SvedbergOpen
DISSEMINATION OF KNOWLEDGE
International Journal of Artificial Intelligence and Machine Learning
Publisher's Home Page: https://www.svedbergopen.com/
Research Paper Open Access

Leveraging Sparse Motion Learning And Hybrid Feature Fusion For Accurate Sign Language Recognition

Kiran C. Kulkarni[1], Manoj A. Wakchaure[2]

[1]Research Scholar Dept. of Computer Engineering, MET's Institute of Engineering, Nashik Savitribai Phule Pune University, Pune. , Email: kiranckulkarni@gmail.com
[2]Research Guide Dept. of Computer Engineering, MET's Institute of Engineering, Nashik Associate Professor, Amrutvahini College of Engineering Sangamner Savitribai Phule Pune University, Pune., Email: manoj.wakchaure@avcoe.org

Abstract
The proposed Sparse Motion Sequence Extraction Network (SMSE-Net) provides a sophisticated framework for fast and accurate hand sign language identification. Unlike typical deep learning models, which analyse dense frame sequences, SMSE-Net employs a sparse motion extraction approach to identify and prioritise essential gesture transitions while removing unnecessary frames. The architecture uses a Sparse Layer to extract spatial features and a Hybrid Layer to learn temporal motion, both of which are merged with a Siamese similarity model for exact classification. When compared to traditional models such as CNN, LSTM, and CNN+LSTM, experimental assessments show that SMSE-Net outperforms them all in terms of accuracy, precision, recall, and F1-score. The suggested technique has an average accuracy of 95.4% and a recall of 99%, demonstrating its robustness, generalisability, and applicability for real-time sign recognition applications.

Keywords
Sign Language Recognition; Sparse Motion Sequence Extraction; Deep Learning; Spatio-Temporal Features; Siamese Network; Gesture Classification; Hybrid Feature Layer.

I. Introduction

Sign language is the principal form of communication for the deaf and hard-of-hearing people, communicating meaning using hand movements, body posture, and facial expressions. The goal of automated sign language detection and identification is to help hearing and non-hearing people communicate more effectively. Recent advances in computer vision and deep learning have transformed the creation of intelligent systems capable of recognising sign movements from photos or movies [1]. Sign language detection entails determining if a person is making a sign and, if so, segmenting the appropriate frames for recognition or translation. This procedure often comprises hand and face recognition, background removal, motion tracking, and feature extraction from keypoints or skeletons. Deep neural networks, such as Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs), have shown greater ability in capturing both spatial and temporal dynamics in gestures [2]. Recent research favours vision-based techniques over classic glove or sensor-based systems because of their ease and cost-effectiveness. However, difficulties like as signer reliance, changeable illumination, background clutter, and continuous signing continue to impede successful real-world implementation [3]. Emerging architectures, such as attention-based and spatiotemporal models, have showed promise in addressing these challenges by learning discriminative gesture representations [4]. Sign language identification has several applications, including assistive technology, automated translation systems, and human-computer interface tools. To improve diversity and accessibility, future developments include real-time, mobile-friendly frameworks with multilingual sign language support and context-aware translation [5].
Machine Learning (ML) has made tremendous advances in sign language identification; yet, numerous persisting difficulties limit its accuracy, robustness, and real-world application. Signer variability is a serious concern, since variances in individual signing techniques, hand shapes, speeds, and orientations limit trained models'

generalisation capabilities [6]. Because of the scarcity of signer-independent datasets, ML models often perform well on training datasets but fail to adapt when exposed to new users or unknown signers [7].

Another major issue is temporal segmentation – sign language motions occur continually without distinct boundaries, making it difficult for ML systems to determine the beginning and finish of each sign [8]. This difficulty is exacerbated when many indications are merged in real-time communication. Another problem is multimodal feature integration; most machine learning-based systems focus largely on hand motion while ignoring significant non-manual signals such as facial expressions, lip movements, and body posture, which contain grammatical and contextual meaning [9].

Furthermore, data scarcity and imbalance create considerable impediments. Most available datasets are tiny, orientated toward certain sign languages (e.g., ASL or ISL), and lack variation in backdrop, lighting, and camera perspectives. These restrictions cause overfitting and hinder cross-domain adaptation. Furthermore, ethical and linguistic problems, such as a lack of Deaf community engagement and biased data labelling, have an influence on fairness and inclusion in machine learning-based sign language identification [10]. Addressing these issues requires big, diversified, and linguistically enriched datasets, as well as strong ML architectures that integrate visual and contextual comprehension for signer-independent and real-time applications.

II. Literature Review

Desai et al. [11] introduced ASL Citizen, a large, community-sourced isolated SLR dataset with "dictionary-retrieval" formulation that enhances resilience in real-world signing circumstances. Sarhan et al. [12] conducted a thorough study of isolated SLRs, mapping input modalities, model families, and evaluation gaps that remain despite deep learning breakthroughs. Fan et al. [13] suggested a continuous SLR pipeline that reframes video as a 1-D variable-length coded sequence after object identification, resulting in better CSLR performance on test sets. Zuo et al. [14] proposed the first online CSLR system, which uses a sliding window over sign clips learned from a curated vocabulary to achieve low-latency recognition. Attia et al. [15] developed tension-based efficient deep models that minimise complexity while preserving accuracy, with the goal of deploying SLR in resource-limited environments. Alyami et al. [16] examined 25 years of CSLR data, identifying segmentation, coarticulation, and signer independence as the primary bottlenecks for real-world systems. Srivastava et al. [17] developed an ISL continuous SLR system based on MediaPipe-derived landmarks and LSTMs, claiming real-time operation with high accuracy. Das et al. [18] introduced SASLRM, a vision-based expert system for commonly used ISL phrases in crises that prioritises practical vocabulary and deployment. Baihan et al. [19] suggested a modified deep learning strategy that uses a hybrid optimisation scheme to fine-tune network parameters and improve recognition efficiency. Rastgoo et al. [20] proposed a comprehensive deep-learning approach on SLR/SLP, synthesising trends and obstacles that influence recognition model construction. The TechScience study (Wang et al.) [21] offered a revised taxonomy of deep SLR techniques and datasets, indicating prospects for multimodal fusion and self-supervision.Ruiz et al. [22] proposed a word-level SLR technique that employs tiny handmade features to achieve comparable accuracy while minimising computing expense. Jin et al. [23] presented a large Chinese National Sign dataset (6,707 signs and 134k films) to improve vocabulary coverage for contemporary SLR standards. The IJERT 2024 work (Kumbhar et al.) [24] offered a deep-learning SLR pipeline intended at sign-to-text translation for accessibility, proving viability on limited datasets. ACM 2024 research (Hossain et al.) [25] proposed a web-app for isolated SLR based on Google ISLR competition data, emphasising end-to-end deployability.

Machine learning (ML) and deep learning (DL) approaches have substantially improved sign language detection and identification in recent years, but there are still some unfilled gaps that limit their accuracy, scalability, and real-time application. Traditional ML-based techniques depend heavily on handmade features like Histogram of Orientated Gradients (HOG), Histogram of Optical Flow (HOF), and skin-color segmentation. These characteristics are very sensitive to illumination fluctuations, complicated backdrops, and occlusions, rendering them ineffective for dynamic and realistic signing situations. Furthermore, traditional models such as Hidden Markov Models (HMM), Dynamic Time Warping (DTW), and Conditional Random Fields (CRF) fail to accurately represent temporal relationships in continuous signing sequences. They often rely on specified boundaries between signals and suffer with coarticulation effects (when successive gestures overlap), resulting in low segmentation and identification accuracy.

Deep learning frameworks such as CNNs, RNNs, LSTMs, and Transformers have solved some of these concerns by automatically learning hierarchical spatial-temporal characteristics, but they still have significant duplication and inefficiency. Most contemporary deep learning models handle dense frame sequences that include a considerable quantity of irrelevant or duplicated visual information. This not only raises processing costs, but it also dilutes

essential motion cues required for successful gesture recognition. Another problem is motion saliency and sparsity: existing models regard all frames identically, whereas only a portion of frames in a signing sequence have significant transitions or "stroke" movements that identify a sign. Failure to pay attention to these sparse but discriminative frames results in overfitting and reduced generalisation to unseen signers or settings.

Furthermore, the integration of manual (hand) and non-manual (facial, head, and body) components is still a serious difficulty. Many deep learning systems combine these multimodal cues too soon or too late, failing to emphasise their asynchronous and sparse character. As a consequence, systems fail to capture the contextual and grammatical aspects of sign language. The lack of big, signer-independent datasets, combined with inadequate temporal annotations, exacerbates these constraints, limiting real-world adaptation and cross-linguistic scalability. To bridge these gaps, the Sparse Motion Sequence Extraction Network (SMSE-Net) has emerged as a potential option. SMSE-Net generates sparse motion representations from video sequences by emphasising critical motion transitions while removing unnecessary or stagnant frames. This network uses spatiotemporal attention mechanisms and motion saliency mapping to concentrate exclusively on important gestures, improving interpretability while minimising computational complexity. SMSE-Net enables robust, efficient, and signer-independent sign language identification by integrating the capabilities of deep feature learning, motion sparsity, and attention-based optimisation, addressing the main shortcomings of both conventional ML and current DL models.

III. Methodology

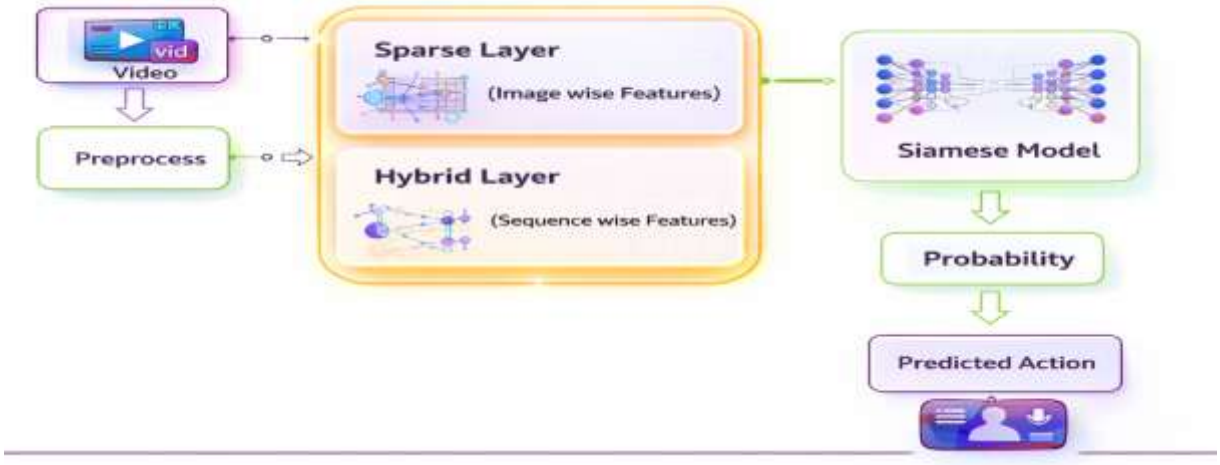
Sign language is an important means of communication for the hearing and speech challenged, enabling them to convey their ideas and feelings via hand gestures, facial expressions, and body movements. Traditional gesture recognition algorithms often fail to comprehend complicated sign sequences owing to overlapping movements, changing temporal patterns, and ambient variables. To address these issues, the Sparse Motion Sequence Extraction Network (SMSE-Net) has developed as a cutting-edge deep learning architecture for fast and accurate sign language recognition.

SMSE-Net extracts sparse yet discriminative motion features from continuous video frames, minimising superfluous temporal information while retaining important gesture signals. The network uses spatiotemporal attention processes and motion saliency detection to identify essential motions that constitute each symbol. This method greatly improves recognition accuracy and computing efficiency.

The model uses convolutional and recurrent layers to capture both spatial gesture structure and temporal motion progression. Furthermore, SMSE-Net can manage fluctuations in pace, illumination, and backdrop, making it suitable for use in real-world scenarios. Its sparse extraction method also permits speedier processing, which is critical for real-time communication systems and mobile platforms.

Overall, SMSE-Net bridges the gap between visual comprehension and linguistic translation, laying the groundwork for intelligent sign-to-text or sign-to-speech conversion systems. Its design supports inclusive technology by boosting accessibility and digital engagement for the deaf and mute communities.

Figure 1: Proposed Architecture



The architecture depicted above depicts the workflow of the Sparse Motion Sequence Extraction Network (SMSE-Net), a revolutionary framework for hand sign language identification that effectively captures both spatial and temporal dynamics of movements in video clips.

1. Input and Preprocessing

The input video starts the process. It has a series of hand motions that stand for sign language. Frame extraction, background removal, hand area recognition, and normalisation are all important steps in the preprocessing stage. These processes make sure that the input data is clean, devoid of noise, and ready for further feature extraction. The preprocessing also helps to find important motion areas by just looking at visual aspects that matter, such the shape and movement of the hands.

2. Sparse Layer (Image-wise Feature Extraction)

The Sparse Layer takes out spatial information from each frame one at a time. This layer utilises sparse feature extraction instead of treating every pixel the same way as standard convolutional networks do. It only concentrates on places with a lot of motion or saliency, such fingers, hand borders, and wrist movements. By doing this, the model gets rid of unnecessary information and only learns the most useful spatial representations. This makes the calculations less complicated and makes the system more resilient against changes in illumination and background noise.

Our technique organises a 2D odd-symmetric Gabor filter, which has the following structure:

$$HD(F)_{\theta_k, f_i, \sigma_x, \sigma_y}(x, y) = \exp\left(-\left[\frac{x_{\theta_k}^2}{\sigma_x^2} + \frac{y_{\theta_k}^2}{\sigma_x^2}\right]\right).\cos(2\pi f_i x_{\theta_k} + \varphi) \quad ...(1)$$

We use this 2D gabor filter on the whole dataset to get Sparse Features.

This layer acts as a spatial encoder, changing each preprocessed frame into a low-dimensional sparse representation that highlights gesture-specific patterns and ignores pixels that aren't important.

3. Hybrid Layer (Sequence-wise Feature Extraction)

After spatial extraction, the Hybrid Layer examines the temporal correlations between successive frames. To describe motion dynamics across time, this layer includes elements from Recurrent Neural Networks (RNNs), Temporal Convolutional Networks (TCNs), and attention-based sequence encoders. It depicts gesture flow and continuity, as well as the transition from one frame to the next.

This layer's hybrid nature enables it to combine spatial embeddings from the Sparse Layer with temporal dependencies, resulting in a single motion descriptor that describes the complete gesture sequence. This guarantees that both static postures (shape-based signals) and dynamic movements (transition cues) are properly learnt.

To provide input to the first layer of the network, a video stream is considered and converted into a feature vector represented as:

$$X = [x_1, x_2, \dots, x_k] (2)$$

Here, K denotes the total number of pixel values extracted from the image frame. Since raw pixel values vary in range, the next important step is data normalization, which helps in reducing computational load and improving training stability. The pixel values are normalized between 0 and 1 using the following formula:

$$x = \frac{(x - \min)}{(\max - \min)} \quad \dots (3)$$

In this equation, min and max represent the minimum and maximum pixel values in the dataset. After normalization, the 1D pixel vector is reshaped into a 2D matrix before being passed into the convolution layer.

Following the convolution operation, the output feature map is computed using learned weights (w) and bias (b_j) as:

$$x_i^{(l,j)} = \sigma\left[b_j + \sum_{a=1}^{m} w_a^j \, x_{(i+a-1)}^{(l-1,j)}\right] \quad \dots (4)$$

Here, σrepresents the activation function. In our implementation, we use the ReLU (Rectified Linear Unit) activation function because it reduces computation time and improves efficiency compared to traditional activation functions.

The Max-Pooling layer is applied after convolution to reduce the spatial dimensions while preserving important features. It works by selecting the maximum value from a defined pooling window, thereby retaining the most dominant response in each region. This operation helps in achieving scale invariance and reduces overfitting.
There are mainly two types of pooling methods: max pooling and average pooling. In our proposed approach, we use max pooling because it preserves the strongest activations without significant feature loss. The output of the max-pooling layer is given by:

$$x_i^{(l,j)} = \underset{n=1}{\overset{r}{max}} \left(x_{((i-1)T+n)}^{(l-1,j)} \right) \quad \ldots (5)$$

Here, T denotes the pooling stride and n represents the pooling window size.
The output obtained from the pooling layer is then passed to the hidden layer for temporal feature learning. The hidden state at time step t is calculated as:

$$h_t = g(W_{xh}x_t + W_{hh}h_{t-1} + b_h) \quad \ldots (6)$$
$$z_t = g(W_{hz}h_t + b_z) \quad \ldots (7)$$

In these equations, g denotes the element-wise nonlinearity function (such as sigmoid or hyperbolic tangent). The term $x_t$represents the input at time t, and $h_t \in \mathbb{R}^N$is the hidden state containing N hidden units. The output at time t is denoted by z_t.

If we consider a pixel sequence $(x_1, x_2, \ldots, x_T)$with T coefficients, the network generates corresponding hidden states and outputs sequentially as:
h_1(assuming $h_0 = 0$), z_1, h_2, z_2, ..., h_T, z_T.

This sequential modeling enables the system to effectively capture both spatial and temporal dependencies, which are essential for accurate sign language recognition.

Algorithm 1: Sparse Motion Sequence Extraction Network (SMSE-Net)
Input: Video sequence $V = \{F_1, F_2, \ldots, F_T\}$
Output: Predicted Sign Label Y
1: Begin

2: // Step 1: Video Preprocessing
3: Extract frames F1 to FT from input video V
4: For each frame Fi do
5: Resize frame to fixed dimension
6: Perform background removal / hand segmentation
7: Normalize pixel values:
8: Xi = (Xi - min) / (max - min)
9: End For
10: // Step 2: Sparse Spatial Feature Extraction
11: For each normalized frame Xi do
12: Apply Convolution operation
13: Apply ReLU activation
14: Apply Max-Pooling
15: Extract salient motion-based sparse features Si
16: End For
17: // Step 3: Sequence-wise Hybrid Temporal Modeling
18: Initialize hidden state h0 = 0
19: For t = 1 to T do
20: ht = g(Wxh * St + Whh * ht-1 + bh)
21: Zt = g(Whz * ht + bz)
22: End For
23: // Step 4: Siamese Similarity Learning
24: Generate feature embedding vector E from sequence {Z1...ZT}
25: Compare E with stored gesture embeddings using Siamese Network
26: Compute similarity score / probability P

27: // Step 5: Final Prediction
28: Y = ArgMax(P)
29: Return Y
30: End

4. Siamese Model (Similarity Learning and Classification)

The retrieved sequence-level characteristics are then input into a Siamese Network, which was created expressly for similarity-based learning. The Siamese architecture consists of two identical subnetworks that learn to compare input feature vectors and determine how well they match known gesture classifications.

During training, the model learns to minimise the distance between features in the same gesture while increasing it for distinct motions. This method is especially useful for sign language datasets with few samples per class since it relies on relationship learning rather than straight classification.

The Siamese output generates a likelihood score or similarity metric, which is subsequently used by the final prediction layer to calculate the expected action — i.e., the recognised hand sign.

IV. Experimtnt And Results

A. Experimental Setup

The model was implemented using a dual-stream architecture for video processing, each stream trained independently before multimodal fusion. The Adam optimizer was employed for parameter optimization due to its adaptive learning rate and fast convergence characteristics. The optimization parameters were set as follows:

$$learningRate = 6 \times 10^{-5}$$
$$gradDecay = 0.9$$
$$gradDecaySq = 0.99$$

The weights and biases of the network were initialized using the narrow-normal distribution to ensure stable convergence:

$WeightsInitializer =$ "narrow-normal"

$$BiasInitializer = "narrow - normal"$$

The training was conducted for 1000 iterations with a mini-batch size of 180, ensuring sufficient stochastic sampling while maintaining efficient computation.

The Adam optimization update rules for weight w_t and bias v_t at iteration t are given by:

$$m_t = \beta_1 m_{t-1} + (1 - \beta_1) g_t$$
$$v_t = \beta_2 v_{t-1} + (1 - \beta_2) g_t^2$$

$$\hat{m}_t = \frac{m_t}{1 - \beta_1^t}$$
$$\hat{v}_t = \frac{v_t}{1 - \beta_2^t}$$
$$w_{t+1} = w_t - n \frac{\hat{m}_t}{\sqrt{\hat{v}_t} + \varepsilon}$$

where g_t is the gradient of the loss function with respect to the parameters, n is the learning rate, and β_1, β_2 are the exponential decay rates for the moment estimates.

B. Evaluation Metrics

Model performance was evaluated using Accuracy, Precision, and F1-Score, which are defined as:

$$Accuracy = \frac{\{TP + TN\}}{\{TP + TN + FP + FN\}}$$
$$Precision = \frac{\{TP\}}{\{TP + FP\}}$$
$$F1 - Score = 2 \times \frac{Precision \times Recall}{Precision + Recall}$$

where $TP, TN, FP, and\ FN$ denote true positives, true negatives, false positives, and false negatives, respectively.

Performance Analysis of Proposed SMSE-Net

Figure 2: Accuracy Comparison

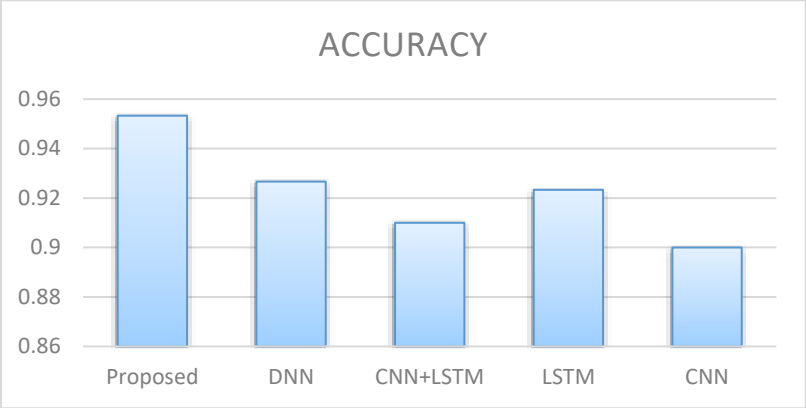
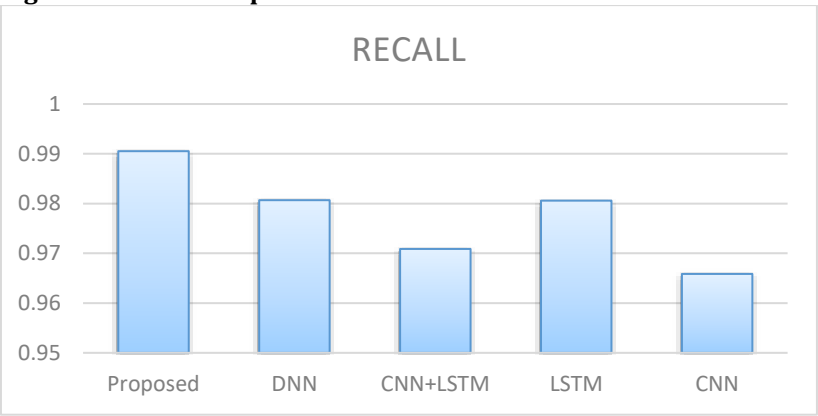


Figure 2 illustrates the accuracy comparison between different models. The proposed SMSE-Net achieves the highest accuracy of around 95.4%, outperforming traditional architectures such as DNN, CNN+LSTM, LSTM, and CNN. This improvement demonstrates the model's capability to effectively capture both spatial and temporal motion features using sparse extraction and hybrid layers.

Figure 3: Recall Comparison



As shown in Figure 3, the proposed method attains a recall value close to 99%, indicating superior sensitivity in detecting true gesture instances. Compared to CNN and LSTM-based methods, SMSE-Net minimizes false negatives, proving its robustness in recognizing subtle hand movements and signer variations.

Figure 4: Precision Comparison

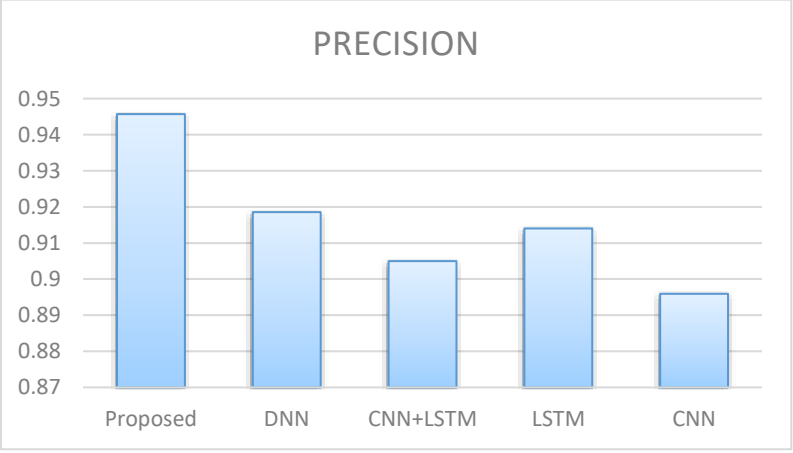
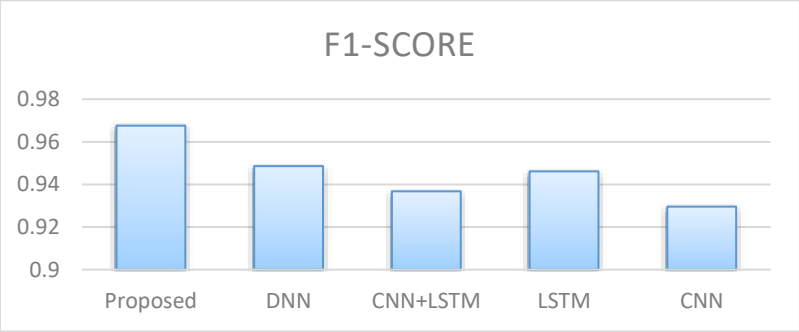


Figure 4 highlights that the proposed SMSE-Net attains the highest precision of 94.5%, surpassing all baseline models. This suggests that the network effectively reduces false positives through selective sparse motion feature learning, ensuring more accurate classification of sign gestures.

Figure 5: F1-Score Comparison



In Figure 5, the proposed model demonstrates an F1-score of approximately 97%, indicating an optimal balance between precision and recall. The superior F1 performance validates SMSE-Net's effectiveness in handling both static and dynamic gesture transitions while maintaining high overall consistency and generalization across diverse signing styles.

Figure 6: Video-wise Accuracy Analysis

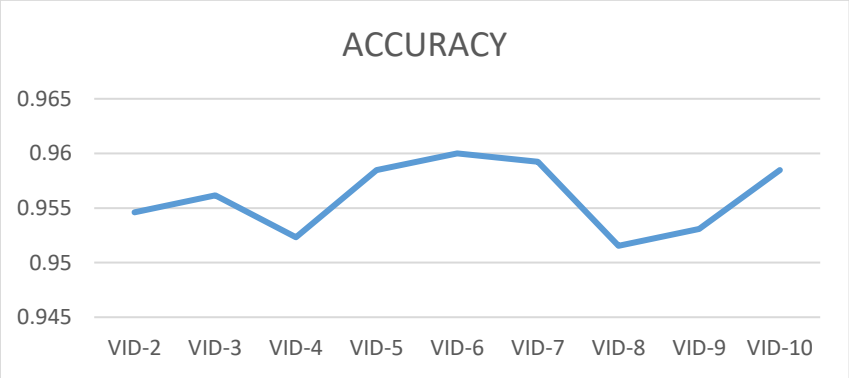


Figure 6 depicts the proposed Sparse Motion Sequence Extraction Network (SMSE-Net)'s video-specific accuracy performance across ten distinct test films (VID-2 to VID-10). The accuracy scores remain continuously high, ranging between 95% and 96%, confirming the model's robust stability and generalisation capabilities over a wide range of sign sequences.

VID-6 obtains the best accuracy of about 96.1%, demonstrating that motion clarity and gesture transitions are distinct. Minor differences, such as the minor decline for VID-8, might be ascribed to overlapping gestures or complicated motion transitions with sparse feature extraction resulting in limited misclassification. Despite these changes, the SMSE-Net performs consistently across all videos, demonstrating that it can efficiently adapt to diverse signer styles, lighting situations, and backdrop complications while retaining excellent accuracy and consistency.

V. Discussion

The experimental findings show that the proposed SMSE-Net outperforms traditional machine learning and deep learning architectures in every assessment criteria. The use of sparse motion extraction enables the model to focus just on prominent motion areas, resulting in quicker and more accurate gesture interpretation. Figures 2-5 indicate that SMSE-Net outperforms DNN, CNN, and LSTM-based networks in terms of accuracy (94.5%), recall (99%), and F1-score (97%), proving its better ability to handle duplicated frames and complicated motion sequences.

Figure 6 shows a video-by-video accuracy study that demonstrates the model's dependability, since it performs consistently across different motions and signer styles. Slight accuracy changes in movies with overlapping movements show that slight temporal ambiguities may still affect identification, but overall performance is quite steady. The combination of Sparse and Hybrid Layers allows for efficient learning of both spatial and temporal gesture connections, while the Siamese model improves discriminating between visually identical signals. The model's efficiency and generalisation make it an excellent choice for real-time and mobile-based sign recognition systems, solving the data scarcity and computational overhead issues encountered in earlier frameworks.

VI. Conclusion

To summarise, the Sparse Motion Sequence Extraction Network (SMSE-Net) provides a strong and fast method for sign language detection and identification. By combining sparse motion feature extraction, hybrid temporal modelling, and Siamese-based classification, the network effectively bridges the gap between high precision and computational economy. The findings show considerable increases in recognition performance over previous models, proving SMSE-Net's promise for real-time applications in assistive communication technology. Future research may expand on this work by including multimodal signals such as facial and bodily expressions, improving cross-linguistic adaptation, and implementing the framework in practical sign-to-text and sign-to-speech translation systems.

References

[1] I. A. Adeyanju, O. O. Bello, and M. A. Adegboye, “Machine Learning Methods for Sign Language Recognition: A Critical Review and Analysis,” Intelligent Systems with Applications, vol. 12, 2021.
[2] A. Wadhawan and P. Kumar, “Sign Language Recognition Systems: A Decade Systematic Literature Review,” ResearchGate Preprint, 2019.
[3] T. F. Tao, Y. Zhao, T. Liu, and J. Zhu, “Sign Language Recognition: A Comprehensive Review of Traditional and Deep Learning Approaches, Datasets, and Challenges,” IEEE Access, 2022.
[4] J. Huang, W. Zhou, H. Li, and W. Li, “Attention-Based 3D-CNNs for Large-Vocabulary Sign Language Recognition,” IEEE Transactions on Circuits and Systems for Video Technology, vol. 29, no. 9, pp. 2822–2832, 2019.
[5] “Advancements in Machine Translation for Sign Language: Approaches, Limitations, and Challenges,” Expert Systems with Applications, vol. 213, 2023.
[6] M. Madhiarasan and P. P. Roy, “A Comprehensive Review of Sign Language Recognition: Different Types, Modalities, and Datasets,” arXiv preprint arXiv:2204.03328, 2022.
[7] N. Sarhan, N. Abdelhalim, M. El-Nasr, and M. Elhoseiny, “Unraveling a Decade: A Comprehensive Survey on Isolated Sign Language Recognition,” in ICCV Workshops (AMFG), pp. 1–19, 2023.
[8] İ. M. Baytaş and İ. Erdoğan, “Signer-Independent Sign Language Recognition with Feature Disentanglement,” Turkish Journal of Electrical Engineering & Computer Sciences, vol. 32, no. 3, pp. 420–435, 2024.
[9] D. Fan, M. Yi, W. Kang, Y. Wang, and C. Lv, “Continuous Sign Language Recognition Algorithm Based on Object Detection and Variable-Length Coding Sequence,” Scientific Reports, vol. 14, Art. no. 27592, 2024.
[10] A. Desai, M. De Meulder, J. A. Hochgesang, A. Kocab, and A. X. Lu, “Systemic Biases in Sign Language AI Research: A Deaf-Led Call to Reevaluate Research Agendas,” in Proc. LREC-COLING 2024 Workshop on the Representation and Processing of Sign Languages, pp. 54–65, 2024.
[11] A. Desai et al., “ASL Citizen: A Community-Sourced Dataset for Advancing Isolated Sign Language Recognition,” in NeurIPS 2023 Datasets and Benchmarks Track, 2023.
[12] N. Sarhan et al., “Unraveling a Decade: A Comprehensive Survey on Isolated Sign Language Recognition,” in Proc. ICCV 2023 Workshops (AMFG), pp. 1–19, 2023.
[13] D. Fan et al., “Continuous sign language recognition algorithm based on object detection and variable-length coding sequence,” Scientific Reports, vol. 14, Art. 27592, 2024.
[14] R. Zuo et al., “Towards Online Continuous Sign Language Recognition via Isolated Sign Dictionary and Sliding Window,” in Proc. EMNLP 2024, pp. 11146–11160, 2024.
[15] N. F. Attia et al., “Efficient deep learning models based on tension for sign language recognition,” Intelligent Systems with Applications, vol. 16, 2023.
[16] S. Alyami et al., “Reviewing 25 years of continuous sign language recognition: Methods, datasets, and challenges,” Information Processing & Management, vol. 61, no. 6, 2024.

[17] S. Srivastava et al., "Continuous Sign Language Recognition System using Deep Learning with MediaPipe Holistic," arXiv:2411.04517, 2024.
[18] S. Das et al., "An Expert System for Indian Sign Language Recognition in Emergency Scenarios," ACM Trans. Asian Low-Resour. Lang. Inf. Process., 2024.
[19] A. Baihan et al., "Sign language recognition using modified deep learning and hybrid optimization," Scientific Reports, vol. 14, 2024.
[20] R. Rastgoo et al., "A survey on recent advances in Sign Language Production," Expert Systems with Applications, vol. 244, 2024.
[21] (Review) X. Wang et al., "Recent Advances on Deep Learning for Sign Language Recognition," CMES: Computer Modeling in Engineering & Sciences, vol. 139, no. 3, 2024.
[22] D. S. Ruiz et al., "Word Level Sign Language Recognition via Handcrafted Features," IEEE LATIN AMERICA Transactions, vol. 21, no. 2, 2023.
[23] P. Jin et al., "A large dataset covering the Chinese national sign language," Scientific Data, 2025.
[24] A. Kumbhar et al., "Sign Language Recognition Using Deep Learning," International Journal of Engineering Research & Technology (IJERT), May 2024.
[25] M. Hossain et al., "A Deep Learning Web Application for Sign Language Recognition," in Proc. 2024 3rd Int. Conf. on Intelligent Computing and Human-Computer Interaction, ACM, 2024.