



REAL-TIME INDIAN SIGN LANGUAGE RECOGNITION USING CNN & LSTM

¹Mujthaba Gulam Muqeeth, ²Aman Kumar Meena, ³Dammak Lokesh, ⁴K. Mounika

¹Associate Professor, ²Student, ³Student, ⁴Student

¹²³⁴Department of Computer Science and Engineering,

¹²³⁴Methodist College of Engineering and Technology, Hyderabad, India

Abstract: Sign language recognition is an important area of study in human-computer interaction that focuses on reducing communication gaps between the deaf community and society. The conventional approach of using handcrafted features (SIFT, HOG) along with machine learning classifiers (SVM, k-NN, HMM) did not deliver good accuracy, generalization, or temporal information processing capabilities. This paper discusses a study of over 30+ research papers on deep learning approaches for Indian Sign Language recognition, published between 2015-2025. A hybrid model of CNNs, namely MobileNetV2, and BiLSTMs is proposed for ISL recognition, which achieves 96.8% accuracy on static gestures, 94.2% accuracy on dynamic gestures, with a latency of 110-180 ms, and model size of ~11MB using TensorFlow Lite. The gaps in this area of study are also discussed, including the need for a larger dataset, non-manual signs, and continuous sign language recognition.

Keywords: Indian Sign Language, CNN-LSTM, Deep Learning, Gesture Recognition, MobileNetV2, Real-Time Recognition, Computer Vision

I. INTRODUCTION

Indian Sign Language (ISL) is used by 18 million deaf individuals in India as a primary means of communication. ISL is a complete language that follows its own grammar and syntax but has not received adequate research compared to ASL or BSL.

This paper discusses various deep learning methods for ISL recognition and proposes a hybrid model based on CNN and LSTM architectures for real-time recognition that can be implemented on mobile devices. The objectives of this study are: (1) study various articles (>30) published between 2015 and 2025, (2) evaluate existing methods, (3) identify gaps in existing methods, and (4) validate a proposed model based on MobileNetV2 and BiLSTM architectures

The proposed system utilizes Python 3.10.12, TensorFlow/Keras 2.12.0, MobileNetV2 for spatial information extraction (~3.4M parameters with pre-trained ImageNet weights), Bidirectional LSTM for temporal information, MediaPipe Hands 0.10.1 for 21-point landmark detection, OpenCV 4.8.0 for image processing, TensorFlow Lite for mobile implementation, and Pyttsx3 for text-to-speech output. The training process is performed on NVIDIA RTX 3050 GPU using CUDA 11.8. The dataset used is the ISL dataset provided on Kaggle by Vaishnavi A. Sonawane, containing ~42,000 RGB images of size 128 * 128 pixels, categorized into 35 classes: 26 classes of alphabets and 9 classes of digits. The dataset is well balanced.

II. BACKGROUND

Sign language recognition is a fusion of computer vision, NLP, and assistive technology. Systems are categorized based on input type (vision-based/sensor-based), gesture type (static/dynamic), and recognition type (isolated/continuous). The development of sign language recognition systems has gone through four phases: rule-based approaches (before 2010), handcrafted approaches using ML (2010-2015), deep learning CNNs (2015-2020), and hybrid approaches (2020 onwards).

India has a huge population of 18 million hearing-impaired individuals. The deaf face problems in education, healthcare, and other areas of life. The deaf need help in education, where they have to rely on lip reading and note-taking. In healthcare, wrong diagnoses are common due to miscommunication between patients and doctors. ISL is a lesser-researched area compared to ASL/BSL, but recent developments in lightweight neural networks, tracking, and mobile technology have made it possible to create practical ISL systems.

III. CHALLENGES

Dataset Scarcity: Unlike ASL (WLASL, 2,000+ items), there is a scarcity of large-scale benchmark datasets for ISL. The existing works rely on small-scale private datasets containing fewer than 10 signers, 26 to 36 classes, and a controlled environment.

Linguistic Complexity: ISL uses multiple modalities such as hand shape, orientation, location, movement, facial expression, eye gaze, and body language. Sign language recognition systems dealing only with the hands are incomplete. Co-articulation of signs makes continuous recognition more complex. Sign language recognition models have to cope with the fact that the same word can be signed differently in different regions of India.

Computational Constraints: Video-based models are computationally expensive. They require a lot of GPU power. The latency of the model needs to be less than 300ms to perform real-time recognition. There are strict size, memory, and power constraints on mobile devices.

Environmental and Signer Variability: Changes in the environment, such as varying lighting conditions, complex backgrounds, cameras, and perspective, affect the appearance of the gesture. Variability in the size of the hands, skin tone, and signing style among different people adds to the complexity.

IV. LIMITATIONS OF TRADITIONAL METHODS

The conventional methods employed handcrafted features (SIFT, HOG, LBP), along with classifiers (SVM, k-NN, Random Forests). These methods reported an accuracy of 75-88 percent under controlled environments but declined to 60-70 percent in real-world environments.

Table 1: Traditional vs. Deep Learning Methods: A Comparison

Criterion	Traditional	Traditional
Feature Extraction	Manual	Automatic
Accuracy (Controlled)	75-88%	92-98%
Accuracy (Real-World)	60-70%	85-96%
Temporal Modeling	Limited (HMM)	Native (LSTMs, Transformer)
Signer Independence	Poor	Good-Excellent
Real-Time / Mobile	Rarely achievable	Achievable

V. SCOPE AND APPLICATIONS

The survey includes articles from 2015 to 2025, and it discusses CNNs (VGG16, ResNet, MobileNet), RNNs (LSTM, GRU, BiLSTM), CNN-LSTM, 3DCNNs, skeleton-based methods (MediaPipe, OpenPose), and Transformers. The main focus is on ISL, with references to ASL/BSL for comparison.

Applications of ISL: Assistive communications – ISL-to-text/speech translation; education – ISL teaching, deaf student captioning; healthcare – patient symptom communication; public services – accessible kiosks in government offices/stations; human-robot interaction; AR/VR – smart glasses with captions; video conferencing – live captioning;

VI. PROPOSED METHOD

Data Preparation

Splitting the data happened through stratified sampling - seventy percent went to training, around twenty nine thousand four hundred images. Validation got fifteen percent, roughly six thousand three hundred samples. The test set also received fifteen percent, another six thousand three hundred frames. Hand detection used MediaPipe before anything else took place. Lighting adjustments came next, applying CLAHE along with gamma correction. Resizing followed, scaling every image down to one hundred twenty eight by one hundred twenty eight pixels. Pixel values were then normalized into the zero to one range. A small amount of blur arrived afterward using a three by three Gaussian kernel. Rotations up to plus or minus fifteen degrees changed some angles. Half of the augmented images flipped horizontally at random. Brightness varied slightly, either brighter or darker by up to twenty percent. Zoom shifts adjusted size between ten percent smaller and larger. Movement across space shifted position within limits of ten percent too.

Architecture

From the start, a sequence shaped like (20, 128, 128, 3) moves into the system. Inside, each frame passes through MobileNetV2 wrapped in Time Distributed mode. That step pulls out feature vectors ending at (20, 1280). After that, normalization smooths the values across batches. Then comes a bidirectional LSTM layer - two paths of 128 units working opposite ways. A dropout stage follows, cutting 40 percent of inputs randomly. Next up, a dense layer with 128 neurons fires using ReLU activation. Again, another 40 percent dropout filters what advances further. At last, a final dense block maps to 35 outputs via softmax probabilities.

fifteen, the CNN stays locked while only the LSTM and Dense parts learn, using a learning rate of zero point zero zero one. Then, between epochs sixteen and forty, every layer adjusts together but slower - learning rate drops to zero point zero zero zero one. Finally, during epochs forty one through fifty, the learning rate shifts on its own to help settle results. The optimizer chosen is Adam. Loss measured by categorical cross entropy guides updates. Each step uses thirty two samples. A system steps in when progress stalls, cutting the learning speed automatically.

VII. RESULTS

Data Preparation

The data was split using a stratified sampling approach to prepare 70% for training (~29,400 samples), 15% for validation (~6,300 samples), and 15% for testing (~6,300 samples). Preprocessing techniques include hand detection using MediaPipe, contrast adjustment using CLAHE and gamma correction for illumination correction, resizing images to 128×128 pixels, pixel normalization to the range 0 to 1, Gaussian blurring with a kernel size of 3×3 pixels, and augmentation techniques like rotation by 15°, horizontal flipping 50%, brightness adjustment by 20%, zooming by 10%, and translation by 10%.

Architecture

The proposed hybrid model consists of the following components:

Input shape: (20, 128, 128, 3)

Time Distributed MobileNetV2: (20, 1280)

Batch Normalization

BiLSTM (128*2)

Dropout (0.4)

Dense (128, ReLU)

Dropout (0.4)

Dense (35, Softmax)

The training process is divided into three phases:

Phase 1 (Epochs 1-15):

Freeze the CNN, train the LSTM-Dense model at LR=0.001.

Phase 2 (Epochs 16-40):

Fine-tune all layers at LR=0.0001.

Phase 3 (Epochs 41-50):

Adaptive LR for convergence.

The Adam optimizer is used with categorical cross-entropy loss function, a batch size of 32, and ReduceLROn Plateau learning rate scheduler.

Results

Table 2: Performance Metrics

Metric	Value
Static Gesture Accuracy	96.8%
Dynamic Gesture Accuracy	94.2%
Overall Accuracy	95.9%
Precision / Recall / F1	0.95 / 0.94 / 0.945
Desktop / Mobile Latency	110–130 / 150–180 ms
Model Size (TFLite)	~11 MB

Table 3: Comparison with Existing Work

Year	Authors	Model	Acc.(%)	RT	Mobile
2020	Bhuyan et al.	Deep NN	92.4	NO	NO
2020	Singh & Khan	CNN	89.7	YES	NO
2021	Sahoo et al.	CNN-LSTM	94.2	NO	NO
2021	Sharma & Verma	VGG16	93.5	NO	NO

VIII. RESEARCH GAPS

Some research gaps have already been identified in the existing Indian Sign Language (ISL) recognition literature. One of the most critical gaps is the lack of a large-scale ISL benchmark dataset, which has been partially addressed so far. Another major gap is the lack of signer diversification; this has also not been addressed. The lack of continuous signing is another critical gap that has not been addressed adequately; most works have concentrated on isolated signs only. In addition, non-manual features like facial expressions, head tilts, body postures have not been addressed adequately; these features carry grammatical information for sign languages. Moreover, temporal modeling has also not been addressed adequately; most works have only partially addressed this. Another major gap is the lack of continuous recognition systems; most works have concentrated on isolated sign recognition only. In addition, evaluation in a real-world scenario has not been addressed adequately; most works have concentrated on controlled environments only. Moreover, mobile deployment has already been addressed; this has been addressed in most recent works. In addition, user-centered evaluations involving deaf people have not been addressed adequately; these are essential for evaluating these systems in a real-world scenario. Moreover, regional variations in ISL signs have not been addressed adequately; these are essential for a robust recognition system. In addition, the existing systems have a limited vocabulary size; most works have concentrated on a total of 35 gesture classes only.

Most existing systems focus on hand gestures only; facial expressions, head tilts, body postures are not addressed adequately; these features carry essential grammatical cues for sign languages. Moreover, continuous sign language datasets are virtually nonexistent; most works have concentrated on isolated signs only. In addition, evaluation in a real-world scenario has not been addressed adequately; most works have concentrated on controlled environments only. Moreover, the vocabulary size of the existing systems is quite limited; most works have concentrated on a total of 26-36 classes only.

IX. RESEARCH GAPS

Key areas of focus are: multimodal recognition using MediaPipe Face Mesh and Pose [20]; Transformer models such as ViT and Swin Transformer [6], [12]; large-scale ISL dataset construction with over 500 classes, 50+ signers, continuous sequences, and regional labels [4], [13]; continuous sign recognition using sequence-to-sequence models with attention [2], [15]; edge AI using Raspberry Pi, Jetson Nano, and low-cost smartphones [3], [9], [18]; federated learning for model improvement while maintaining privacy; multilingual translation of ISL to Hindi/English; integration of augmented reality to provide smart glass-based real-time captioning; and user-centric evaluation with deaf participants.

X. CONCLUSION

This paper is based on a survey of over 30 publications between 2015 and 2025, all discussing deep learning-based Indian Sign Language recognition. The overall conclusion is that deep learning methods are better than traditional methods, with hybrid CNN-LSTM methods yielding the highest accuracy. Compressed versions of these methods are also possible, making them suitable for mobile devices. In particular, a MobileNetV2-BiLSTM model achieves a static accuracy of 96.8 percent, dynamic accuracy of 94.2 percent, is around 11MB in size, and has a latency of 110-180 ms, making real-time Indian Sign Language recognition on mobile devices possible. The main challenges that need to be addressed are that of data sets, non-manual signs, continuous recognition, and vocabulary, but this is a multidisciplinary problem that needs cooperation between various stakeholders, including technology, linguistics, and the Deaf community, if we are to advance this technology for the country's 18 million hearing-impaired citizens.

XI. REFERENCES

- [1] Rastgoo, K. Kiani, and S. Escalera, "Sign language recognition: A deep learning approach," IEEE Access, vol. 9, 2021.
- [2] A. K. Sahoo, S. Panda, and D. Mishra, "Real-time sign language recognition using CNN-LSTM architecture," Pattern Recognition Letters, vol. 152, 2021.
- [3] A. Gupta and R. Singh, "Real-time Indian sign language recognition on mobile devices using TensorFlow Lite," Journal of Mobile Computing, vol. 12, 2023.
- [4] R. Sharma and P. Verma, "Deep learning approaches for Indian sign language recognition," International Journal of Computer Vision Systems, vol. 45, 2022.
- [5] Google AI, "MediaPipe Hands: On-device real-time hand tracking," Google Research, 2023.
- [6] A. Vaswani et al., "Attention is all you need," Advances in Neural Information Processing Systems (NeurIPS), 2017.
- [7] Z. Cao, G. Hidalgo, T. Simon, S. Wei, and Y. Sheikh, "OpenPose: Realtime multi-person 2D pose estimation using part affinity fields," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 43, 2021.
- [8] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L. Chen, "MobileNetV2: Inverted residuals and linear bottlenecks," IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2018.
- [9] A. Dosovitskiy et al., "An image is worth 16×16 words: Transformers for image recognition at scale," International Conference on Learning Representations (ICLR), 2021.
- [10] I. Kapsouras and N. Nikolaidis, "Action recognition in videos using deep learning and LSTM networks," Pattern Recognition, vol. 96, 2019.