Grenze International Journal of Engineering and Technology, June Issue

From Motion to Meaning: Decoding Sign Language Gestures using Deep Learning

Stuti Jain[1], Srishti Agarwal[2], Subhanshi Agarwal[3] and Ayushi Agarwal[4]
[1-4]Computer Science and Engineering Dept. ABES Engineering College, Ghaziabad, India
Email: stutijain.208@gmail.com, srishtiagl142@gmail.com, subhanshi17ag@gmail.com, ayushi.agarwal@abes.ac.in

***Abstract*— Automating the recognition of Indian Sign Language (ISL) is a challenging task that involves various methods for processing movement over time and spatial shapes. This study presents a dual-pathway analysis comparing deep learning architectures in two areas: static fingerspelling (36 classes) and dynamic word recognition (61 classes). The research looks at the balance between feature stability and architectural efficiency for the static pathway. We thoroughly assessed a MobileNetV2 baseline, a High-Capacity Vision Transformer (ViT-Huge), and a lightweight Custom Sequential CNN. Despite the ViT-Huge achieving 100% accuracy and showing great resilience under stress with a Robustness Score of 90.94%, its high number of parameters (706.48 M) resulted in a slow inference speed of 0.16 FPS on standard CPUs. In contrast, the Custom CNN emerged as the best choice for edge deployment, achieving 8.20 FPS and 95.44% accuracy with a small size of 0.69 M parameters. The study also looks at the spatiotemporal complexity of 61 isolated sign words for the dynamic pathway. We used a Bidirectional Long Short-Term Memory (BiLSTM) network to capture subtle co-articulation patterns through MediaPipe Holistic for real-time feature extraction (258 keypoints). This dynamic model achieved a weighted F1-score of 0.889 and a test accuracy of 89.2% on an independent test set, setting a strong benchmark. This study concludes that lightweight Convolutional and Recurrent architectures remain practical for latency-sensitive, real-time communication systems. Large-scale Transformers can provide the necessary robustness for "in-the-wild" static recognition.**

***Index Terms*— Indian Sign Language (ISL), Vision Transform- ers (ViT), Convolutional Neural Networks (CNN), Bidirectional LSTM (BiLSTM), Spatiotemporal Analysis, MediaPipe Holistic, Computational Efficiency, Robustness Stress Testing, Transfer Learning, Edge Computing, Human-Computer Interaction.**

I. INTRODUCTION

Automated Sign Language Recognition (SLR) enhances communication and accessibility for the Deaf and hard-of- hearing community. Creating trustworthy Indian Sign Lan- guage (ISL) recognition systems is crucial for facilitating inclusive communication in social interaction, public services, and education.

A. Motivation and Problem Statement

Automated Sign Language Recognition (SLR) is a crucial field of study in Human–Computer Interaction that improves communication and accessibility for the Deaf and hard-of- hearing.

Grenze ID: 01.GIJET.12.2.38_20

The intricate visual-gesture language known as Indian Sign Language (ISL) integrates non-manual information like facial expressions with hand shape, orientation, placement, and motion. For inclusive communication to be possible in social interaction, public services, and education, trustworthy ISL recognition systems must be developed. Managing two types of information at once—static hand configurations and dynamic motion patterns—is a major issue in ISL recognition. Dynamic gestures necessitate modeling temporal motion and co-articulation effects, whereas static gestures depend on exact spatial configurations of fingers and palms.

Pixel-based computer vision techniques are the foundation of the majority of current ISL recognition techniques. These methods are susceptible to illumination changes, backdrop complexity, camera views, and occlusions, notwithstanding their effectiveness in controlled conditions. Furthermore, edit- ing unprocessed photos or videos adds redundancy, which raises processing and storage demands and restricts real-time performance on devices with limited resources. This paper suggests an ISL recognition approach based on small skeletal landmark characteristics rather than pixel- level inputs to get around these drawbacks. Pose and hand geometric characteristics lower the dimensionality of the data and the sensitivity to the surroundings. In order to enable improved spatial and spatiotemporal modeling, the system further divides recognition into two pathways: a dynamic pathway for word-level gesture recognition and a static pathway for fingerspelling and numeric signs.

B. Scope and Research Objectives

This work's focus is on evaluating computational and archi- tectural trade-offs when creating an effective ISL recognition system. In-depth non-manual indicators like gaze and facial emotions are left for later research, while hand pose and coarse body pose elements are given priority in this study.

The primary objectives of this research are as follows:

- Feature Engineering for Robustness: Compact skeletal landmark representations can be used in place of pixel- based inputs to increase resilience against environmental changes.
- Dual-Path System Design: Split ISL recognition into two pathways: a dynamic pathway for spatiotemporal gesture modeling and a static pathway for fingerspelling identification.
- Architectural Trade-Off Analysis: Consider real-time performance on edge devices while assessing the trade-off between computing efficiency and recognition accuracy.

II. LITERATURE REVIEW

A. Narrative Review of Existing Works

- Deep Learning Method for Sign Language Recog- nition (2023): A seven-layer Convolutional Neural Network (CNN) with preprocessing methods including scaling and background removal, reaching almost 99% accuracy. Nev- ertheless, the method's applicability to real-world dynamic circumstances is confined to isolated static gestures and is still susceptible to changes in illumination and image quality.
- D-Talk: Sign Language Recognition System Using Machine Learning and Image Processing (2020): The D-Talk system used Hidden Markov Models (HMMs) in con- junction with traditional computer vision methods like SIFT, SURF, BRISK, and HSV color-space processing. The system only achieved approximately 60% accuracy and relied on a predefined gesture vocabulary, which limited its ability to handle complicated or continuous sign language expressions, even though it was capable of basic real-time interaction.
- Gesture Recognition Based on Sign Language: A Decade Systematic Literature Review (2019): This survey examined machine learning techniques such as CNNs, SVMs, HMMs, KNNs, DTW, and ensemble models and examined 117 publications that were published between 2007 and 2017. Important issues include restricted generalization between signers, a lack of standardized datasets, and reliance on controlled experimental settings were noted in the review.
- QazSL Sign Language Recognition System for Physi- cally Impaired Individuals (2025): Using models like SVM, LSTM, GRU, VGG16, ResNet-50, and YOLOv5, this work developed a hybrid machine learning and deep learning frame- work for Kazakh Sign Language detection. The system's comparatively small dataset and language-specific architecture restrict its scalability and cross-linguistic application, despite its high accuracy levels (up to 100%).
- Deep Learning Techniques for Continuous Sign Lan- guage Recognition: State-of-the-Art Review: Continuous sign language recognition techniques were divided into CNN- HMM, CNN-RNN, 3D CNN, Graph Convolutional Networks, and Transformer-based models like ViT and SignBERT in this review. Even while Transformer and hybrid CNN-RNN models performed well, issues with real-time deployment, computational cost, and dataset availability still exist.

- Towards Real-Time Recognition of Continuous Indian Sign Language Using RGB and Pose (2025): This work presented SignFlow, a continuous ISL recognition framework that combines an eight-layer Transformer trained with Con- nectionist Temporal Classification (CTC) with 3D ResNet. Although the system exhibited real-time chatbot integration and attained a Word Error Rate (WER) of 19, it still has issues with quick signing and gestures that are not in its repertoire.
- Sign Language Recognition: State-of-the-Art Re- view and Future Directions (2024): CNNs, RNNs, LSTMs, HMMs, and SVMs were among the deep learning and conven- tional machine learning techniques examined in this work. The authors emphasized that while using non-manual cues can in- crease identification accuracy, issues with dataset accessibility, mobile deployment, and sentence-level recognition still exist.
- Indian Sign Language Hand Gesture Recognition Using Deep Learning (2023): Using a dataset of 7,800 photos and HSV color-space preprocessing, this study assessed a CNN-based ISL recognition algorithm. The system's lim- ited dataset diversity and lack of dynamic gesture modeling raise questions regarding real-world generalization, even if it achieved up to 99% accuracy for static motions.
- Deepsign: Sign Language Detection and Recognition Using Deep Learning (2022): The Deepsign system identified isolated ISL signs from the 11-class IISL2020 dataset by combining LSTM and GRU networks. The method is con- strained by its restricted vocabulary and incapacity to represent continuous sign language sequences, even though it achieves 97% accuracy.
- Isolated Video-Based Sign Language Recognition Using a Hybrid CNN-LSTM Framework (2024): This work suggested a lightweight hybrid CNN-LSTM model that uses LSTM with attention for temporal modeling and MobileNetV2 for spatial feature extraction. The system only achieved 84.65% accuracy and shown low robustness in unconstrained real-world situations, despite being computationally efficient with about 0.374 million parameters.

III. METHODOLOGY

A. Overall Framework and Architecture

To separate the complex problem of Sign Language Recog- nition (SLR) into two optimization tasks, Static "State" Clas- sification (fingerspelling) and Dynamic "Command" Classifi- cation (gesture words), this study adopts a strong dual-path research approach. To address the different computational needs of spatial shapes and temporal motion, the proposed framework divides the processing pipeline into two parallel tracks: Track A for Static and Track B for Dynamic, as shown in Fig. 1. This follows the "Divide-and-Conquer" principle. The workflow has three main stages.

- Phase I (Acquisition): The image data or raw videos are ingested and routed in the system , based on the recognition task (State or Command).
- Phase II (Feature Engineering): This stage focuses on improving and standardizing data. It involves geometric enhancements and adjusting pixel sizes (64 X 64 vs. 224 X 224) for the static track. For the dynamic track, it uses MediaPipe to extract accurate skeletal graphs.
- Phase III: Specialized deep learning models receive the optimized features. These include lightweight Custom CNNs for edge efficiency and High-Capacity Vision Transformers (ViT) and BiLSTMs for maximum robust- ness.

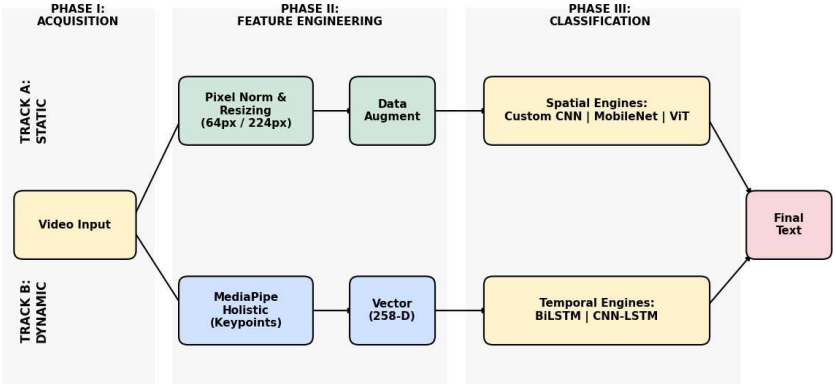

Fig. 1. Dual-Pathway Framework for ISL Recognition

B. Data Acquisition and Feature Extraction

The foundation of the dynamic system and the ROI selector for the static system is the MediaPipe Holistic framework. Traditional SLR approaches often rely on raw pixel intensity (RGB data). This reliance creates significant computational demands and makes models sensitive to changes in lighting, skin tone, and complex backgrounds. To reduce these environ- mental impacts, we use MediaPipe to extract a "topologically- aware" mesh.

- Topological Feature Vectorization: The system removes the RGB data and extracts a 258- dimensional feature vector, for each processed frame. This compact geometric representation captures the biomechanics of the signer without keeping visual noise.
  The vector v ∈ R258 consists of:
  - Pose Landmarks: 33 four-dimensional points (x, y, z, visibility). These points record the overall body posture and arm movement, which is crucial for recognizing signs that involve large sweeping motions. (33 X 4 = 132 dimensions).
  - Hand Landmarks: 21 three-dimensional coordinates (x, y, z) for each hand (Left and Right). To recognize detailed fingerspelling (2 X 21 X 3 = 126 dimensions), these coordinates capture the specific finger movements and joint angles only.
- Advantages of Skeletal Modeling: This vectorization provides three main benefits over pixel-based methods. First, the extracted coordinates are invariant to lighting changes and background clutter, directly addressing the "in-the-wild" robustness issue. Second, it inherently pro- tects user privacy by discarding facial and background imagery. Finally, compressing a standard HD frame into a simple vector of 258 values cuts data transmission by over 99.9%, allowing for highly efficient, low-latency processing on CPU-based hardware.

C. Spatial Analysis-Based Static Gesture Recognition

The static pathway assesses the trade-off between fea- ture invariance and architectural complexity for ISL finger- spelling recognition. Vision Transformer, MobileNetV2, and a lightweight Custom CNN are three architectures that were created to examine this balance. While CNN-based models rely on local spatial data and perform well in cleaner environ- ments, transformers capture global context and offer higher robustness for complicated hand articulations.

- Dual-Pipeline Preprocessing & Data Preparation: The static dataset includes around 36, 000 photos from 36 classes (ISL Alphabets A-Z and Digits 0-9). We cre- ated training (N = 28, 811) and validation (N = 7, 200) sets from the data. We set up two preprocessing pipelines to address the different needs of the architectures.
  - Efficiency of the Low-Res Pipeline: We normal- ized images to [0, 1] and resized them to 64 X 64 X 3. This pipeline feeds the Custom CNN to replicate edge environments with limited resources.
  - High-Res Pipeline (Capacity): To retain fine de- tails, images were upscaled to 224 X 224 X 3. The Mo- bileNetV2 and Vision Transformer models use this pipeline.
- Architecture I: Lightweight Custom Sequential CNN (The "Efficiency" Model)
  We built a compact Sequential CNN from scratch to set a standard for computational efficiency.
  - Structure: The network consists of three convo- lutional blocks with increasing filter depth (32 to 64 to 128). Each block uses a 3 X 3 kernel, Batch Normalization, and 2 X 2 Max Pooling.
  - Parameter Constraint: The architecture was specifically limited to a "Micro" scale with 0.69 million parameters.
  - Objective: This model aims to test the idea that simple, local feature extraction is sufficient for clean data, even if it lacks robustness under complex geometric stress.
- Architecture II: MobileNetV2 (The "Transfer Learn- ing" Baseline)
  We used MobileNetV2, pre-trained on ImageNet, as a common industry example.
  - Mechanism: This architecture employs Depthwise Separable Convolutions, which consist of a depth-wise spatial filter followed by a pointwise 1 X 1 com- bination.

$$\{Y\}_{\{i,j,k\}} = \sum_{\{u,v\}} K_{\{u,v,k\}} \cdot X_{\{i+u,j+v,k\}} \qquad (i)$$

  - Training Approach: We fine-tuned the upper layers to adapt the feature extractors to specialized ISL hand geometries after initializing the model with ImageNet weights.

- Scale: With 2.30 million parameters, this model balances the deep feature extraction capabilities of pre-trained networks and the efficiency of the Custom CNN.

➢ Architecture III: Vision Transformer with High Ca- pacity (ViT-Huge)
We created a High-Capacity Vision Transformer to ex- plore the upper limits of "Global Context" modeling. Unlike CNNs, this model processes the image as a series of flattened patches using Multi-Head Self-Attention (MSA).

- Attention Mechanism: The model uses scaled dot- product attention to calculate global dependencies

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

- "Over-Parameterization" Approach: We inten- tionally used a large architecture with 706.48 mil- lion parameters. This over-parameterization, sup- ported by aggressive data augmentation, seeks to ensure strong robustness against occlusion and to improve the model's ability to learn complex, non- local geometric relationships.

➢ Robustness Stress Test Protocol: To measure the "Price of Precision", we applied syn- thetic distortions to test the models. This included adding Gaussian Noise (σ 50) to simulate low-light sensor grain (intensity shifts of 20%), applying a 25% Center Occlusion mask to imitate physical blockages of the palm, and rotating inputs by 90◦ to check for orientation invariance.

C. Spatio-Temporal Analysis-Based Dynamic Gesture Recog- nition

This pathway is focused on categorizing full-word, dynamic signs, which consist of 61 classes. It emphasizes capturing temporal relationships and motion dynamics. The method changes continuous video input into standardized keypoint sequences by using both pure recurrent models and hybrid models.

➢ Sequence Preprocessing and Data Preparation: The dynamic models were trained on a custom-collected dataset of about 1, 952 videos. These videos covered 61 ISL word signs and lasted between 10 and 15 seconds.

- Feature Extraction: We used MediaPipe Holistic as our main method for extracting features to ensure reliable, real-time landmark detection. The input is a 258-keypoint vector for each frame. This vec- tor includes 126 Hand Landmarks and 132 Pose Landmarks (Set B). We confirmed this method with ablation research.
- Sequence Standardization: Sequences were set to a fixed length of 40 frames. This length was chosen because it represents the 75th percentile of the processed frame distribution.
- Efficiency trade-off: To lower computational costs, Frame Skipping (1-in-2 downsampling) was ap- plied. This caused a slight decrease of 0.9% in test accuracy. This was expected because of the significant drop in processing time.

➢ Architecture IV: Bidirectional LSTM (BiLSTM) The BiLSTM model was chosen as a lightweight bench-mark for measuring long-term relationships over time. Its bidirectional nature looks at the sequence in both directions. This enables the model to identify nuanced spatiotemporal co-articulation.

- Model Architecture: The architecture consists of two stacked BiLSTM layers that handle 258 fea- tures at T = 40. Following this, there is a dense classification layer that encompasses 61 classes. The overall parameter count is 263, 229.
- Recurrent Mechanism: The output gate, ot, and the internal cell state, ct, of the LSTM cell dictate how the hidden state, ht, is updated. This enhances memory retention for sequential data.

$$h_t = o_t \odot \tanh(c_t) \qquad \text{(iii)}$$

- Training Setup: The Adam optimizer (with a learn- ing rate of 1 10^{-3}) and categorical crossentropy loss function were utilized to optimize the model. It was then trained with early stopping.

➢ Architecture V: CNN-LSTM Hybrid (Spatiotemporal Model): This hybrid architecture performs spatiotemporal anal- ysis by combining spatial feature encoding from each frame with temporal learning on the keypoint sequence

$$N_{batch}, T = 40, F = 258).$$

- Spatial Feature Encoding (TimeDistributed Dense): Two TimeDistributed(Dense) layers, with 128 units and 64 units, act as a shared encoder for each frame. They project the 258 keypoint features, x_t, into a compressed space, y_t, at each time step, t:

$$y_t = \sigma(W_D x_t + b_D) \qquad \text{(iv)}$$

This component effectively compresses the spatial features from each frame.

- Temporal Feature Learning (LSTM): A com- pressed sequence $(40, 64)$ is fed into an LSTM layer with 128 units to capture the dynamics of motion. The cell state update (c_t) models time-dependent relationships by using the Forget (f_t) and Input (i_t) gates:

$$c_t = f_t \odot c_{t-1} + i_t \odot \tilde{c}_t \qquad \text{(v)}$$

- Classification: The output includes a Dropout layer ($p = 0.5$) followed by a Dense layer with Softmax activation for 61 classes.

➢ Temporal Robustness Stress Protocol: To evaluate "Stability under Temporal Distortion," we examine how the sequence models respond to artifacts common in dynamic sign acquisition.

- Sequence Dilation (Speed Irregularity): This method goes back to 40 frames after skipping up to 20% of them.
- Boundary Stress (Temporal Clipping): This ap- proach replaces the first or last 15% of frames with zeros as padding.
- Keypoint Feature Dropout: This technique ran- domly sets the entire 258-feature vector to zero for 10% of all frames.

D. Evaluation Strategy

The dataset was partitioned using a stratified split strategy to ensure class balance:

➢ Static Dataset: 36,000 images split into 80% training and 20% validation.

➢ Dynamic Dataset: 1,464 videos split into 85% training and 15% testing.Evaluation metrics include Top-1 Ac- curacy, Inference Latency (FPS), and Robustness Scores (for static models), providing a holistic assessment of real-world viability.

IV. Outcomes and Discussion

A. Analysis of Static Gesture Recognition

Complexity and Real-Time Viability

A crucial trade-off between latency and capacity is revealed by benchmarks on a standard CPU. With 8.20 FPS (121.46 ms) and a small footprint (0.69M parame- ters, 7.95 MB), the Custom CNN turned out to be the most practical for edge deployment. However, the ViT- Huge (706.48M parameters) faced a major bottleneck at 0.16 FPS (about 6.35 seconds per frame). This indicates that, despite Transformers' higher capacity, their quadratic computational cost requires specialized hardware acceleration for live translation.

Sturdiness Under Stress

Adversarial testing revealed specific architectural fea- tures, including:

➢ Noise Invariance: MobileNetV2 performed poorly at 10.88% due to its reliance on texture. By serving as a low-pass filter, the Custom CNN (83.81%) outperformed the baseline, while the ViT (94.43%) predominated through global attention mechanisms.

➢ Occlusion Stability: CNNs were unable to general- ize (< 18%) with 25% masking because they lacked context inference. In order to validate its "Global Receptive Field" for inferring missing geometry, the ViT retained 80.21%.

➢ Rotational Invariance: ViT (98.18%) demonstrated better invariance due to strong position embeddings. In contrast, the scratch-trained CNN struggled under rotation, with a performance of only 11.38%.

Table I: Computational Benchmark: Efficiency Vs. Robustness

Metric	Custom CNN	MobileNetV2	ViT(High-Capacity)
Input Resolution	64 × 64	224 × 224	224 × 224
Parameters	0.69 M	2.30 M	706.48 M
Disk Size	7.95 MB	9.72 MB	~2.6 GB
Inference (CPU)	8.20 FPS	6.90 FPS	0.16 FPS
Latency	121.46 ms	144.08 ms	6,355.26 ms
Clean Accuracy	95.44%	98.10%	100.00%
Robustness Score*	35.90%	38.37%	90.94%
*Mean accuracy across Noise, Occlusion, and Rotation stress tests.			

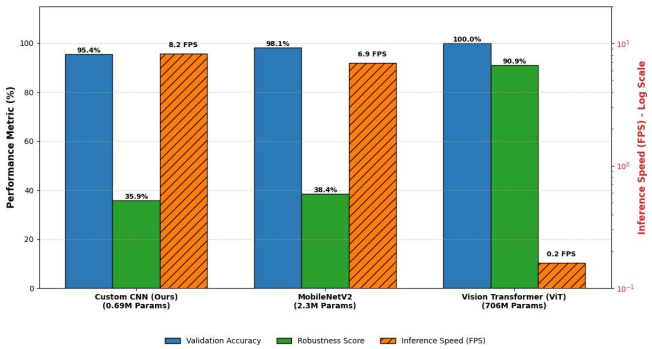

Fig. 2. Model benchmark- Precision, Robustness and Efficiency

B. Analysis of Dynamic Gesture Recognition

Quantitative Performance and Model Efficacy:
The comparison highlights the importance of accurate input features for recurrent models.

TABLE II: PERFORMANCE AND COMPUTATIONAL COMPLEXITY ANALYSIS

Metric	BiLSTM (Proposed)	CNN–LSTM (Baseline)
Test Accuracy (%)	89.2	74.24
Weighted F1-score	0.889	0.738
Parameters (M)	0.263	0.347
Inference Profile	Low Latency CPU	Moderate CPU Latency

- Efficacy Hypothesis Validation: The performance gap of over 14.9% shows that compressing interme- diate features harms the engineered keypoint data. The TimeDistributed Dense layers in the hybrid model lowered the quality of the 258-feature vector. This caused a loss of subtle motion dynamics. The pure BiLSTM performed better because it directly processed the rich, uncompressed input and used its bidirectional context.

Training Dynamics and Error Analysis:
- Training Stability and Convergence: The train- ing convergence curves (Validation Accuracy vs. Epochs) show that the BiLSTM has a better op- timization landscape. The BiLSTM reached faster convergence and a higher final accuracy (approxi- mately 91%) compared to the CNN-LSTM Hybrid (approximately 75.5%). This visually confirms that the hybrid structure added unnecessary complexity and instability for this task.
  The chart shows that the BiLSTM converges faster and achieves a higher final validation accuracy of about 91%. This supports the idea that focusing solely on temporal modeling is better for keypoint features.
- Error Analysis: Misclassification in the BiLSTM mainly happened between signs that had high visual or temporal similarity, such as Tiger vs. Lion. This suggests that the system currently struggles with small ambiguities between closely related classes.

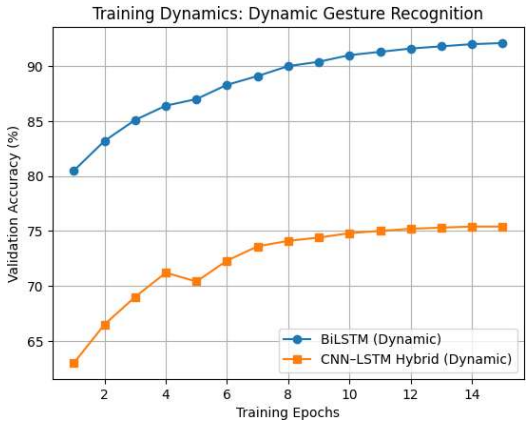

Fig. 3. Training Dynamics of Dynamic Gesture Recognition Models

V. Conclusion

In order to recognize Indian Sign Language (ISL), this study suggested a dual-pathway paradigm that distinguished between dynamic gesture analysis and static fingerspelling. To comprehend the trade-off between computing efficiency, accuracy, and robustness, several deep learning architectures were assessed.

With 100% accuracy and a robustness score of 90.94% for static gesture detection, the Vision Transformer (ViT-Huge) demonstrated great feature invariance. Nevertheless, its low inference speed (0.16 FPS) and significant computing cost make it impractical for real-time implementation. On the other hand, the Custom Sequential CNN is appropriate for edge devices with limited resources, achieving 95.44% accuracy with just 0.69M parameters and 8.20 FPS.

Skeletal landmarks that were modeled using a BiLSTM network were extracted using MediaPipe Holistic for dynamic gesture detection. The model demonstrated successful tem- poral modeling of gesture sequences, achieving 89.2% test accuracy and a weighted F1-score of 0.889.

Overall, the findings show that while massive Transformer models offer greater robustness for challenging real-world circumstances, lightweight convolutional and recurrent archi- tectures continue to be the most practical alternatives for real- time sign language detection.

Future Scope

In future study, the suggested framework can be extended to Continuous Sign Language Recognition (CSLR), which interprets whole sentences from unsegmented video sequences. This can be accomplished using approaches like Connectionist Temporal Classification (CTC), which allows for sequence-to- sequence learning without the need for accurate frame-level annotation. Furthermore, adding attention mechanisms into the BiLSTM architecture may improve recognition resilience by allowing the model to prioritize the most informative hand and body motion patterns during gesture interpretation.

References

[1] Bambang Krismono Triwijoyo, Lalu Yuda Rahmani Karnaen, Ahmat Adil, “Deep Learning Approach for Sign Language Recognition,”2023.

[2] Bayan Mohammed Saleh, Reem Ibrahim Al-Beshr, Muhammad Usman Tariq., D-Talk: Sign Language Recognition System for people with disability using Machine Learning and Image Processing, 2020.

[3] Ankita Wadhawan, Prateek Bhatia, Sign Language Recognition System: A Decade Systematic Literature Review, 2019.

[4] Lazzat Zholshiyeva, Tamara Zhukabayeva, Dilaram Baumuratova, Aza- mat Serek, “Design of QazSL Sign Language Recognition System for Physically Impaired Individuals.,” 2025.

[5] Asma Khan, Seyong Jin, Geon-Hee Lee, Gul E. Arzu, L. Minh Dang, Tan N. Nguyen, Woong Choi, Hyeonjoon Moon, “Deep Learning Ap- proaches for Continuous Sign Language Recognition: A Comprehensive Review,” 2025.

[6] M. Geetha, Neena Aloysius, Darshik A. Somasundaran, Amritha Raghu- nath, Prema Nedungadi, “Toward Real-Time Recognition of Continuous Indian Sign Language: A Multi-Modal Approach Using RGB and Pose,” 2025.

[7] Bashaer A. Al Abdullah, Ghada A. Amoudi, Hanan S. Alghamdi, Ad- vancements in Sign Language Recognition: A Comprehensive Review and Future Prospects, 2024.

[8] Harsh Kumar Vashisth, Tuhin Tarafder, Rehan Aziz, Mamta Arora, Alpana (Manav Rachna University, India), Hand Gesture Recognition in Indian Sign Language Using Deep Learning, 2023.

[9] D. Kothadiya, C. Bhatt, K. Sapariya, K. Patel, A.-B. Gil Gonzalez, J.M. Corchado, Deepsign: Sign Language Detection and Recognition Using Deep Learning, 2022.

[10] Diksha Kumari and Radhey Shaym Anand, Isolated Video-Based Sign Language Recognition Using a Hybrid CNN-LSTM Framework Based on an Attention Mechanism, 2024.

[11] Foteinos, K., Cani, J., Linardakis, M., Radoglou-Grammatikis, P., Ar- gyriou, V., Sarigiannidis, P., ... & Papadopoulos, G. T. (2025). Visual Hand Gesture Recognition with Deep Learning: A Comprehensive Re- view of Methods, Datasets, Challenges and Future Research Directions. arXiv preprint arXiv:2507.04465.

[12] Zholshiyeva, L., Manbetova, Z., Kaibassova, D., Kassymova, A., Tashen- ova, Z., Baizhumanov, S., ... & Aikhynbay, K. (2024). Human-machine interactions based on hand gesture recognition using deep learning methods. International Journal of Electrical & Computer Engineering (2088-8708), 14(1).

[13] Hax, D. R. T., Penava, P., Krodel, S., Razova, L., & Buettner, R. (2024). A novel hybrid deep learning architecture for dynamic hand gesture recognition. IEEE Access, 12, 28761-28774.

[14] Eid, A., & Schwenker, F. (2023). Visual static hand gesture recognition using convolutional neural network. Algorithms, 16(8), 361.

[15] Buttar, A. M., Ahmad, U., Gumaei, A. H., Assiri, A., Akbar, M. A., & Alkhamees, B. F. (2023). Deep learning in sign language recognition: a hybrid approach for the recognition of static and dynamic signs. Mathematics, 11(17), 3729.

[16] Yaseen, Kwon, O. J., Kim, J., Jamil, S., Lee, J., & Ullah, F. (2024). Next- gen dynamic hand gesture recognition: Mediapipe, inception-v3 and lstm-based enhanced deep learning model. Electronics, 13(16), 3233.
[17] Mazhar, O., Ramdani, S., & Cherubini, A. (2021). A deep learning framework for recognizing both static and dynamic gestures. Sensors, 21(6), 2227.
[18] Malviya, S., Mahajan, A., & Sethi, K. K. (2025, May). Machine Learning Framework for Intelligent Hand Gesture Recognition: An Application to Indian Sign Language and Hand Talk. In International Conference on Recent Advancements and Modernisations in Sustainable Intelligent Technologies and Applications (RAMSITA 2025) (pp. 496- 512). Atlantis Press.
[19] Ingoley, S., & Bakal, J. (2025). Efficient Models Based on Deep Learning Technique for Indian Sign Language. Procedia Computer Science, 258, 318-331.
[20] Mohammed, A. A. Q., Lv, J., & Islam, M. S. (2019). A deep learning- based end-to-end composite system for hand detection and gesture recognition. Sensors, 19(23), 5282.