Received: 23 January 2025 | Revised: 13 May 2025 | Accepted: 7 June 2025 | Published online: 7 July 2025

RESEARCH ARTICLE

Journal of Computational and Cognitive Engineering
2025, Vol. 00(00) 1–12
DOI: 10.47852/bonviewJCCE52025288

BON VIEW PUBLISHING

An Efficient Attention Model with Critical Frames Identification for Sign Language Recognition (CRAM-SLR)

Renjith Sasidharan[1], Aneesh Varghese[2], Manazhy Rashmi[3] and Poorna S. Surendran[3,*]

[1]*Department of Computer Science and Engineering, Amrita Vishwa Vidyapeetham-Amritapuri, India*

[2]*Amazon Web Services, USA*

[3]*Department of Electronics and Communication Engineering, Amrita Vishwa Vidyapeetham-Amritapuri, India*

Abstract: Sign language recognition (SLR) plays a crucial role in enhancing communication accessibility for individuals who are deaf or hard of hearing. This paper introduces the convolutional recurrent attention model (CRAM), a novel deep learning framework specifically designed to improve recognition performance in low-resource sign languages such as Indian Sign Language (ISL) and Arabic Sign Language (ArSL). CRAM features a Critical Frames Identification algorithm that leverages the histogram of oriented gradients descriptor to extract the most informative key frames from sign videos, thereby reducing computational overhead while retaining essential gesture information. The model architecture combines convolutional layers to extract rich spatial features, bidirectional long short-term memory networks for effective temporal sequence modeling, and an attention mechanism to dynamically prioritize crucial frames. This integration enables CRAM to capture complex spatial-temporal dependencies inherent in sign gestures. Extensive experiments conducted on ISL and ArSL datasets validate the model's effectiveness, with CRAM achieving state-of-the-art accuracy, precision, and recall. The results highlight CRAM's potential in advancing robust and inclusive SLR solutions for underrepresented sign languages, promoting more effective gesture-based human-computer interaction.

Keywords: sign language, classification, hybrid CNN-BiLSTM, convolutional recurrent attention

1. Introduction

Sign language is a form of communication that is used as the first language for many in the hearing-impaired and hard-of-hearing communities, who use it to interact, convey their emotions and thoughts, and understand others in a primarily spoken language world. Although accessible sign language communication is important, many regions, especially those with limited resources, suffer considerable challenges in effectively deploying sign language recognition (SLR) technologies. Due to limited data availability, lack of technical infrastructure, and other cultural perceptions that deprioritize technological solutions to the deaf community, countries such as India and many more in the Arabian region continue to stay behind in SLR advancements as well [1]. Such a lack of proper channels to communicate denies the deaf in poor areas from education, jobs, or social engagement, further isolating them from society. Furthermore, since sign languages are so rich and artistically elaborate, they cannot be transposed to traditional communication forms such as lip-reading or written text, which is insufficient to express the meaning delivered by gestures, feelings, and nuances. This highlights an immediate need for accessible technology-based solutions specifically targeted toward sign language users in resource-constrained settings [2].

The field of artificial intelligence and machine learning [3] has seen rapid advancements, bringing the potential for more accurate and effective SLR systems. Deep learning techniques, in particular, have demonstrated the ability to improve gesture recognition by converting signs into comprehensible text or spoken language. While some studies, such as those referenced in Sasidharan et al. [4], have explored sign languages with abundant datasets, the choice of these languages for research often stems from other factors, such as their linguistic prominence, availability of structured annotations, or broader applicability, rather than solely the availability of data. For SLR to become truly inclusive, it is essential to address the needs of marginalized communities by focusing on the diversity and complexity of sign languages worldwide.

We introduce the Critical Frames Identification (CFI) algorithm and convolutional recurrent attention model (CRAM) specifically designed for handling low-resource languages in the SLR domain pretty effectively. The CFI algorithm utilizes the histogram of oriented gradients (HOG) technique for selecting seven crucial frames from sign language videos rather effectively. CFI reduces computational demands rather significantly and boosts efficiency notably without much loss of accuracy by isolating the most informative frames. CRAM effectively captures spatial

***Corresponding Author:** Poorna S. Surendran, Department of Electronics and Communication Engineering, Amrita Vishwa Vidyapeetham-Amritapuri, India. Email: poornass@am.amrita.edu

and temporal dynamics of sign language gestures with convolutional neural networks (CNNs) and bidirectional long short-term memory (BiLSTM) networks alongside an attention mechanism. Attention mechanism plays a crucial role in enabling the model to focus selectively on high-value frames within a sequence, improving interpretability significantly and boosting accuracy. In contrast to well-documented sign languages such as American Sign Language (ASL), Arabic Sign Language (ArSL) and Indian Sign Language (ISL) face significant challenges due to limited publicly available datasets and a lack of standardized linguistic annotation. For instance, datasets like Word Level American Sign Language for ASL comprise over 21,000 signs, whereas Indian Sign Language – Continuous Sign Language Translation and Recognition contains only 700 annotated videos, underscoring the disparity in resource availability [5]. Furthermore, socioeconomic factors in regions where ArSL and ISL are prevalent contribute to limited technological advancements and research efforts in these languages. This underrepresentation is also evident in bibliometric analyses, where studies on ArSL and ISL are significantly fewer compared to ASL [2]. Addressing these challenges is critical for developing inclusive SLR systems.

Our research focuses on inclusiveness and accessibility for hearing-impaired and hard-of-hearing individuals in underserved areas, with particular attention to low-resource Asian sign languages such as ISL and ArSL. We conduct rigorous experimentation and evaluation on these datasets to verify the model's efficiency and effectiveness in identifying and interpreting sign gestures. By targeting low-resource settings and diverse linguistic communities, this work aims to foster a more inclusive and accessible world where advanced technology serves as a bridge between sign language users and non-signers. Below, we summarize the main contributions of our proposed CRAM-SLR model as follows:

1) **Crucial frame identification using HOG descriptors:** We propose a lightweight Critical Frame Identification (CFI) algorithm based on HOG, which selects seven highly informative frames from each sign language video. This approach captures motion dynamics while significantly reducing computational load, enabling efficient recognition in real-time systems.
2) **Hybrid attention-based CNN-BiLSTM architecture:** The CRAM model integrates CNN for extracting spatial features from selected frames and BiLSTM for modeling temporal dependencies, enhanced by an attention mechanism that emphasizes discriminative gesture patterns.
3) **Adaptability to low-resource sign languages:** CRAM is specifically designed for resource-constrained scenarios and tested on two low-resource datasets—ISL and ArSL. It demonstrates competitive accuracy with significantly fewer frames and simpler architectures.
4) **Comprehensive evaluation with visual and quantitative analysis:** We provide an in-depth performance evaluation including ablation studies, visualization of attention maps, critical frame selections, and comparison with recent Transformer-based and skeleton-based methods, showing CRAM's superiority in efficiency and robustness.

The proposed CRAM model offers a novel and efficient solution for SLR by combining crucial frame selection with a hybrid deep learning approach. Designed for low-resource sign language communities, it bridges a critical gap by enabling accessible and inclusive communication technologies for underserved regions worldwide.

2. Related Literature

SLR has seen significant advancement over the years, with a growing number of methods exploring visual, sensor-based, and hybrid approaches to bridge communication gaps for the hearing impaired [6, 7]. These techniques often depend on a variety of modalities including RGB video, depth sensors, and wearable devices to process both manual and non-manual components of signs [8, 9].

In vision-based recognition, two primary approaches are commonly used: static and dynamic gesture recognition. Static approaches interpret individual frames showing isolated hand signs, while dynamic recognition captures continuous sign language motion, incorporating temporal dependencies and facial expressions. Sensor-based systems, on the other hand, utilize wearable gloves embedded with accelerometers and gyroscopes to accurately capture motion, orientation, and hand shapes. Depth-sensor-based recognition leverages devices like Microsoft Kinect or Intel RealSense to collect 3D spatial information, improving gesture understanding by adding a third dimension [10].

Traditional machine learning methods such as hidden Markov models and support vector machines have been effective in earlier systems. However, with the rise of deep learning, models based on CNNs and recurrent neural networks (RNNs) have become dominant due to their superior performance in visual and sequential data processing. These models allow for efficient spatial and temporal feature extraction, making them suitable for complex gesture interpretation. The effectiveness of each model depends heavily on the use case, available computational resources, and the requirement for real-time processing. Recent literature shows significant improvements when integrating non-manual features into recognition systems. For example, Alyami et al. [11] demonstrated that combining face and hand keypoints led to a 4% accuracy increase over using hand keypoints alone. Their approach employed three model architectures—temporal convolution networks, Transformer-based models, and LSTM networks—and showed that pose-based Transformer models outperformed others with 99.74% signer-dependent accuracy in the KArSL-100 dataset.

In ASL recognition, Mohsin et al. [12] explored transfer learning strategies using deep architectures like VGG16, ResNet50, MobileNetV2, and a custom CNN with InceptionV3. Their InceptionV3 model achieved the highest accuracy of 96%, while the others followed closely. This work highlights that pre-trained models fine-tuned on domain-specific sign language data can yield remarkable performance even in the presence of limited labeled datasets. A substantial contribution to the dataset development was made by Hasib et al. [13], who introduced the BdSL dataset comprising 29,490 annotated RGB images representing 49 Bangla alphabet signs. Smartphone cameras snapped pictures that formed a robust basis for machine learning models adept at deciphering Bengali Sign Language nuances fairly well. They contributed significantly toward the development of the Malaysian Word Sign Language dataset, featuring diverse hand signs from individuals having different skin tones. Trained on this dataset, a sequential CNN model attained quite impressively 98% accuracy. RGB and depth modalities were utilized by Poonia et al. [14] rather effectively inside CNN-based architectures for recognizing ISL gestures somewhat accurately. Dynamic gesture recognition via optical flow and skeleton-based models enhances classification performance significantly in Indonesian Sign Language research. Hybrid deep learning techniques are expanding rapidly into low-resource sign languages also. Graph convolutional networks (GCNs) were implemented by Khartheesvar in 2024 using Chinese Sign Language data to represent spatial-temporal relationships effectively. Korean Sign Language research has utilized attention-based models heavily, prioritizing meaningful gesture segments during recognition somewhat effectively lately [15].

Several recent works have advanced the field of signal and modulation classification using lightweight and attention-driven

models. Zheng et al. proposed an asymmetric trilinear attention network with noisy activation functions to improve robustness in automatic modulation classification under complex channel conditions [16]. In another effort, the authors developed a real-time constellation image classification framework leveraging the MobileViT architecture, which demonstrated efficient processing of wireless communication signals with low latency and computational cost [17]. Further extending their contributions, the MobileRaT framework introduced a lightweight radio transformer optimized for drone-based communication systems, achieving both high accuracy and model compactness suitable for edge deployment [18].

Despite these advancements, SLR still faces challenges such as variability in signing styles, limited annotated datasets, background interference, and difficulties in capturing dynamic gestures. Sign variations across individuals, lighting conditions, and complex backgrounds contribute to the complexity of real-time gesture recognition.

Our proposed CRAM addresses these challenges through a hybrid approach. For handling variability in sign styles, CRAM integrates CNNs with RNNs to extract both spatial and temporal features. The use of diverse datasets enhances the generalization and robustness of unseen signs. To minimize background noise, preprocessing techniques such as background subtraction and noise filtering are employed. Moreover, CRAM uses BiLSTM layers that allow for better temporal understanding of continuous gestures, critical for recognizing dynamic movements accurately.

Overall, CRAM effectively addresses heterogeneity, limited data, and environmental challenges. By combining sensor-based and vision-based modalities and leveraging deep learning, it demonstrates enhanced performance across multiple sign languages and use environments.

3. Problem Statement

The goal is to develop a robust and accurate system for recognizing sign language gestures to facilitate communication for individuals with hearing impairments.

Let X represent the input space, consisting of sign language videos, where each video $x_i \in X$ is composed of T frames. Each frame contains features representing spatial and temporal aspects of the gestures.

Let Y represent the output space of gesture classes, where $y_j \in Y$ denotes the class label corresponding to a gesture x_i. The goal is to learn a mapping function $f : X \rightarrow Y$ that predicts the correct class label y_j for an input x_i, such that:

$$f(x_i) = y_j, \quad \text{for } x_i \in X \text{ and } y_j \in Y \tag{1}$$

Given the challenges in low-resource languages such as ISL and ArSL:

1) The dataset X is limited in size, leading to insufficient training samples for traditional deep learning models.
2) Sign gestures exhibit complex spatial-temporal dependencies, making it difficult to extract meaningful features $\phi(x_i)$ from the input.

The problem can be further formalized as finding the optimal function f^* that minimizes a loss function L over the dataset (X, Y):

$$f^* = \arg \min_f \sum_{i=1}^{N} L(f(x_i), y_i) \tag{2}$$

where L measures the error between the predicted class $f(x_i)$ and the ground truth class y_i.

To address these challenges, this research incorporates:

1) **Critical frames identification (CFI):** Selecting a subset $x_i' \subset x_i$ of crucial frames that preserve temporal dynamics, reducing computational cost.
2) **Hybrid model design:** A CRAM that extracts spatial features $\phi_s(x_i')$ using a CNN, models temporal dependencies $\phi_t(x_i')$ using BiLSTM, and applies attention mechanisms to focus on salient gesture patterns.

The optimization problem is thus reformulated as:

$$f^* = \arg \min_f \sum_{i=1}^{N} L(f(\phi_s(\phi_t(x'_i))), y_i) \tag{3}$$

where x_i' represents the critical frames selected by the CFI algorithm. The model aims to achieve high accuracy while minimizing computational complexity, making it suitable for low-resource sign languages.

4. Proposed Model

We adopt the convolutional recurrent attention (CRA) model as we strive to design a system for SLR. This novel approach is created with the aim of capturing both spatial and temporal characteristics, which are inherent in sign language gestures, thus making it possible for accurate recognition to be achieved. The CRA model combines CNNs with recurrent ones to effectively process spatial information from input images as well as capture sequential frames' temporal dependencies using their respective strengths. The holistic nature of this strategy enables the model to differentiate between slight variations in hand movements, facial expressions, and body postures, thus necessary for the correct interpretation of sign language. Figure 1 is a block diagram that represents the proposed model. The suggested pipeline for SLR includes frame extraction, preprocessing, and CRA to provide a complete method of SLR based on temporal-spatial feature representation.

4.1. Frame extraction

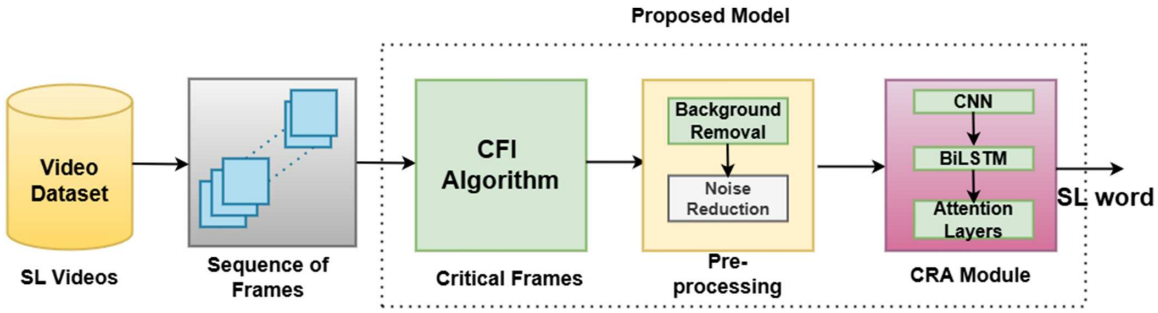
The initial part of an SLR system is frame extraction. Because sign language gestures are characterized by dynamic hand movements and configurations that change over time, it is important to take into account the temporal context for accurate recognition. To represent this aspect properly, video frames should be captured at regular intervals of around 50 milliseconds or so-called "extracted frames," which will serve as input sequences during following preprocess steps (Figure 1).

Temporally preserving details and the context of sign language gestures is critical—this is where correct frame extraction becomes invaluable. We take snapshots at fixed rates, thereby maintaining continuity through subsequent processing stages where communication using signs involves such things as dynamic hand movements, facial expressions, or body postures.

4.2. Critical Frames Identification (CFI)

In this stage, each frame f_i from the video sequence is processed through the CFI algorithm, designed to extract frames that reflect meaningful transitions in the gesture. Unlike fixed-threshold methods, our approach integrates gradient-based motion analysis with adaptive thresholding and local extrema detection to robustly capture the most informative frames.

Figure 1
Block diagram of proposed model



Each frame f_i is first resized to a standard dimension to normalize the spatial resolution:

$$f_i' = \text{resize}(f_i) \tag{4}$$

This preprocessing step ensures consistent computation of HOG descriptors across all frames, reducing sensitivity to resolution differences and enabling computational efficiency. Next, the HOG descriptor is calculated for each resized frame f_i' as:

$$H_i = \sum_{c \in \mathscr{C}} \text{hist}(\theta_c, w_c) \tag{5}$$

where each image is divided into cells c, $\theta_c = \arctan(I_y/I_x)$ is the gradient orientation in cell c, and $w_c = \sqrt{I_x^2 + I_y^2}$ is the gradient magnitude. The function hist$(\cdot)$ accumulates the gradient orientations weighted by magnitudes into a fixed-bin histogram. This representation captures shape and contour information crucial for distinguishing gesture changes. The algorithm then computes the motion difference between consecutive frames using the L2 norm of HOG descriptors:

$$\Delta_i = \| H_i - H_{i-1} \|_2 \tag{6}$$

The resulting sequence $\{\Delta_1, \Delta_2, \ldots, \Delta_{n-1}\}$ represents the motion energy between each pair of frames. An adaptive threshold T is then calculated using the statistical properties of motion differences:

$$T = \mu_D + \lambda \cdot \sigma_D \tag{7}$$

where μ_D and σ_D are the mean and standard deviation of the difference sequence D and λ is a scaling factor (e.g., 0.5–1.0) that controls sensitivity. This dynamic threshold adapts to each video's unique motion profile. To identify the most meaningful transitions, the algorithm detects local maxima and minima in the motion difference sequence. For each index i, a point is selected if it satisfies:

$$(\Delta_{i-1} < \Delta_i > \Delta_{i+1}) \quad \text{or} \quad (\Delta_{i-1} > \Delta_i < \Delta_{i+1}) \tag{8}$$

These local extrema represent turning points in gesture dynamics. If such a point also satisfies $\Delta_i > T$, the corresponding frame f_i is added to the set of critical frames C. The first and last frames are also added to C to preserve the starting and ending pose of the sign gesture:

$$C = \{f_0\} \cup C \cup \{f_n\} \tag{9}$$

The final output set C thus contains frames that capture salient gesture transitions, filtered through both structural motion analysis and adaptive selection criteria. This significantly reduces redundant frames while retaining the expressive components necessary for accurate recognition, enabling real-time performance and improved robustness in low-resource SLR applications.

Algorithm 1 Critical Frame Identification (CFI)

Input: Sequence of video frames $f_0, f_1, \ldots, f_n$
Output: Set of critical frames C
Initialize $C = \{f_0\}$, $D = [\,]$
for each i from 1 to n **do**
Resize frame $f_i' = \text{resize}(f_i)$
Compute gradients: $I_x = \partial f_i'/\partial x$, $I_y = \partial f_i'/\partial y$
Compute magnitude: $w = \sqrt{I_x^2 + I_y^2}$, orientation: $\theta = \arctan(I_y/I_x)$
Compute HOG descriptor $H_i = \sum_{c \in \mathscr{C}} \text{hist}(\theta_c, w_c)$
if $i > 1$ **then**
Compute difference $\Delta_i = \| H_i - H_{i-1} \|_2$
Append Δ_i to D
end if
end for
Compute adaptive threshold $T = \mu_D + \lambda \cdot \sigma_D$ ▷mean + scaled std-dev
for each i from 2 to $n-2$ **do**
if $\Delta_{i-1} < \Delta_i > \Delta_{i+1}$ **or** $\Delta_{i-1} > \Delta_i < \Delta_{i+1}$ **then**
if $\Delta_i > T$ **then**
Add f_i to C
end if
end if
end for
Add final frame f_n to C
return C

Figures 2 and 3 illustrate the application of the CFI algorithm for the SL word "Fall." Figure 2 shows all frames from the gesture, while Figure 3 presents the subset identified as critical frames. This selection is based on numerical differences computed between consecutive frames, and a frame is selected as ▷critical if the computed difference exceeds a predefined threshold. In our approach, we empirically set the threshold to T = 0.75, which effectively captures significant pose transitions. Figure 3 is annotated with both the threshold line and the computed values to clearly indicate where T > 0.75. This approach selectively captures key transitions, resulting in a more efficient and accurate SLR process by focusing on the frames most representative of each sign. The steps involved in the CFI algorithm are outlined in Algorithm 1.

Figure 2
Extracted frames of the SL word 'Fall'

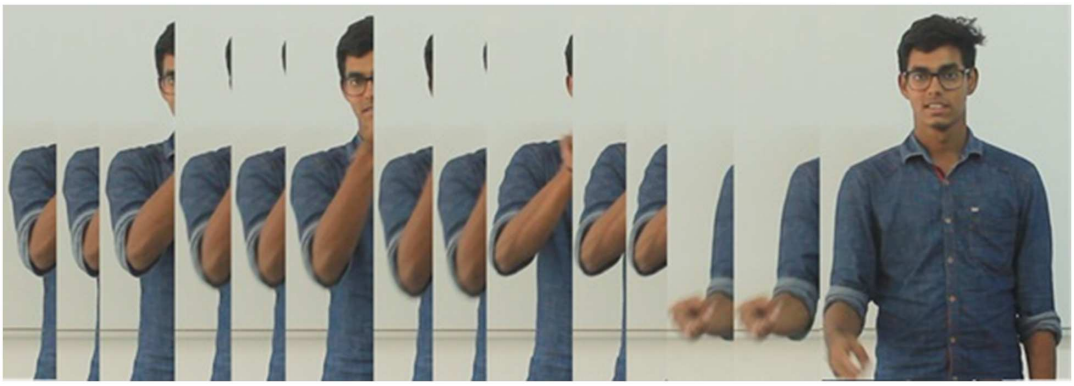


Figure 3
Critical frames identified for the SL word 'Fall'

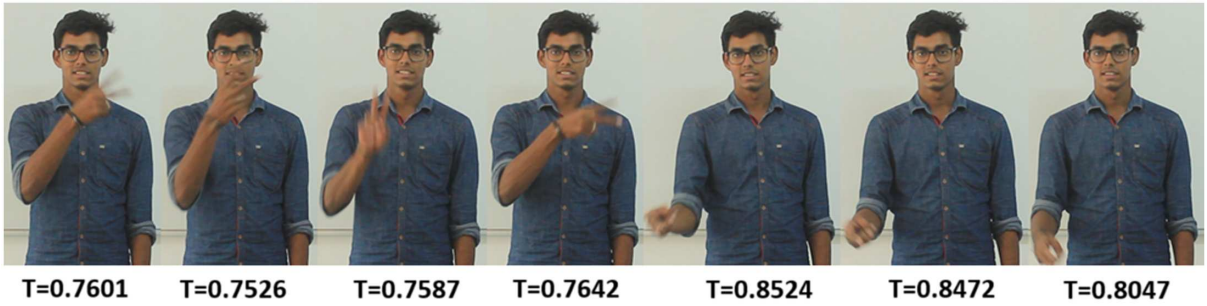
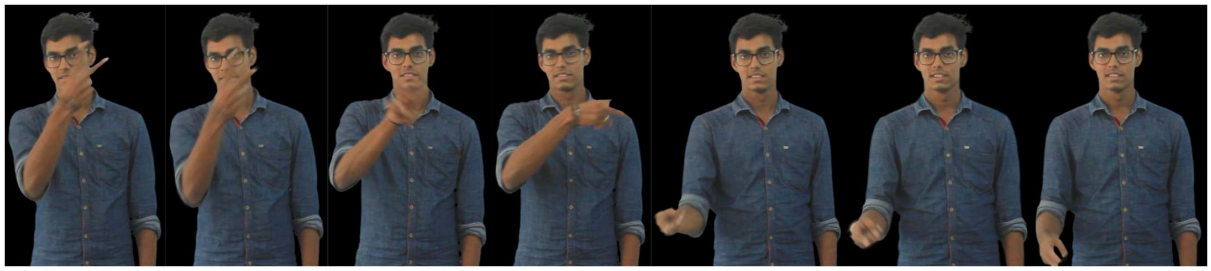


Figure 4
Background removed from the critical frames for SL word 'Fall'



4.3. Pre-processing

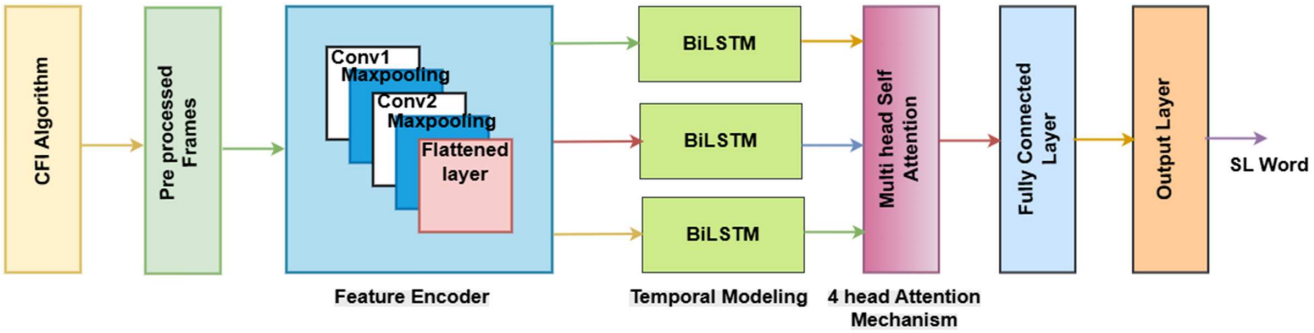
After videos have been broken down into crucial frames, they undergo preprocessing within the SLR system. This step aims at making input data more appropriate for efficient feature extraction by improving its quality and bringing it into a uniform format prior to being fed into the model. The typical operations performed during preprocessing may involve standardization of pixel values, scaling images to some common size, while dealing with noise/artifacts in datasets also forms part of them. Standardizing input data enables patterns learned from different examples by the model. Among other additional preprocessing steps in our approach are background subtraction and noise reduction, which were introduced to enhance model robustness when working under challenging environments. These actions help eliminate irrelevant information, thus allowing only relevant hand gestures to be considered by the model, hence increasing overall system accuracy.

Background subtraction works by separating the foreground (usually the signer's hands) from the background, thereby eliminating stationary parts of the video, hence exposing moving hand details. We used popular methods like frame differencing for accurate background subtraction as well as more sophisticated ones like Gaussian mixture models. Another technique employed for noise reduction in our method is temporal smoothing, which looks at temporal consistency across neighboring frames either to stabilize erratic motions or discard short-lived ripples; this can be done by taking temporal averages or applying moving average filters over a sequence of frames. By using these preprocessing techniques that make input data more suitable for feature extraction later, the SLR system's robustness and accuracy in real-world environments are improved. Figure 4 depicts images of the background removed from the criticalframes for the SL word 'Fall.'

4.4. Convolutional recurrent attention model (CRAM)

The proposed CRAM-SLR system uses a hybrid literature that combines CNN and BiLSTM with attention models. This design is

Figure 5
Proposed model architecture



aimed at exploiting natural spatial as well as temporal features found in sign language movements. The model's proposed architecture is shown in Figure 5.

4.4.1. CNN component

The model's first section is devoted to employing a CNN to extract spatial features. This part can extract hierarchical characteristics from the input photos with ease. The following layers make up the CNN component of the suggested model.

1) Convolutional layers: The model starts with a CNN for extracting features from input images. The convolutional layer applies a convolution operation on the input image using a set of learnable filters (kernels). Given an input image tensor I with shape (64, 64, 3) and a convolutional kernel K, the convolution operation is defined as:

$$(I*K)(x,y) = \sum_{i=1}^{m} \sum_{j=1}^{n} I(x-i, y-j) \cdot K(i,j) \qquad (10)$$

where (x, y) represents the spatial coordinates in the output feature map, (i, j) represents the spatial coordinates in the kernel, and m and n represent the dimensions of the kernel. Two convolutional layers with 32 and 64 filters of size (3, 3), respectively, are applied with rectified linear unit (ReLU) activation functions.

$$ReLU(x) = \max(0, x) \qquad (11)$$

That is, the first convolutional layer applies 32 filters denoted as W_1 of size (3, 3), resulting in feature maps with shape (62, 62, 32) after applying ReLU activation: $H_1 = \text{ReLU}(X*W_1 + b_1)$, where b_1 is the bias.

The second convolutional layer applies 64 filters of size (3, 3), resulting in feature maps H_2 with shape (29, 29, 64) after applying ReLU activation.

2) Pooling layers: Max pooling layers with a pool size (2, 2) are used to downsample the feature maps.

$$MaxPooling(x,y) = \max_{i,j \in \text{pool}} I(x+i, y+j) \qquad (12)$$

Downsampling feature maps using max pooling layers preserves important information while lowering spatial dimensions. The model's capacity to concentrate on pertinent features is improved by pooling. The input feature map is divided into tiny 2×2 blocks, and the largest value in each block is kept, while the other values are discarded. This downsampling reduces the size of the feature map without sacrificing significant information. The output form of the MaxPooling2D((2, 2)) operation is (31, 31, 32). The feature maps' spatial dimensions are decreased using downsampling. The 32 feature maps are now divided into (31,31)grids. The final max pooling layer with a pool size of (2, 2) downsamples the feature maps to P_2 with shape (14, 14, 64).

3) Flattening: The process of flattening in the CNN component transforms the 2D feature maps into a 1D vector. This step is crucial as it prepares the spatial features extracted by the convolutional and max pooling layers to be fed into the subsequent BiLSTM layers for sequence modeling. Mathematically, if we have feature maps of size $14 \times 14 \times 64$, flattening them results in a 1D vector of length $14 \times 14 \times 64 = 12544$. Each value from the 14×14 grid is lined up in a single row, resulting in a long list containing all the values from the feature maps.

Indeed, the flattened representation acts as the input sequence for the BiLSTM layers. By doing so, it enables the BiLSTM layers to capture temporal dependencies within the spatial features acquired by the CNN. This integration of spatial and temporal information is crucial for effectively recognizing intricate patterns inherent in sign language gestures.

4.4.2. Attention mechanism

The attention mechanism computes attention scores between the input sequence and itself, resulting in attention weights denoted as α, Let H_{BiLSTM} denote the output sequence from the BiLSTM layer. The attended encoding can be represented as

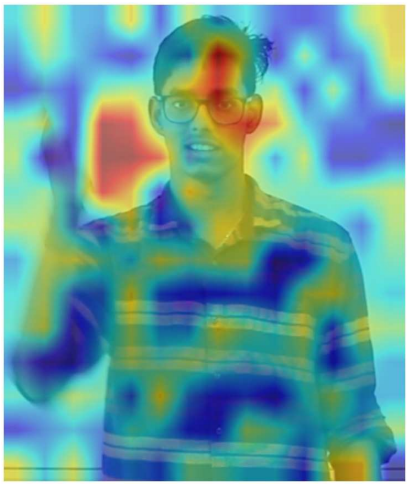
$$E = \alpha \cdot H_{\text{BiLSTM}} \qquad (13)$$

where $\cdot$ denotes element-wise multiplication. The attention mechanism can be mathematically expressed as

$$\alpha = \text{Attention}(H_{\text{BiLSTM}}) \qquad (14)$$

The Attention layer computes the attention scores between the input sequence and itself, enhancing the focus on relevant temporal features. The Permute layer transposes the dimensions of the attention scores to match the sequence length. Another Reshape layer is used to reshape the attended encoding to match the sequence length of the input. The Concatenate layer concatenates the original outcome of the BiLSTM layer with the attended encoding obtained from the attention mechanism. Figure 6 illustrates the attention heatmap of a selected critical frame, highlighting concentrated focus on the hand and facial regions. This visualization demonstrates that the model effectively attends to semantically relevant areas corresponding to key gesture components, thereby supporting the interpretability and functional reliability of the attention mechanism.

Figure 6
Attention heatmap of a critical frame



4.4.3. BiLSTM component

To address the temporal aspect of SL gestures, the model incorporates BiLSTM layers. Unlike traditional LSTMs, BiLSTMs process sequences in both forward and backward directions, capturing temporal dependencies effectively. The BiLSTM component has the following operations:

1) Temporal reshaping: The flattened output from the CNN component is reshaped to accommodate the temporal dimension. This reshaping prepares the data for input into the BiLSTM layers. It rearranges the values from the flattened list into a new shape, where each row contains 64 values. The output shape after reshaping is (196, 64), indicating that the data has been organized into 196 rows, each containing 64 values. This reshaping fits the data into the BiLSTM layer, designed to understand and process sequences.
2) Bidirectional LSTM layers: The model can represent dependencies in both past and future contexts because of these layers' bidirectional processing of sequences. Each LSTM cell stores and retrieves information over extended sequences, which is crucial for understanding sign language gestures' temporal nuances. At each step, the 64 memory cells of the first BiLSTM layer return sequences. Sequences are not returned by the second BiLSTM layer despite having 64 memory cells as well. For the first layer, the output shape is (196, 128). It means at each step, the model considers both the forward and backward sequences, resulting in a total of 128 values. For the second layer, the output shape is (128). This is obtained by concatenating the forward and backward hidden states, resulting in a condensed representation.
3) Dense layers for classification: The output from the concatenated BiLSTM layer is fed into fully connected dense layers for classification. Dense layers process the combined spatial-temporal features, further enhancing the model's ability to discern intricate patterns. Nonlinear activation functions introduce complexity, enabling the model to learn intricate relationships within the data. The Dense layer has 64 nodes and uses the ReLU activation function. The output shape is (64), indicating 64 values produced by this Dense layer. These values are a condensed representation of the information learned from the earlier layers.
4) Softmax output layer: The final layer employs softmax activation for multiclass classification, assigning probabilities to different sign language words. The Dense output layer with Softmax transforms the condensed information from previous layers into probabilities for each class. The class with the highest probability is considered the model's prediction for the input SL gesture.

5. Results and Discussion

The CRA model was developed for better SLR and was evaluated comprehensively using both the ISL dataset and the ArSL dataset. The evaluation of performance primarily focused on accuracy, which showed that the model can indeed recognize sign language gestures correctly most of the time. These findings indicate that the model could help promote communication inclusivity among people with hearing impairments.

5.1. Experiments

The CRA model underwent evaluation in a Jupyter Notebook environment utilizing Python 3.11. The evaluation took place on a personal computer with an AMD Ryzen 5000 series CPU, NVIDIA graphics, and 24 GB of RAM.

5.2. Dataset

5.2.1. ArSL dataset

ArSL database, called SignsWorld Atlas Database [19], records various body postures, gestures, and movements represented through different classes of manual as well as non-manual signs using digital cameras having high-quality video and image resolutions. Three scenarios of medical, roadside, and educational contexts are covered in the clips, totaling 178 motions. These include about 76 ArSL words represented by four skilled ArSL signers demonstrating them as movements. For analysis purposes, a subset of 25 ArSL words was selected from this database. These words are 'Alif,' 'Ba,' 'Ta,' 'One,' 'Two,' 'Hello,' 'Goodbye,' 'Mother,' 'Father,' 'Brother,' 'Sister,' 'Red,' 'Blue,' 'Monday,' 'Friday,' 'January,' 'July,' 'Morning,' 'Evening,' 'Happy,' 'Sad,' 'Thank you,' 'Anthem,' 'Cross,' and 'Mosque.'

5.2.2. ISL dataset

The ISL dataset [20] is a comprehensive and diverse resource for ISLR research. It consists of 4,287 videos spanning 263 classes across 15 semantic categories, such as adjectives, animals, clothes, colors, people, and places. The videos are captured at a resolution of 1920 × 1080 pixels, with an average duration of 2.57 seconds, approximately 56 frames per video, and a frame rate of 28 FPS. The dataset includes recordings from multiple signers to reflect signer variations and ensure robustness. It supports isolated word-level gesture recognition and is particularly valuable for training and evaluating models that utilize deep learning and computer vision techniques for improved accessibility and communication for the hearing-impaired community. For simplicity purposes, our research focused on 25 chosen ISL words. These are 'Hello,' 'Fall,' 'Good evening,' 'Happy,' 'Eat,' 'Red,' 'Dog,' 'Apple,' 'Mother,' 'Morning,' 'Tree,' 'Sun,' 'Car,' 'Doctor,' 'Book,' 'Medicine,' 'Computer,' 'Football,' 'Guitar,' 'Shirt,' 'Chair,' 'Train,' 'Teacher,' 'Birthday,' and 'high.'

5.3. Evaluation strategy

We employed evaluation criteria such as accuracy, precision, recall, F-measure, and confusion matrix.

The technique that we used is called 5-fold cross-validation, which is a common procedure in research on SLR and which is quite robust. Here, the dataset is divided into five parts known as folds or subsets. The model has undergone training and evaluation five times each, using different combinations of four folds for training and one for evaluation in every one of these cases. Such an iterative process allows the model to be tested across multiple subsets of data, where it can be seen how well it generalizes. A confusion matrix is employed to provide an overview of how accurately the model performed across all experiments or folds.

All frames were resized to a uniform resolution of 256 × 256 pixels. For HOG descriptor calculation, frames were converted to grayscale and normalized to reduce redundancy and standardize feature extraction. The model was trained for 50 epochs using the Adam optimizer with an initial learning rate of 0.001, categorical cross-entropy loss, and a batch size of 32. A learning rate decay schedule was applied every 10 epochs to enhance convergence stability.

To mitigate confounding variables and biases, several precautions were taken during experimentation. Both datasets used—ISL and ArSL—contain diverse signer identities, backgrounds, and recording conditions, which help improve the generalizability of the model. Signer-independent splits were ensured where possible to avoid overfitting to specific individuals. All video frames were resized to a fixed resolution prior to HOG descriptor extraction to standardize input features across samples.

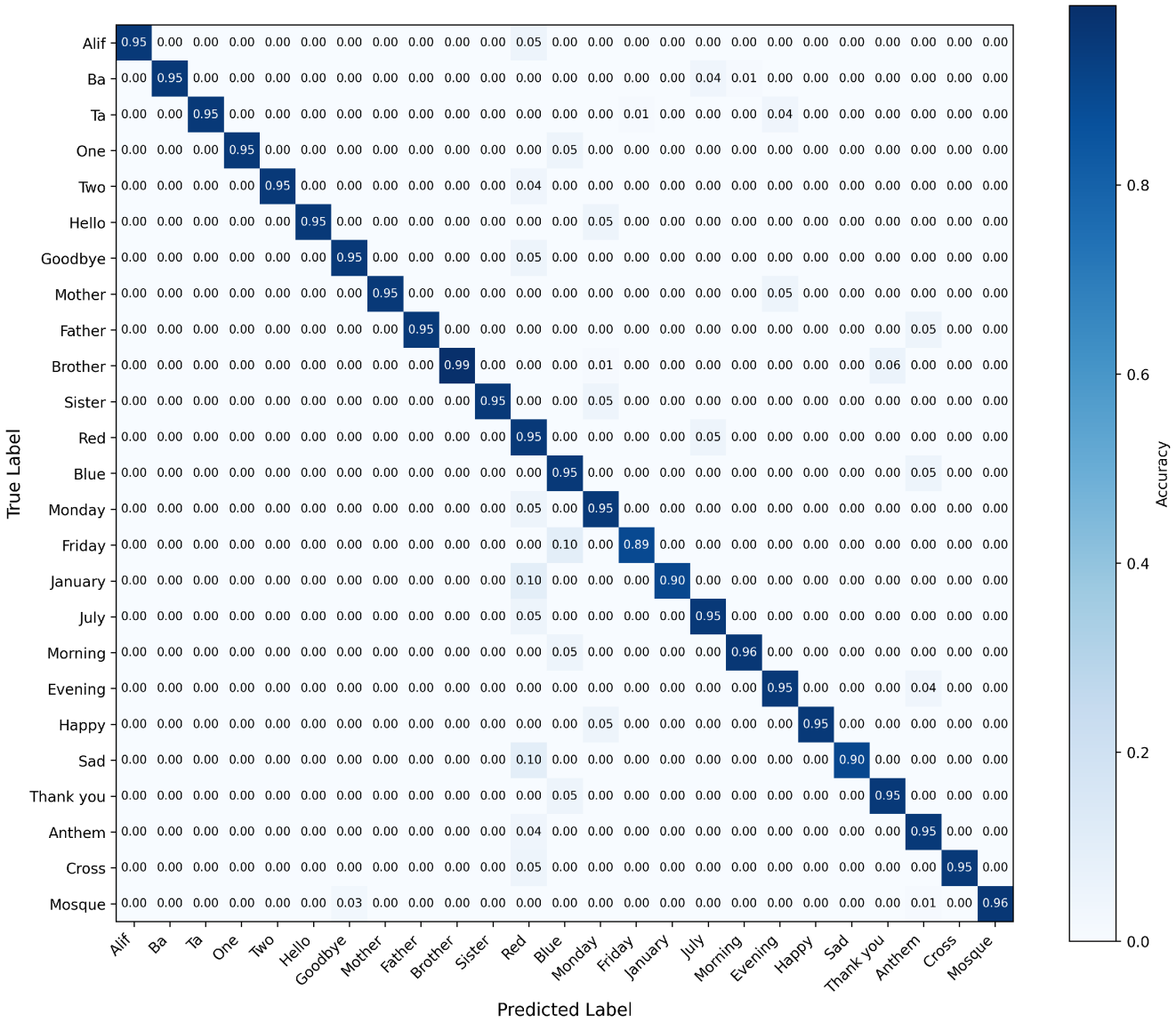
Randomized train-test splits were performed using fixed random seeds to ensure reproducibility and to eliminate selection bias. Additionally, class distributions were verified to be balanced during evaluation, and multiple metrics (accuracy, precision, recall, and $F1$-score) were used to ensure fairness across all classes. These steps collectively ensure that the results reflect the model's true performance rather than being skewed by data artifacts or procedural inconsistencies.

5.4. Results

5.4.1. ArSL dataset

Strong performance by the CRA model is observed from its evaluation results in the ArSL dataset. The experiment was

Figure 7
Confusion matrix of 5th fold of proposed model on ArSL dataset



conducted for 35 epochs. This model has an accuracy of 92.87%, illustrating its ability to predict signs correctly across sign classes. Its high precision of 92.04% explains how well it predicts positive predictions; its recall of 91.84% indicates that the model is good at recognizing actual instances of SL gestures, respectively, and an F-measure score of 91.01% shows this as a balanced performance, which means that the model achieves both high precision and recall. The aforementioned findings suggest that the Hybrid CNN-BiLSTM model demonstrates robustness in recognizing and classifying ArSL gestures, achieving a favorable balance between comprehensiveness and accuracy, as depicted in Figure 7, which illustrates the confusion matrix of the proposed architecture.

5.4.2. ISL dataset

The evaluation results in the ISL dataset with respect to the proposed CRA model indicate its commendable performance. The model achieved an accuracy of 93.46%; therefore, it shows a high level of correctness when predicting sign language gestures. Moreover, the statistics displayed in Table 1 such as precision at 93.14%, recall at 92.76%, and F-measure at 92.14% indicate accurate positive predictions and capture more positive instances. The above measures collectively emphasize the strength of this approach in relation to practical sign language translation and recognition tasks, thus making it an important contribution to the field. Figure 8 shows the confusion matrix of the proposed model on the ISL

Table 1
Performance metrics of the proposed model

Dataset	Accuracy	Precision	Recall	F-measure
ISL	93.46	93.14	92.76	92.14
ArSL	92.87	92.04	91.84	91.01

Figure 8
Confusion matrix of 5th fold of the proposed model on ISL dataset

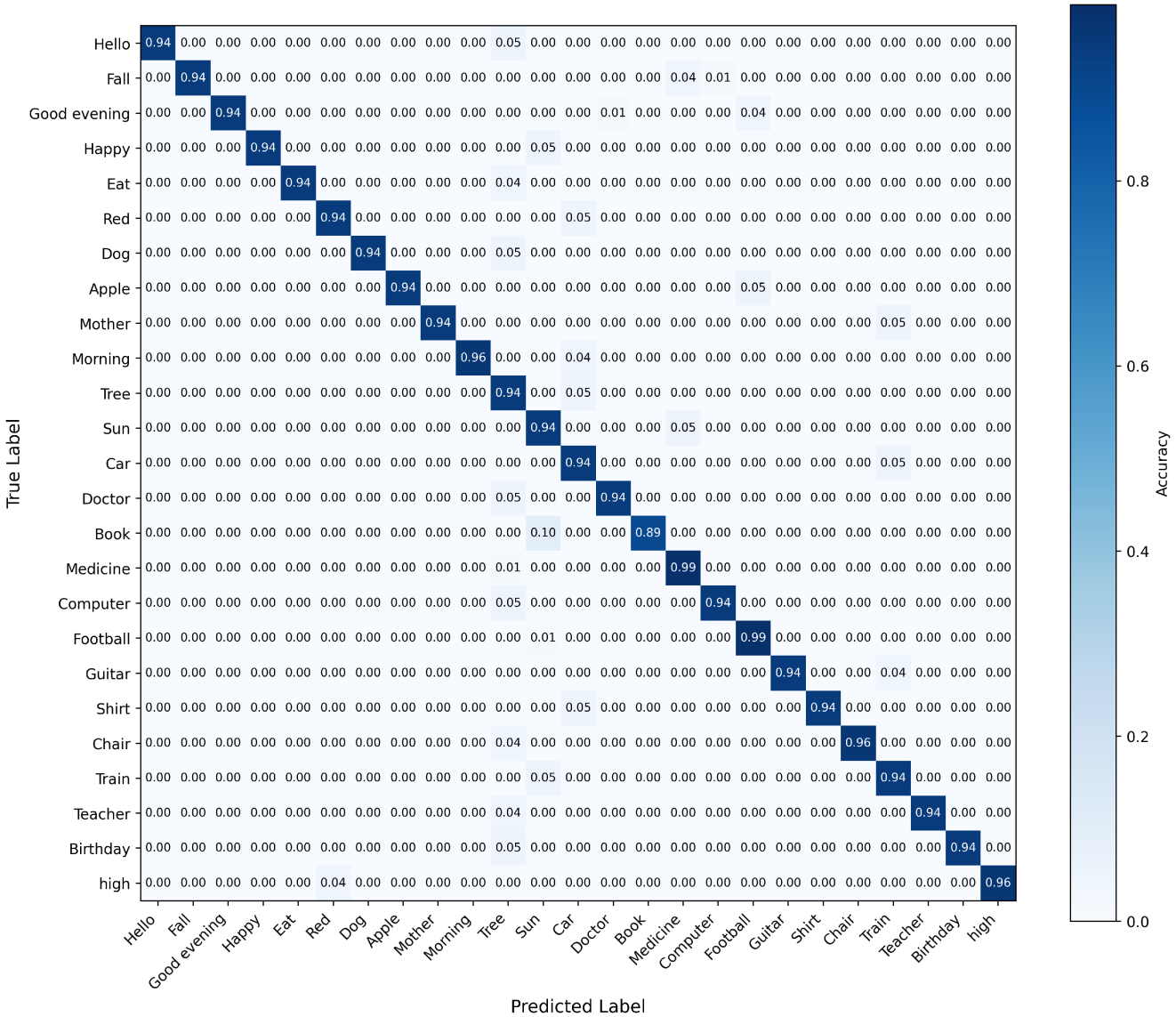
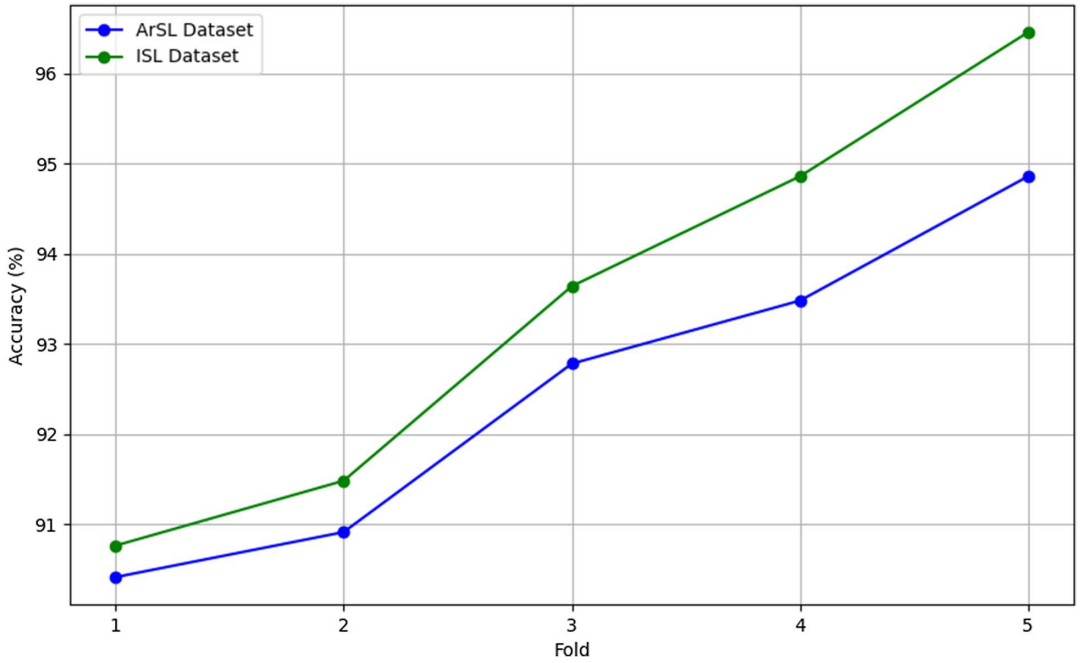


Figure 9
5-fold cross-validation accuracy of proposed model



dataset. Table 1 shows the performance evaluation for both datasets. Figure 9 shows the 5-fold cross-validation accuracy of the proposed model on both datasets.

5.5. Ablation study

The assessment juxtaposed three models—CNN, CNN+BiLSTM, and CNN+BiLSTM+Attention (CRAM)—using two sign language datasets: ISL and ArSL. The accuracy kept on increasing with each model variation in both datasets. For the ISL dataset, the CNN achieved 89.24% accuracy; this was improved to 91.47% by CNN+BiLSTM and further increased to 93.46% by CNN+BiLSTM+Attention (CRAM). Similarly, in the ArSL dataset, the accuracy grew from 87.59% for CNN to 89.51% for CNN+BiLSTM and then to 92.87% for CNN+BiLSTM+Attention (CRAM). These findings attest that incorporating temporal information as well as attention mechanisms can greatly increase sign recognition systems' precision over different databases. Table 2 depicts the results of an ablation study done on the proposed model on the two distinct datasets.

Table 2
Results of ablation study

Dataset	Model	Accuracy
ISL	CNN	89.24
ISL	CNN+BiLSTM	91.47
ISL	CNN+BiLSTM+Attention (CRAM)	93.46
ArSL	CNN	87.59
ArSL	CNN+BiLSTM	89.51
ArSL	CNN+BiLSTM+Attention (CRAM)	92.87

5.6. Computational complexity analysis

5.6.1. Computational complexity of the CFI algorithm

The computational complexity of the CFI algorithm can be analyzed in terms of its main components:

1) **Frame resizing:** Each of the N video frames is resized to a fixed dimension. This operation is linear in the number of pixels, assumed constant due to fixed resolution. Therefore, the complexity is $O(N)$.
2) **HOG descriptor computation:** For each frame, the HOG is computed. Assuming a fixed image size and fixed number of cells and bins, the computation per frame is constant. Hence, the total complexity across all frames is $O(N)$.
3) **Gradient difference computation:** The algorithm calculates motion differences Δ_i between consecutive frame descriptors using the Euclidean distance. This is computed $N-1$ times with a fixed-length feature vector, resulting in $O(N)$ complexity.
4) **Adaptive thresholding:** The mean μ_D and standard deviation σ_D of the difference sequence are computed once over $N-1$ values. This step is also $O(N)$.
5) **Local maxima/minima detection and frame selection:** This step involves a single pass through the Δ sequence, checking for extrema and comparing with the threshold T. This operation is linear in N, that is, $O(N)$.
6) **Set construction:** The final set C of critical frames is constructed by adding selected frames, which takes $O(K)$, where $K \leq N$.
7) **Overall complexity:** Since all major steps operate linearly with respect to the number of frames N, the total computational complexity of the CFI algorithm is:

$$O(N) \tag{13}$$

Table 3
Comparison with other SOTA models

Model	ISL accuracy (%)	ArSL accuracy (%)
LiST [14]	92.50	–
CNN-LSTM [20]	89.60	–
HCBSLR [21]	87.67	–
DCDW+LSTM [22]	91.30	–
CNN+LSTM [23]	–	88.00
CNN+LSTM [24]	–	90.48
CNN+Skeleton Features [25]	–	89.62
BiLSTM (two stage) [26]	–	92.60
Proposed model (CRAM)	**93.46**	**92.87**

This linear-time complexity makes the CFI algorithm highly suitable for real-time and low-resource SLR systems.

5.7. State-of-the-art model comparison

The SL recognition domain is a partly untouched landscape of SLR. In the past, most research on SLR has focused on recognizing simple words and numeral prints. There have been a few studies in ISL word recognition within the last decade that are mainly concentrated on sensor-based methods and dynamic gesturing approaches. The hybrid convolutional-bidirectional long short-term memory (HCBSLR) system undergoes a thorough 5-fold cross-validation process, which produces an astonishing average accuracy of 89.67% [20–26]. It is important to note that the proposed model has significant advantages in terms of training time compared to established models. Therefore, it can be concluded that the proposed system for ISL and ArSL word recognition outperforms its competitors in terms of accuracy and efficiency. The comparative analysis of various existing models against the proposed model within the same experimental setup is detailed in Table 3, providing a comprehensive overview of their respective performances.

6. Conclusion and Future Work

In this work, we presented CRAM, a lightweight and computationally efficient attention-based model for SLR. The proposed framework introduces a novel CFI algorithm that leverages HOG descriptors and an adaptive thresholding mechanism to extract the most informative frames, significantly reducing redundancy while preserving key spatiotemporal dynamics. CRAM combines spatial feature extraction using CNNs with temporal modeling through BiLSTM, enhanced by an attention mechanism that helps the model focus on critical gesture segments. The model demonstrates strong performance on two low-resource sign language datasets—ISL and ArSL—while maintaining low computational overhead, making it suitable for deployment in resource-constrained environments.

We plan to explore multimodal fusion by incorporating semantic signals like gloss-level text annotations or subtitles, allowing the model to resolve ambiguities between similar gestures through a multimodal attention mechanism. To enhance robustness to viewing angle changes and reduce the need for labeled data, we aim to conduct self-supervised pre-training using multi-view sign language videos. We will also implement spatiotemporal decoupling by separating spatial and temporal feature extraction paths, which will be recombined via a dynamic fusion layer for better gesture modeling. Additionally, we propose leveraging transfer learning with pre-trained backbones from large-scale sign language datasets and fine-tuning them for the target domain. Finally, we will introduce GCNs to model the grammatical structure of sign language, capturing topological relationships between key points or frames and improving the expression of complex gestures.

Acknowledgments

The authors express their heartfelt gratitude to Mata Amritanandamayi Devi for her divine blessings and spiritual encouragement, which inspired and gave them the strength to carry out this research work. The views and opinions expressed in this paper are solely those of the authors and do not represent the official position or endorsement of Amazon or its affiliates.

Ethical Statement

This study does not contain any studies with human or animal subjects performed by any of the authors.

Conflicts of Interest

The authors declare that they have no conflicts of interest to this work.

Data Availability Statement

Data are available on request from the corresponding author upon reasonable request.

Author Contribution Statement

Renjith Sasidharan: Conceptualization, Methodology, Software, Validation, Formal analysis, Investigation, Resources, Data curation, Writing – original draft. **Aneesh Varghese:** Software, Resources, Visualization. **Manazhy Rashmi:** Writing – review & editing, Supervision, Project administration. **Poorna S. Surendran:** Writing – review & editing, Project administration.

References

[1] Sasidharan, R., Sumi Suresh, M. S., & Rashmi, M. (2025). An effective skeleton-based approach for multilingual sign language recognition. *Engineering Applications of Artificial Intelligence*, *143*, 109995. https://doi.org/10.1016/j.engappai.2024.109995

[2] Sasidharan, R., & Manazhy, R. (2024). Sign language: A systematic review on classification and recognition. *Multimedia Tools and Applications*, *83*(31), 77077–77127. https://doi.org/10.1007/s11042-024-18583-4

[3] Paul, R., & Das, K. N. (2024). Trends of optimization algorithms from supervised learning perspective. *Journal of Computational and Cognitive Engineering*, *3*(4), 447–461. https://doi.org/10.47852/bonviewJCCE32021049

[4] Sasidharan, R., Rashmi, M., & Sumi Suresh, M. S. (2024). Sign language recognition by using spatio-temporal features. *Procedia Computer Science*, *233*, 353–362. https://doi.org/10.1016/j.procs.2024.03.225

[5] Elakkiya, R., & Natarajan, B. (2021). *ISL-CSLTR: Indian sign language dataset for continuous sign language translation and recognition* (Version 1) [Data set]. Mendeley Data. https://doi.org/10.17632/kcmpdxky7p.1

[6] Abdullahi, S. B., Chamnongthai, K., Bolon-Canedo, V., & Cancela, B. (2024). Spatial–temporal feature-based End-to-end Fourier network for 3D sign language recognition. *Expert Systems with Applications*, *248*, 123258. https://doi.org/10.1016/j.eswa.2024.123258

[7] Lieberman, A. M., Mitchiner, J., & Pontecorvo, E. (2024). Hearing parents learning American sign language with their deaf children: A mixed-methods survey. *Applied Linguistics Review*, *15*(1), 309–333. https://doi.org/10.1080/01434632.2024.2359634

[8] Uthus, D., Tanzer, G., & Georg, M. (2023). YouTube-ASL: A large-scale, open-domain American sign language-English parallel corpus. In *Proceedings of the 37th International Conference on Neural Information Processing Systems,* 1264.

[9] Berger, L., Pyers, J., Lieberman, A., & Caselli, N. (2024). Parent American sign language skills correlate with child–but not toddler–ASL vocabulary size. *Language Acquisition*, *31*(2), 85–99. https://doi.org/10.1080/10489223.2023.2178312

[10] Almubayei, D. S. (2024). Sign language choice and policy among the signing community in Kuwait. *Digest of Middle East Studies*, *33*(2), 166–183.

[11] Alyami, S., Luqman, H., & Hammoudeh, M. (2024). Isolated Arabic sign language recognition using a transformer-based model and landmark keypoints. *ACM Transactions on Asian and Low-Resource Language Information Processing*, *23*(1), 3. https://doi.org/10.1145/3584984

[12] Mohsin, S., Salim, B. W., Mohamedsaeed, A. K., Ibrahim, B. F., & Zeebaree, S. R. M. (2024). American sign language recognition based on transfer learning algorithms. *International Journal of Intelligent Systems and Applications in Engineering*, *12*(5s), 390–399.

[13] Hasib, A., Eva, J. F., Khan, S. S., Nipa Khatun, M., Haque, A., & Shahrin, N. (2023). BDSL 49: A comprehensive dataset of Bangla sign language. *Data in Brief*, *49*, 109329. https://doi.org/10.1016/j.dib.2023.109329

[14] Poonia, R. C. (2023). LiST: A lightweight framework for continuous Indian sign language translation. *Information*, *14*(2), 79. https://doi.org/10.3390/info14020079

[15] Du, Y., Li, P., Cheng, L., Zhang, X., Li, M., & Li, F. (2023). Attention-based 3D convolutional recurrent neural network model for multimodal emotion recognition. *Frontiers in Neuroscience*, *17*, 1330077. https://doi.org/10.3389/fnins.2023.1330077.

[16] Zheng, Q., Tian, X., Yu, Z., Yang, M., Elhanashi, A., & Saponara, S. (2025). Robust automatic modulation classification using asymmetric trilinear attention net with noisy activation function. *Engineering Applications of Artificial Intelligence*, *141*, 109861. https://doi.org/10.1016/j.engappai.2024.109861

[17] Zheng, Q., Saponara, S., Tian, X., Yu, Z., Elhanashi, A., & Yu, R. (2024). A real-time constellation image classification method of wireless communication signals based on the lightweight network MobileViT. *Cognitive Neurodynamics*, *18*(2), 659–671. https://doi.org/10.1007/s11571-023-10015-7

[18] Zheng, Q., Tian, X., Yu, Z., Ding, Y., Elhanashi, A., Saponara, S., & Kpalma, K. (2023). MobileRaT: A lightweight radio transformer method for automatic modulation classification in drone communication systems. *Drones*, *7*(10), 596. https://doi.org/10.3390/drones7100596

[19] Shohieb, S. M., Elminir, H. K., & Riad, A. M. (2015). SignsWorld Atlas; a benchmark Arabic Sign Language database. *Journal of King Saud University-Computer and Information Sciences*, *27*(1), 68–76. https://doi.org/10.1016/j.jksuci.2014.03.011

[20] Sridhar, A., Ganesan, R. G., Kumar, P., & Khapra, M. (2020). INCLUDE: A large scale dataset for Indian sign language recognition. In *Proceedings of the 28th ACM International Conference on Multimedia*, 1366–1375. https://doi.org/10.1145/3394171.3413528

[21] Das, S., Biswas, S. K., & Purkayastha, B. (2023). A deep sign language recognition system for Indian sign language. *Neural Computing and Applications*, *35*(2), 1469–1481. https://doi.org/10.1007/s00521-022-07840-y

[22] Dey, A., & Biswas, S. (2024). Gesture recognition for ISL-question signs in videos using DCDW-LSTM attention. In *2023 IEEE 5th PhD Colloquium on Emerging Domain Innovation and Technology for Society*, 1–2. https://doi.org/10.1109/PhDEDITS60087.2023.10373750

[23] Elons, A. S., Ahmed, M., Shedid, H., & Tolba, M. F. (2014). Arabic sign language recognition using leap motion sensor. In *2014 9th International Conference on Computer Engineering & Systems*, 368–373. https://doi.org/10.1109/ICCES.2014.7030987

[24] Suliman, W., Deriche, M., Luqman, H., & Mohandes, M. (2021). Arabic sign language recognition using deep machine learning. In *2021 4th International Symposium on Advanced Electrical and Communication Technologies*, 1–4. https://doi.org/10.1109/ISAECT53699.2021.9668405

[25] Bencherif, M. A., Algabri, M., Mekhtiche, M. A., Faisal, M., Alsulaiman, M., & Mathkour, H. (2021). Arabic sign language recognition system using 2D hands and body skeleton data. *IEEE Access*, *9*, 59612–59627. https://doi.org/10.1109/ACCESS.2021.3069714

[26] Shanableh, T. (2023). Two-stage deep learning solution for continuous Arabic sign language recognition using word count prediction and motion images. *IEEE Access*, *11*, 126823–126833. https://doi.org/10.1109/ACCESS.2023.3332250

How to Cite: Sasidharan, R., Varghese, A., Rashmi, M., & Surendran, P. S. (2025). An Efficient Attention Model with Critical Frames Identification for Sign Language Recognition (CRAM-SLR). *Journal of Computational and Cognitive Engineering*. https://doi.org/10.47852/bonviewJCCE52025288