COLLEGE OF ENGINEERING, GUINDY ANNA UNIVERSITY, CHENNAI - 25

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

Real-Time Indian Sign Language Recognition Using Efficient Multi-Feature Attention Mechanism

Under the guidance of

Dr. C. Valliyammai

Professor

Department of Computer Science and Engineering

Submitted By

2023103046 Rajesh A

2023103546 Kathir Kalidass B

2023103579 Sathyanarayanan P

2023103703 Nandha S

Abstract

Indian Sign Language (ISL) recognition plays a vital role in improving communication between the hearing-impaired community and the general population. This work presents a realtime Indian Sign Language Recognition system based on an Efficient Multi-Feature Attention Mechanism. The proposed approach processes live video input by extracting a fixed sequence of RGB frames, which are used for parallel feature extraction. Spatial features are learned using an ImageNet-pretrained EfficientNet through transfer learning, while temporal features are captured using a CNN–RNN framework to model motion information across frames. An attention mechanism is employed to assign adaptive importance weights to both spatial and temporal feature streams, enabling the model to focus on the most discriminative information. The weighted features are then fused and passed to a GRU-based sequence modeling network to learn gesture-level temporal dependencies. Finally, a fully connected layer with softmax activation classifies the input video into the corresponding ISL gesture class. The proposed system is optimized for real-time performance and evaluated using crossvalidation, demonstrating effective recognition accuracy while maintaining low computational complexity. This framework provides a robust and efficient solution for real-time ISL gesture recognition and can be extended to continuous sign language translation in future work.

1. Introduction

Sign language is a visual-gestural language widely used by the hearing-impaired community for communication. Indian Sign Language (ISL) consists of complex hand shapes, movements, and facial expressions that vary across time, making automated recognition a challenging task. Manual interpretation of sign language requires trained interpreters, which may not always be available in real-world scenarios. Recent advances in deep learning have enabled significant improvements in video-based gesture recognition by effectively modeling spatial and temporal information. However, many existing systems either focus solely on spatial features extracted from individual frames or rely on computationally expensive temporal models that are unsuitable for real-time deployment. This project aims to develop a real-time ISL recognition system that balances accuracy and efficiency. Inspired by the base paper, the proposed approach employs dual feature extraction using EfficientNet for spatial representation and CNN-RNN for temporal modeling. A multi-feature attention mechanism is introduced to enhance discriminative learning by dynamically weighting spatial and temporal features. The system is designed to recognize isolated ISL gestures and is optimized for real-time usage without relying on skeleton extraction or heavy language-level processing modules.

2. Literature Review

1. Overview of Vision-Based Sign Language Translation

Vision-based Sign Language Translation (SLT) has gained increasing attention as a means to bridge communication gaps between hearing-impaired individuals and the general population. Recent advances in deep learning have enabled effective recognition and translation of sign languages from video data. Convolutional Neural Networks (CNNs), Recurrent Neural Networks (RNNs), and attention-based architectures have demonstrated strong performance in extracting

spatial and temporal features from sign language videos. While extensive research exists for American Sign Language (ASL) and Chinese Sign Language (CSL), Indian Sign Language (ISL) remains comparatively under-explored due to linguistic diversity and limited annotated datasets.

2. Indian Sign Language Datasets and Benchmarks

The lack of large-scale standardized datasets has been a major challenge in ISL research. Joshi et al. introduced the iSign dataset, one of the most comprehensive benchmarks for ISL processing, containing over 118,000 ISL–English video– text pairs. The dataset supports multiple tasks, including SignVideo-to-Text, SignPose-to-Text, and Text-to-Pose translation, and provides baseline models for fair comparison and reproducibility. Consequently, iSign has become a foundational dataset for recent ISL recognition and translation systems.

3. Vision-Based ISL Translation Systems

Several studies have explored real-time vision-based ISL translation using deep learning approaches. CNN–LSTM hybrid architectures have been widely adopted to extract spatial and temporal features from video streams, demonstrating effective performance for isolated and short continuous sign sequences. However, these approaches face limitations in modeling long-range temporal dependencies and sentence-level grammatical structures.

4. Advanced Deep Learning Approaches

Recent surveys published in IEEE Access provide a comprehensive analysis of vision-based sign language recognition techniques. Existing methods are categorized into isolated and continuous recognition systems, with increasing emphasis on pose-based features, multimodal representations, and attention mechanisms. Key challenges identified include signer independence, occlusion handling, background variation, and dataset scarcity, which are particularly pronounced in ISL research. Transformer-based architectures have emerged as an alternative to CNN–RNN pipelines for continuous sign language translation. Studies presented at ACM Multimedia demonstrate that attention-based models can effectively model long-range temporal dependencies and improve sentence-level translation accuracy. However, the high computational complexity and large data requirements of Transformer-based models limit their suitability for low-resource and real-time ISL systems.

5. Multimodal Sign Language Translation

Multimodal sign language translation systems integrate visual, pose, and linguistic information to improve recognition robustness. Frameworks combining vision encoders with language models have demonstrated improved performance under signer and background variations. Despite these advantages, multimodal approaches increase system complexity and require large annotated datasets, posing challenges for practical ISL deployment.

7. Summary of Research Gaps

The reviewed literature highlights several limitations, including the scarcity of large-scale ISL datasets, limited bidirectional translation systems, inadequate handling of continuous signing, and the high computational complexity of existing models. These limitations indicate the need for efficient and scalable ISL recognition frameworks tailored to low-resource and real-time environments.

3. Methodology

The proposed methodology describes a comprehensive visionbased framework for Indian Sign Language (ISL) recognition and translation using deep learning techniques. The system is designed to process RGB video sequences containing sign language gestures and convert them into meaningful textual and spoken outputs. The methodology follows the core idea of the base paper "Deep Learning-Based Sign Language Recognition Using Efficient Multi-Feature Attention Mechanism" and extends it by incorporating sequence modeling, alignment, language processing, and speech generation modules. The primary objective of the proposed system is to accurately learn both spatial visual patterns and temporal motion characteristics present in ISL videos. Sign language gestures are inherently dynamic; therefore, a single frame is insufficient to represent a sign. To address this challenge, the system adopts a multi-stage learning pipeline that extracts frame-level visual features, models temporal dependencies across frames, focuses on important gesture components through attention mechanisms, and finally translates the learned representations into readable and audible outputs.

The complete methodology consists of the following stages:

1. Video input acquisition and preprocessing
2. Spatial feature extraction using EfficientNet
3. Temporal feature extraction using CNN–RNN
4. Attention-based feature weighting
5. Feature fusion 6. Sequence modeling using Transformer
7. Temporal alignment using CTC
8. Decoding and language refinement
9. Text and speech output generation

4.Architecture Diagram

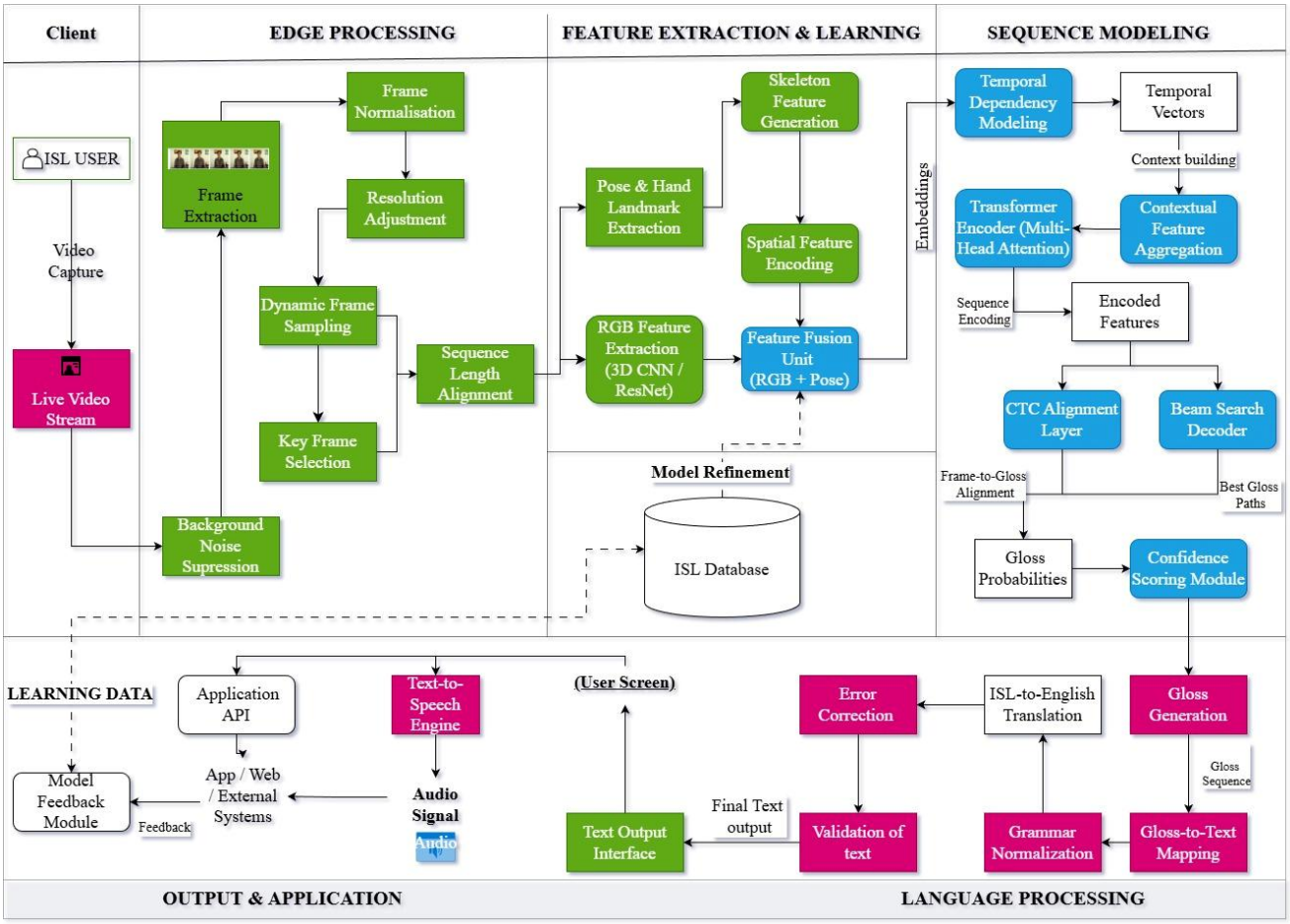

Fig 1. Architecture diagram of proposed model

5. Detailed Design

5.1 Video Input Acquisition and Preprocessing

The input to the proposed system is an RGB video sequence captured using a standard camera device. The video represents an ISL gesture sequence performed by a signer. Since raw video data cannot be directly processed by deep learning models, several preprocessing steps are applied to transform the video into a structured and learnable format.

1) Frame Extraction : The input video is first decomposed into a sequence of individual frames using uniform frame sampling. This step ensures that temporal information is preserved while avoiding redundant frames that may not contribute meaningful information. A fixed number of frames is extracted from each video to maintain uniformity across samples. This fixed-length representation is essential for efficient batch-based training.

2) Frame Resizing and Normalization: Each extracted frame is resized to a predefined resolution suitable for convolutional neural networks. Resizing ensures consistent spatial dimensions across all inputs. Pixel normalization is then applied to scale pixel intensity values to a standard range. This normalization step improves numerical stability during training and reduces sensitivity to variations in lighting and background conditions.

3) Temporal Sequence Standardization ISL videos may vary in duration depending on the gesture and signer. To address this variability, all frame sequences are standardized to a fixed temporal length. Shorter sequences are padded, while longer sequences are truncated. This step ensures that all inputs have consistent temporal dimensions, enabling efficient model training and inference

5.2 Spatial Feature Extraction Using EfficientNet

Spatial feature extraction is a critical component of the proposed system, as sign language gestures rely heavily on hand shape, orientation, and visual appearance. To extract robust spatial representations, the system employs EfficientNet as a transfer learning backbone. EfficientNet is pretrained on the ImageNet dataset, which contains millions of labeled images across diverse object categories. Through transfer learning, the model leverages previously learned visual representations and adapts them to the sign language recognition task. Each preprocessed video frame is independently passed through EfficientNet, where multiple convolutional layers extract hierarchical visual features ranging from low-level edges to high-level semantic patterns. The output of EfficientNet is a compact feature vector that encodes spatial information such as finger configuration, palm orientation, and overall hand appearance. Using EfficientNet provides two major advantages: high recognition accuracy and reduced computational complexity, making it suitable for realtime ISL applications.

5.3 Temporal Feature Extraction Using CNN–RNN

While spatial features describe what appears in each frame, temporal features describe how gestures evolve over time. To capture motion dynamics, the proposed system employs a CNN–RNN architecture. In this stream, a lightweight convolutional neural network first extracts frame-level features from each video frame. These features are then fed sequentially into a recurrent neural network. The recurrent network processes the frame features in temporal order and learns dependencies across consecutive frames. Through this temporal modeling, the system captures essential motion information such as direction of hand movement, speed of execution, and transitions between gesture components. This is particularly important for distinguishing gestures that have similar visual appearance but differ in motion patterns.

5.4 Multi-Feature Attention Mechanism

The spatial and temporal feature streams provide complementary information. However, not all frames or features contribute equally to gesture recognition. To address this, a multi-feature attention mechanism is introduced. The attention module computes importance weights for both spatial and temporal features. These weights are learned during training and dynamically adjusted based on the relevance of features to the recognition task. Important frames and discriminative features receive higher weights, while redundant or less informative features are suppressed. By focusing computational resources on the most relevant information, the attention

mechanism improves recognition accuracy and enhances robustness against noise and background variations.

5.5 Feature Fusion

After attention weighting, the spatial and temporal features are combined using a feature fusion strategy. This fusion integrates appearance-based information from EfficientNet with motion-based information from the CNN– RNN stream. The resulting fused representation forms a unified spatiotemporal feature sequence that captures both what the gesture looks like and how it moves over time. This fused feature sequence serves as the input to higher-level sequence modeling components.

5.6 Sequence Modeling Using Transformer

To model long-range temporal dependencies and contextual relationships across the entire gesture sequence, the fused feature sequence is passed to a Transformer encoder. The Transformer employs multi-head self-attention to analyze relationships between all frames simultaneously. Positional encoding is applied to preserve temporal order information within the sequence. Unlike traditional recurrent models, the Transformer processes the sequence in parallel, enabling efficient modeling of long and complex gesture sequences. This capability is particularly beneficial for continuous ISL translation tasks.

5.7 Temporal Alignment Using CTC

The output of the Transformer encoder consists of frame-level predictions that must be aligned with the corresponding output labels. To perform this alignment, a Connectionist Temporal Classification (CTC) layer is used. CTC allows the model to learn alignments between input sequences and output label sequences without requiring exact frame-level annotations. This flexibility is crucial for sign language recognition, where precise alignment between frames and labels is difficult to obtain.

5.8 Decoding Using Beam Search

During inference, beam search decoding is applied to the CTC outputs. Beam search explores multiple candidate sequences and selects the most probable output sequence based on cumulative probability scores. This decoding strategy improves prediction reliability compared to greedy decoding.

5.9 Language Processing and Grammar Refinement

The decoded output sequence is converted into readable text using a gloss-to-text mapping module. Since sign language structure differs from spoken language structure, grammar refinement is applied to improve sentence fluency and syntactic correctness. This step ensures that the generated text is understandable and suitable for end users.

5.10 Output Generation and Text-to-Speech Conversion

The final refined text is displayed as the textual output of the system. Additionally, a text-to-speech module converts the text into audible speech. This enables seamless communication

between signing users and non-signing individuals and enhances the practical usability of the system.

6. Module Design

This chapter describes the functional modules of the proposed Indian Sign Language recognition system. Each module is responsible for a specific stage in the processing pipeline, ranging from video acquisition to language and audio output generation.

6.1 Video Acquisition and Preprocessing

6.1.1 Frame Extraction and Normalization

This module is responsible for converting the raw video input into a structured and standardized format suitable for deep learning models. The system accepts an RGB video stream captured using a standard camera. Since deep learning models cannot directly operate on continuous video streams, the video is first decomposed into individual frames.

Frame extraction is performed using uniform sampling, where a fixed number of frames are selected from each video sequence. Uniform sampling ensures that temporal information is preserved while avoiding redundant frames that do not contribute meaningful information. This step is especially important for real-time processing, as it reduces computational overhead without losing critical motion cues.

Once extracted, each frame is resized to a predefined spatial resolution compatible with convolutional neural networks. Frame normalization is then applied by scaling pixel intensity values to a standard range. Normalization improves numerical stability during training and ensures consistent feature learning across videos captured under varying lighting conditions.

Finally, all frame sequences are standardized to a fixed temporal length using padding or truncation, enabling efficient batch processing during training and inference.

Output:
A fixed-length sequence of normalized RGB frames.

6.1.2 Pose and Landmark Extraction

This sub-module extracts structured motion and positional information from the video frames. From each frame, key body, hand, and facial landmarks are detected to represent the signer’s movements in a compact and meaningful form. These landmarks encode geometric relationships such as joint positions, hand orientation, and finger movement patterns.

Pose and landmark information provides complementary motion cues that are invariant to background noise and illumination changes. By representing gestures in terms of relative positions and movements, the system improves robustness to environmental variations. The extracted landmark features are temporally aligned with the corresponding RGB frames to preserve synchronization between visual appearance and motion information.

Output:
A sequence of landmark feature vectors aligned with video frames.

6.2 Feature Extraction and Fusion

6.2.1 RGB Feature Extraction using CNN

This module extracts spatial appearance features from the RGB frames using a Convolutional Neural Network (CNN). Each normalized frame is passed through multiple convolutional layers that learn hierarchical visual representations. Early layers capture low-level features such as edges and textures, while deeper layers learn high-level semantic features such as hand shape, orientation, and configuration.

The CNN transforms each frame into a compact feature vector that encodes visual characteristics relevant to sign language gestures. These features capture fine-grained appearance details that are critical for distinguishing between visually similar signs.

Output:
Frame-wise RGB feature vectors.

6.2.2 Pose Feature Extraction and Attention-Based Fusion

This sub-module processes the landmark feature vectors extracted in the preprocessing stage and integrates them with RGB features using an attention-based fusion mechanism. Pose features capture motion trajectories and relative positional changes, while RGB features provide detailed appearance information.

An attention mechanism is applied to dynamically assign importance weights to both RGB and pose features. During training, the model learns which feature type contributes more to recognizing a particular sign. For example, signs with subtle finger movements may rely more on landmark features, while signs with distinctive hand shapes may rely more on RGB features.

The attention-based fusion module suppresses irrelevant or noisy features and emphasizes discriminative information. The weighted RGB and pose features are then combined to form a unified spatiotemporal representation.

Output:
Fused feature sequence containing both appearance and motion information.

6.3 Continuous Sign Recognition

6.3.1 Temporal Modeling using Transformer or BiLSTM

This module models the temporal evolution of the fused feature sequence. Since sign language gestures unfold over time, temporal modeling is essential for understanding gesture progression and context.

Two alternative architectures can be employed:

- **BiLSTM:** Processes the feature sequence in both forward and backward directions, capturing past and future temporal dependencies.

- **Transformer:** Uses multi-head self-attention to model long-range dependencies across the entire sequence simultaneously, without relying solely on recurrence.

Temporal modeling enables the system to learn motion patterns, transitions between gesture components, and contextual relationships between frames. This is particularly important for continuous sign recognition, where multiple signs appear in a single video sequence.

Output:
Context-aware temporal feature sequence.

6.3.2 Sign Prediction using CTC Decoder

The temporally modeled feature sequence is passed to a Connectionist Temporal Classification (CTC) decoder. CTC enables sequence-to-sequence learning without requiring explicit alignment between frames and output labels.

During training, the CTC loss function allows the model to learn the most probable label sequence corresponding to the input feature sequence. During inference, a decoding strategy such as beam search is applied to generate the final predicted sign or sign sequence.

CTC is particularly effective for continuous sign recognition, as it handles variable-length input and output sequences and naturally accounts for timing variations in gesture execution.

Output:
Predicted sign or sign sequence.

6.4 Language and Audio Output

6.4.1 Sentence Correction and Caption Buffer

The predicted sign sequence is converted into textual form using a gloss-to-text mapping mechanism. Since the grammatical structure of sign language differs from spoken language, the generated text may contain syntactic inconsistencies.

A sentence correction module refines the generated text by correcting word order, grammatical errors, and incomplete phrases. A caption buffer is maintained to accumulate recognized signs over time, enabling smooth sentence formation in continuous recognition scenarios.

Output:
Grammatically corrected text sentences.

6.4.2 Text-to-Speech Output

In the final module, the corrected text is converted into audible speech using a text-to-speech (TTS) engine. The TTS system generates natural-sounding audio corresponding to the recognized sign language content.

This module enables real-time verbal communication between signing users and non-signing individuals, significantly enhancing the practical usability of the system in assistive communication scenarios.

Output:
Spoken audio output.

7. Performance Metrics

There are many criteria that can be used to evaluate the performance of SLR model. For isolated SL, the commonly used evaluation metric is Accuracy rates. Precision, Recall and F-score are other commonly used metrics besides accuracy. For the proposed system, accuracy is the chosen criterion to make comparisons with other studies in the literature. The performance metrics we used in this study are given in (1), (2), (3), and (4).

$$\text{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN} \quad (1)$$

$$\text{Precision} = \frac{TP}{TP+FP} \quad (2)$$

$$\text{Recall} = \frac{TP}{TP+FN} \quad (3)$$

$$\text{FScore} = \frac{2TP}{2TP+FP+FN} \quad (4)$$

where TP is True Positive, TN is True Negative, FP is False Positive and FN is False Negative.

8. Conclusion

This paper presented an efficient real-time Indian Sign Language recognition framework based on multi-feature attention and dual-stream deep learning architectures. By combining EfficientNet-based spatial features with CNN–RNN temporal modeling and attention-driven feature fusion, the proposed system achieves effective recognition accuracy with low computational complexity. Future work will extend the framework toward continuous sign language translation and sentence-level understanding.

References

[1] E. Yenisari and S. Yavuz, "Deep Learning-Based Sign Language Recognition Using Efficient Multi-Feature Attention Mechanism," *IEEE Access*, vol. 13, pp. 126684-126699, 2025, doi: 10.1109/ACCESS.2025.3586096.

[2] A. Joshi et al., "iSign: A benchmark for Indian sign language processing," *arXiv preprint arXiv:2407.05404*, 2024.

[3] S. Pandey, S. Tahseen, R. Pathak, H. Parveen, and M. Maurya, "Real-time vision-based Indian sign language translation using deep learning techniques," *Int. J. Innov. Res. Comput. Sci. Technol. (IJIRCST)*, 2025.

[4] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, May 2015.

[5] M. Kumar and P. Bhatia, "A survey on vision-based sign language recognition using deep learning," *IEEE Access*, 2024.

[6] H. Li, X. Wang, and Y. Chen, "Multimodal sign language translation using vision and language models," *Neurocomputing*, Elsevier, 2024.

[7] Y. Zhou, Z. Liu, and J. Huang, "Transformer-based models for multimodal language translation," *ACM Trans. Multimedia Comput. Commun. Appl.*, 2024.