) ROC curve

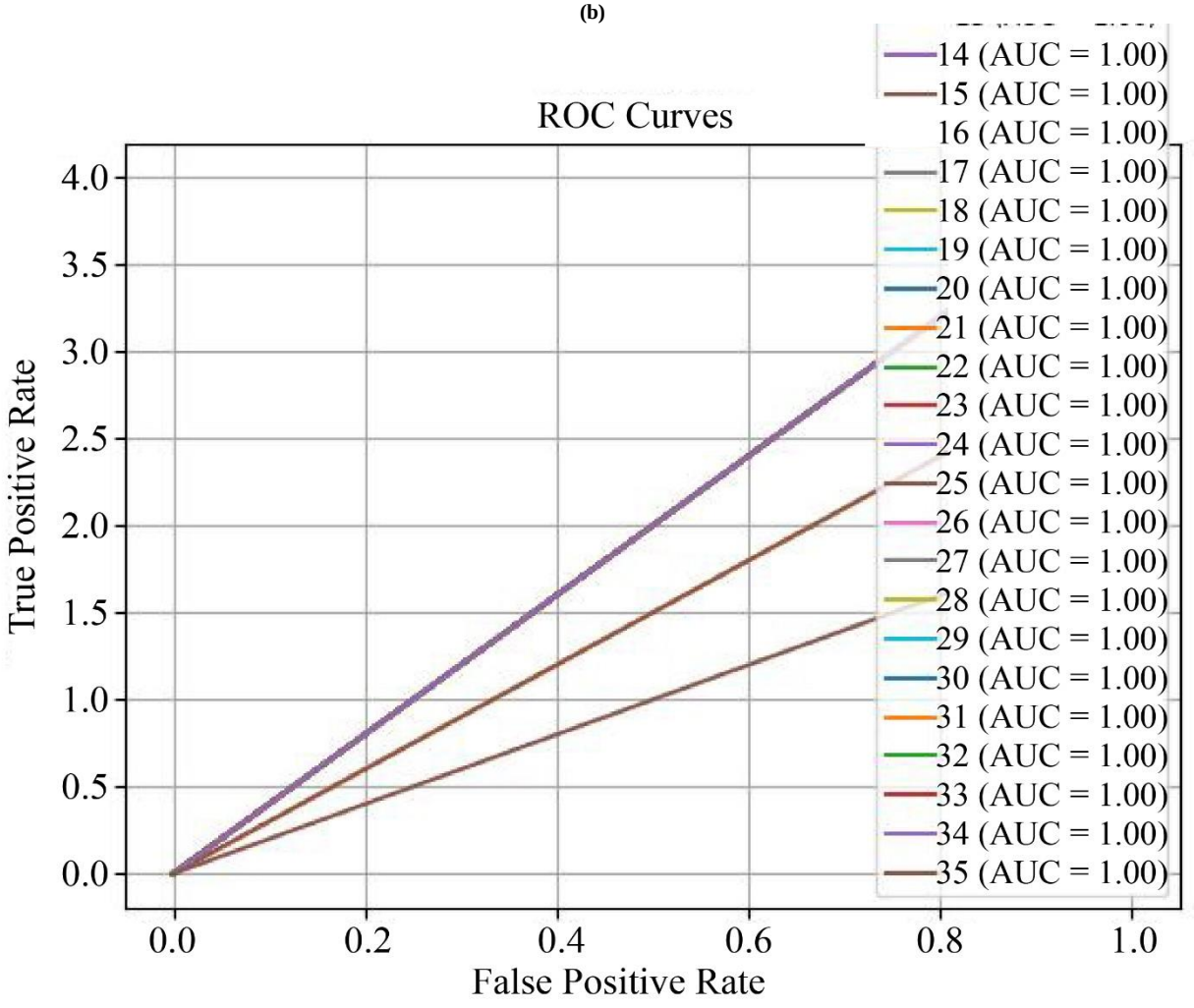
In the HDL-CLM method, the comprehensive evaluation of a model is shown in Figure 7. The model is shown in the confusion matrix (see Figure 7(a)), which indicates good learning, as all thirty-six alphanumeric sign video classes are classified correctly with minimal misclassification. The accuracy and loss plots depicted in Figure 7(b) indicate the ability of our model to generalize with high accuracy and low error rates during training while maintaining stability over the period. Besides that, the ROC curves from Figure 7(c) also validate the performance of the model, having a high sensitivity for all class labels. In general, these results show that the HDL-CLM model is accurate and effective for dynamic sign language recognition.



(a)



(b)



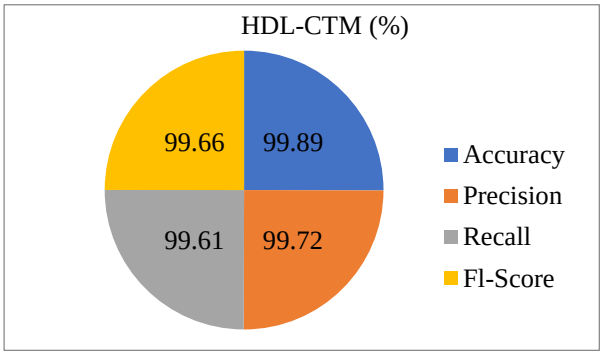
(c)

Fig. 7 Classification analysis of HDL-CLM approach, (a) Confusion matrix, (b) Accuracy-Loss curve, and (c) ROC curve.

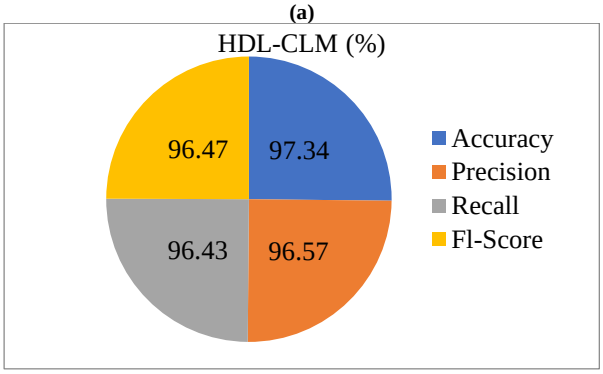
The HDL-CTM and HDL-CLM models' general classification performance can be seen in Table 5 and Figure 8. The experimental results that we list below showcase the performance of the HDL-CTM method on all dominant metrics. In particular, the HDL-CTM model showed remarkable results in static sign image identification with 99.89% accuracy, a precision rate of 99.72%, a recall rate of 99.61%, and an F1-score of 99.66%. Likewise, HDL-CLM also performed well in dynamic video-based classification with 97.34% accuracy, a precision 96.57%, a recall 96.43%, and an F1-score of 96.47%. The performance evaluation results illustrate that the two proposed models can achieve accurate, reliable, and robust static and dynamic sign language recognition tasks.

Table 5. Result analysis of Hybrid approaches with distinct measures

Metrics	HDL-CTM (%)	HDL-CLM (%)
Accuracy	99.89	97.34
Precision	99.72	96.57
Recall	99.61	96.43
F1-Score	99.66	96.47



(a)



(b)

Fig. 8 Result analysis of different metrics (a) HDL-CTM, (b) HDL-CLM.

As a comparative analysis result, the HDL-CTM model outperforms other existing solutions (Table 6 and Figure 9). As shown in the experiment results, both standalone CNN and Transformer models demonstrated lower accuracy rates of 95.76% and 96.62%, respectively. In contrast, hybrid frameworks like CNN-VGG16 and CNN-Inception V3 achieved moderate enhancements in performance rates (i.e., 97.43% and 97.68%, respectively). Of all the new proposed models, it achieved an amazing 99.89% accuracy level with them. This comprehensive investigation establishes HDL-CTM as exceptionally strong and reliable, achieving a new performance benchmark in static sign language recognition.

Table 6. Accuracy analysis of the HDL-CTM model with existing approaches

Methods	Accuracy (%)
CNN	95.76
Transformer	96.62
CNN-VGG16	97.43
CNN-INCEPTION V3	97.68
HDL-CTM	99.89

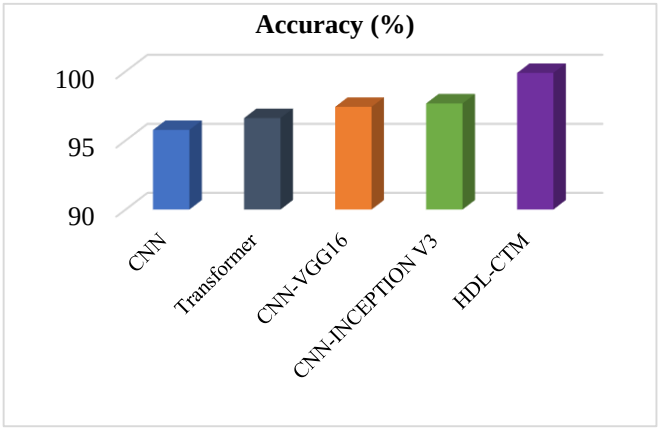


Fig. 9 Accuracy analysis of the HDL-CTM method with other techniques

Table 7 and Figure 10 summarize the performance analysis comparative study, confirming clearly that the HDL-CLM offers a much higher percentage of superiority over all other alternative methods reported in the literature. In 2016, a typical CNN with three convolutional layers only achieved an accuracy of 93.69% while the LSTM alone gave approximately the same result at best accuracy (~94.27%). However, hybrid architectures such as CNN-VGG16 and CNN-Inception V3 showed slight improvements in identification accuracies of ~94.53% and ~95.14%, respectively. Surprisingly, this latest HDL-CLM model saw an enormous performance boost, with a 97.34% accuracy rate! The results also provide evidence for HDL-CLM's performance as a new high-reliability and high-performance approach, which establishes a milestone for dynamic sign language recognition.

Table 7. Accuracy analysis of the HDL-CLM model with existing approaches

Methods	Accuracy (%)
CNN	93.69
LSTM	94.27
CNN-VGG16	94.53
CNN-INCEPTION V3	95.14
HDL-CLM	97.34

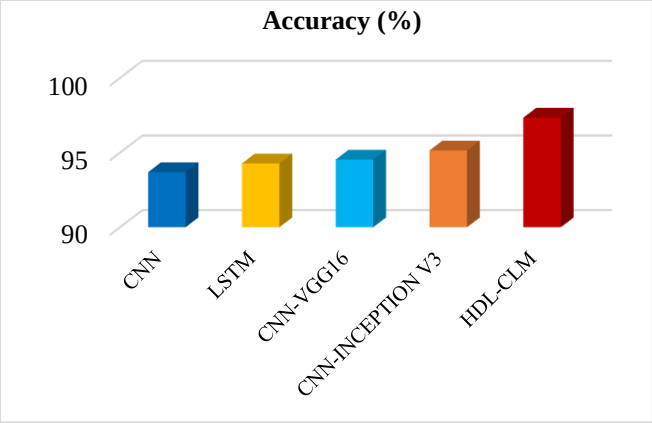


Fig. 10 Accuracy analysis of the HDL-CLM method with other techniques

The experiments were performed on diverse datasets with respect to hand orientations, alternating signing speeds, background settings, and illumination conditions, in order to evaluate signer and environment independence. While the datasets mainly consist of isolated sign samples, normalization, followed by spatial resizing and temporal frame sampling, limits the sensitivity to the stable aspect of words that define a large portion of their lexical meaning. In addition to that, samples of the same class are uniformly spread in training and testing folds by means of stratified splitting data on image-based recognition and the K-fold cross-validation method for video-based recognition, which enhances generalization. The consistently high performance across all sign classes further supports that the proposed models are robust to moderate signer and environmental variation shown in the dataset. Nonetheless, adding to the diversity of signers and using real-world recording conditions continues to be an important direction of future research to improve signer-independent recognition.

4.1. Performance Improvement over Existing Methods

The proposed HDL-CTM and HDL-CLM frameworks are task-oriented hybrid architecture designs, which predominantly increase their performance. All existing methods have one single model trying to predict both static and dynamic tasks, whereas our HDL-CTM framework adopts a dedicated CNN–Transformer model for static sign images tailored to this modality's characteristics, and a separate CNN–LSTM model for dynamic sign videos tailored to the respective modality characteristics. In CNN–Transformer, representations learned with such fine-grained spatial features and global contextual dependencies through self-attention enable better discrimination of visually similar alphanumeric signs, while the CNN–LSTM architecture learns spatiotemporal dependencies in video sequences by learning a joint representation of the spatial and temporal lucas.

Furthermore, robust processing techniques like normalization, image resizing, and temporal fragment sampling have also been proposed, which minimize class internal variations of images over time, promoting generalization during context learning. For image recognition, stratified data splitting was performed, while K-fold cross-validation was used in video recognition to ensure proper evaluation and minimize overfitting. Additionally, the learning rates were tuned using the Adam optimizer for stable convergence. Therefore, when compared with the state-of-the-art existing methods (some of which depend on data availability, computation complexity, or accessibility to various modalities), our proposed visual-only-based highly accurate and efficient approach holds promise for a potential efficient deployment in practical usages of assistive communication systems.

4.2. Ablation-Based Statistical Analysis

The ablation study shows (both separately and where they work together) the contributions of the components CNN, Transformer, and temporal modeling. Although standalone architectures produce competitive performance, the joint use of those attention-enhanced hybrid models achieves much better recognition accuracy. The experimental results prove that the proposed modality-aware hybrid designs outperform generic and partial configurations. Table 8 shows the results of the ablation study for the proposed model.

Table 8. Ablation Study Results for Proposed HDL-CTM and HDL-CLM Models

Model Variant	CNN	Transformer	LSTM / ConvLSTM	Attention	Accuracy (%)	Precision (%)	Recall (%)	F1-score (%)
CNN only	✓	-	-	-	95.76	95.41	95.28	95.34
Transformer only	-	✓	-	✓	96.62	96.18	96.05	96.11
CNN + LSTM	✓	-	✓	-	94.27	93.84	93.69	93.76
CNN + ConvLSTM	✓	-	✓	-	95.14	94.77	94.53	94.65
CNN + Transformer (without attention tuning)	✓	✓	-	-	97.68	97.31	97.05	97.18
HDL-CTM (CNN + Transformer + Attention)	✓	✓	-	✓	**99.89**	**99.72**	**99.61**	**99.66**

HDL-CLM (CNN + ConvLSTM + Attention)	✓	-	✓	✓	**97.34**	**96.57**	**96.43**	**96.47**

5. Conclusion

This paper presents an elaborate hybrid deep learning framework designed for reliable and consistent Indian Sign Language (ISL) recognition in both static images and dynamic video streams. To tackle limitations of classical approaches, two novel architectures based on these principles (HDL-CTM and HDL-CLM) are proposed to exploit the advantages of convolutional networks via a sequential architecture. For sign images, HDL-CTM captures spatial and contextual cues of data with a combination of CNN and Transformer layers; on the other hand, HDL-CLM combines CNN with a ConvLSTM layer to receive spatiotemporal cues from sign videos. Both models outperformed on 3,600 static sign images and 720 sign videos of publicly available ISL datasets in an experimental study. HDL-CTM obtained a 99.89% accuracy, 99.72% precision, 99.61% recall, and an F1-score of 99.66%. In a similar fashion, HDL-CLM was also shown to have high accuracy—97.34% with a very high precision—96.57%, recall—96.43%, and F1-score, which is indicative of the strength and generalization ability of the models, were able to confirm this general ranking order.

Furthermore, competitive comparison with other up-to-date deep learning algorithms such as CNN, LSTM, VGG16, and InceptionV3 demonstrates the superior efficiency and robustness of HDL models proposed in this paper. The models efficiently classify and are stable for each of the 36 ISL alphanumeric classes, confirmed by confusion matrices, ROC curves, and training–validation performance plots. The combination of class balancing, k-fold cross-validation, and the proper use of the Adam optimizer also greatly improved model stability and optimization. The results highlight the capability of hybrid deep learning techniques to enable the improvement of system accuracy for sign language recognition, leading towards the development of inclusive communications systems among hearing and speech-impaired individuals. Moreover, the learned representations have the potential to be applied to crosslingual and multilingual sign language recognition scenarios, providing a scalable building block for future multimodal and language-independent assistive communication systems.

Conflicts of Interest

The authors declare that they have no conflicts of interest to report regarding the present study.

Funding Statement

This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

Acknowledgements

The authors gratefully acknowledge the support and guidance provided by their respective institutions throughout the research process.

Declarations

- **Ethics approval and consent to participate**
  Not applicable. The study does not involve human participants, animals, or sensitive data requiring ethics approval.
- **Consent for publication**
  Not applicable. No identifiable personal data is included in this manuscript.
- **Availability of data and material**
  The dataset used in this study is publicly available and can be accessed through the Kaggle platform. Specific preprocessing steps and code are available upon request from the corresponding author.
- **Competing interests**
  The authors declare no competing interests related to this work.
- **Authors' contributions**
  S.P. Balamurugan: Conceived the study, performed data preprocessing, experimental analysis, and prepared visualizations.

  S. Jayalakshmi: Designed the model architecture, wrote the manuscript, and conducted the statistical evaluation.

  Both authors read and approved the final manuscript.

References

[1] Harsh Kumar Vashisth et al., "Hand Gesture Recognition in Indian Sign Language using Deep Learning," *Engineering Proceedings*, vol. 59, no. 1, pp. 1-11, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[2] Prachi Pramod Waghmare et al., "Deep Learning Approach for Combined Indian Sign Language Recognition and Video Generation Model," *International Journal of Intelligent Systems and Applications in Engineering*, vol. 12, no. 4, pp. 3296-3302, 2024. [Publisher Link]

[3] S. Nithyanandh, "AI-Driven Indian Sign Language Recognition using Hybrid CNN-BiLSTM Architecture for Divyangjan," *International Journal of Emerging Science and Engineering (IJESE)*, vol. 14, no. 1, pp. 14-25, 2025. [CrossRef] [Publisher Link]

[4] Jay Joshi, and Dhaval Patel, "Transformer-based Deep Learning Approach for Indian Sign Language Recognition," *International Journal of All Research Education & Scientific Methods*, vol. 11, no. 12, pp. 2304-2310, 2023. [CrossRef] [Publisher Link]

[5] Bunny Saini et al., "A Comparative Analysis of Indian Sign Language Recognition using Deep Learning Models," *Forum for Linguistic Studies*, vol. 5, no. 1, pp. 197-222, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[6] Ahmed Mateen Buttar et al., "Deep Learning in Sign Language Recognition: A Hybrid Approach for the Recognition of Static and Dynamic Signs," *Mathematics*, vol. 11, no. 17, pp. 1-20, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[7] Md Azher Uddin, Ryan Denny, and Joolekha Bibi Joolee, "Deep Spatiotemporal Network-based Indian Sign Language Recognition from Videos," *Lecture Notes in Networks and Systems*, pp. 171-181, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[8] Hao Chen et al., "SignVTCL: Multi-Modal Continuous Sign Language Recognition Enhanced by Visual-Textual Contrastive Learning," *arXiv preprint*, pp. 1-12, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[9] Maher Jebali, Abdesselem Dakhli, and Wided Bakari, "Deep Learning-based Sign Language Recognition System using Both Manual and Non-Manual Components Fusion," *AIMS Mathematics*, vol. 9, no. 1, pp. 2105-2122, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[10] Sunusi Bala Abdullahi et al., "Spatial-Temporal Feature-based End-to-End Fourier Network for 3D Sign Language Recognition," *Expert Systems with Applications*, vol. 248, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[11] Liqing Gao et al., "Cross-Modal Knowledge Distillation for Continuous Sign Language Recognition," *Neural Networks*, vol. 179, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[12] Sharvani Srivastava et al., "Continuous Sign Language Recognition System using Deep Learning with Mediapipe Holistic," *arXiv preprint*, pp. 1-14, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[13] Anudyuti Ghorai et al., "Indian Sign Language Recognition System using Network Deconvolution and Spatial Transformer Network," *Neural Computing and Applications*, vol. 35, no. 1, pp. 20889-20907, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[14] Zaid Saad Bilal et al., "Advancements in Arabic Sign Language Recognition: A Method based on Deep Learning to Improve Communication Access," *Journal of Internet Services and Information Security*, vol. 14, no. 4, pp. 278-291, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[15] Edwin Shalom Soji, and T. Kamalakannan, "Efficient Indian Sign Language Recognition and Classification using Enhanced Machine Learning Approach," *International Journal of Critical Infrastructures*, vol. 20, no. 2, pp. 125-138, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[16] Arashta Hussain, Nimakhi Saikia, and Chandana Dev, "Advancements in Indian Sign Language Recognition Systems: Enhancing Communication and Accessibility for the Deaf and Hearing Impaired," *Asian Journal of Electrical Sciences*, vol. 12, no. 2, pp. 37-49, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[17] Yanqiong Zhang, and Xianwei Jiang, "Recent Advances on Deep Learning for Sign Language Recognition," *Computer Modeling in Engineering & Sciences*, vol. 139, no. 3, pp. 2399-2450, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[18] Fatma M. Najib, "A Multi-Lingual Sign Language Recognition System using Machine Learning," *Multimedia Tools and Applications*, vol. 84, no. 24, pp. 27987-28011, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[19] Matyáš Boháček, and Marek Hrúz, "Sign Pose-based Transformer for Word-Level Sign Language Recognition," *2022 IEEE/CVF Winter Conference on Applications of Computer Vision Workshops (WACVW)*, Waikoloa, HI, USA, pp. 182-191, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[20] Diksha Kumari, and Radhey Shyam Anand, "Isolated Video-based Sign Language Recognition using a Hybrid CNN-LSTM Framework based on Attention Mechanism," *Electronics*, vol. 13, no. 7, pp. 1-13, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[21] Weichao Zhao et al., "Self-Supervised Representation Learning with Spatial-Temporal Consistency for Sign Language Recognition," *IEEE Transactions on Image Processing*, vol. 33, pp. 4188-4201, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[22] Pranav Chaudhari et al., "Sign Language Recognition using Spiking Neural Networks," *Procedia Computer Science*, vol. 235, pp. 2674-2683, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[23] Vaishnavi Sonawane, Indian Sign Language Dataset, 2020. [Online]. Available: https://www.kaggle.com/datasets/vaishnaviasonawane/indian-sign-language-dataset

[24] UniSerj · Community Prediction Competition, Sign Language Recognition, 2026. [Online]. Available: https://www.kaggle.com/competitions/sign-language-recognition