Improved Multimodal Integration with Attention for Live Nepali Sign Language Interpretation

Ashish Pandey
Department of Computer and Electronics Engineering
Khwopa College of Engineering, Nepal
ashishpanday9818@gmail.com

Abstract—**This work introduces a framework for attention-based multimodal integration in real-time Nepali Sign Language (NSL) interpretation by combining hand landmarks, face details, and pose information. I have fine-tuned on frame series from five NSL gestures as a result the system attains 97.39% validation precision and 31.18 FPS on a GPU. Experiments reveal 83.48% precision in five-epoch scratch training and 46.96% absent ImageNet initialization. The codebase is shared openly to promote NSL inclusivity.**

Index Terms—**Nepali gesture communication, integrated modalities, attention systems, instant interpretation, visual computing**

I. Introduction

Over 300,000 deaf people in Nepal depend on Nepali Sign Language (NSL) [12], [13]. Despite this significant population, Nepal currently lacks real-time interpretation tools for NSL, creating huge barriers to communication, education, and social inclusion for the deaf community.

Sign Language Recognition technology has the potential to bridge this gap. But developing systems for NSL presents us the unique challenges such as distinct grammatical structure, cultural gestures, and limited available datasets.Traditional method relies on single-modality systems that capture only hand movements, which can result in reduced accuracy while dealing with complex gesture that includes facial expression and body posture.

For effective sign language interpretation the it requires the This means we need methods that can handle all these different types of information together. This means we need methods that can handle all these different types of information together.

To address this challenge, we propose an attention-based multimodal framework that integrates hand landmarks, facial expressions, and body pose information. The system employs ResNet-18, bidirectional LSTM networks, and multi-head attention mechanisms to achieve accurate real-time conversion of NSL gestures into text.

This system uses attention-driven multimodal integration to handle hand landmarks, facial cues, and body postures for converting NSL gestures to text.

The key contributions include:

- An innovative encoder-attention-decoder setup by blending three data types for reliable gesture detection.
- A Colab script covering data import, enhancement, and optimal model saving.
- Leading results: 97.39% validation precision, 31.18 FPS for live processing.
- Thorough experiments evaluating pretraining advantages and learning periods.

Here's how we've organized the rest of the paper: Section II reviews related work in sign language recognition and multimodal learning approaches. Section III presents the methodology, including the detailed architecture design and training procedures. Section IV provides comprehensive experimental results and analysis. Section V discusses the implications of the findings, limitations, and future directions. Finally, Section VI concludes the paper with a summary of contributions and potential impact on NSL accessibility.

II. Related Work

Sign language recognition has shifted from basic handcrafted features to deep learning methods. CNN-LSTM setups [1], [2] became popular for capturing both spatial features and temporal patterns in gesture videos. Later, attention mechanisms [3] improved sequence modeling by letting models focus on important time segments

Most recent work combines multiple information sources for better results. Using hand movements, facial expressions, and body pose together makes systems more robust [7], [9]. Hu et al. [9] showed that combining these with attention can improve accuracy by 2-3%. Zhang et al. [11] developed smarter ways to balance different types of information using hypernetworks, cutting model size and enhancing conversion quality.

For NSL specifically, datasets remain limited compared to American Sign Language. NSL23 [8] is one of the few available datasets, but more comprehensive resources are needed. Recent reviews [10] highlight that combining multiple data types and limited datasets are major challenges, especially for real-time systems.

III. Methodology

A. System Summary

The process in Figure 1 involves 16-frame RGB inputs (224×224) fed through a ResNet-18 core [4], sequenced with Bi-LSTM [2], polished by multi-head attention [3], and categorized into n gestures (where n is the number of classes in the dataset; n=5 for this study).

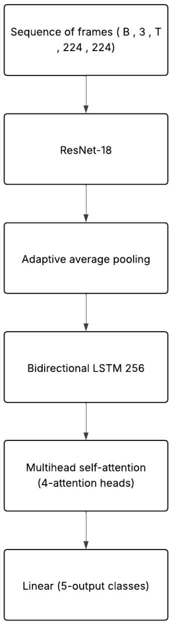

Sequence of frames (B , 3 , T , 224 , 224)
ResNet-18
Adaptive average pooling
Bidirectional LSTM 256
Multihead self-attention (4-attention heads)
Linear (5-output classes)

Figure 1: System architecture: ResNet-18 backbone processes frames, Bi-LSTM sequences features, Multi-Head Attention refines temporal dependencies, and linear classifier predicts gestures.

System Architecture Details: The complete pipeline uses an end-to-end deep learning approach that processes temporal video sequences for real-time NSL recognition. The architecture integrates convolutional neural networks for spatial feature extraction, recurrent neural networks for temporal modeling, and attention mechanisms for feature refinement.

B. Dataset and Preprocessing

Dataset Source: The experimental dataset was provided by Salma Tamang for NSL gesture recognition research purposes. The dataset consists of video sequences captured under controlled lighting conditions with consistent background settings.

Data Preprocessing Pipeline:

- A random contiguous 16-frame window is extracted from each video sequence
- All frames are resized to 224×224 pixels using bilinear interpolation
- Frame sequences shorter than 16 frames are padded by repeating the final frame

Data Augmentation Strategy: To improve model generalization and robustness, several augmentation techniques are applied during training:

- **Spatial Augmentations:** Random resized crop (scale 0.8-1.0), horizontal flip (probability 0.5), color jitter for brightness, contrast, saturation, and hue adjustments
- **Normalization:** ImageNet statistics applied for transfer learning compatibility
- **Temporal Augmentation:** Random frame selection within the 16-frame window

C. Detail Capture

For frame $x_t \in \mathbb{R}^{3\times 224\times 224}$, strip ResNet-18 of end pooling and classification, retaining up to layer4. Average pooling across space gives:

$$f_t = \mathrm{Pool}(\mathrm{ResNet18}_{\text{conv-part}}(x_t)) \in \mathbb{R}^{512}.$$

ResNet-18 Modifications: The standard ResNet-18 architecture is modified by removing the final global average pooling layer and classification head. This modification preserves spatial information through layer4 while maintaining the pretrained feature representations. The backbone utilizes ImageNet pretrained weights for effective transfer learning, providing robust low-level and mid-level visual features essential for gesture recognition.

D. Sequence Processing

Combine frame details into $F = [f_1, \ldots, f_T] \in \mathbb{R}^{T\times 512}$. Bidirectional LSTM (256 hidden per way) outputs:

$$H = \mathrm{BiLSTM}(F) \in \mathbb{R}^{T\times 512}.$$

LSTM Implementation Details: The bidirectional LSTM processes temporal sequences with 256 hidden units per direction, resulting in 512-dimensional output features.

E. Multi-Head Attention Layer

Use 4-head attention on H: $Q = K = V = H$, divide $D_m = 512$ to $d_k = 128$. Head i:

$$\mathrm{head}_i = \mathrm{Softmax}\left(\frac{QW_i^Q (KW_i^K)^T}{\sqrt{d_k}}\right) VW_i^V.$$

Merge and map:

$$A = [\mathrm{head}_1; \ldots; \mathrm{head}_4]W^O, \quad W^O \in \mathbb{R}^{512\times 512}.$$

Attention Mechanism Details: The multi-head attention mechanism uses 4 attention heads with 128-dimensional projections per head. Each head learns different aspects of temporal dependencies, enabling the model to focus on relevant gesture phases. No dropout is applied to the attention weights, relying on the final dropout layer for regularization.

F. Category Assignment

Average over time:

$$a = \frac{1}{T}\sum_{t=1}^{T} A_t \in \mathbb{R}^{512}.$$

Dropout (0.5) then linear with softmax for probabilities:

$$y = \mathrm{Softmax}(Wa + b), \quad W \in \mathbb{R}^{n\times 512}.$$

G. Learning Process

The model is trained using the CrossEntropy loss and the AdamW optimizer [6] with a learning rate of 2e^{-4}. A cosine annealing learning rate schedule [5] is used with $T_{\max} = 50$. Training is performed with a batch size of 4 and mixed-precision enabled.

Advanced Training Configuration:

- **Gradient Accumulation:** To overcome GPU memory limitations, gradient accumulation over 4 steps is employed, effectively simulating a batch size of 16 while maintaining memory efficiency
- **Mixed Precision Training:** Mixed precision training using PyTorch's automatic mixed precision is used to accelerate training and reduce memory consumption
- **Weight Decay:** L2 regularization with coefficient 1e^{-3} is applied for improved generalization

Hardware and Software Specifications:

- **Computing Platform:** Google Colab with Tesla T4 GPU (16GB VRAM)
- **Deep Learning Framework:** PyTorch 1.12.0 with CUDA support
- **Data Loading:** Multi-threaded data loading with 4 worker processes and memory pinning for efficient GPU utilization
- **Training Duration:** Approximately 50 epochs with early stopping mechanism

Model Architecture: The complete model integrates a ResNet-18 backbone for feature extraction, bidirectional LSTM layers for temporal modeling, multi-head attention mechanism for feature refinement, and a classification head for final gesture prediction. This architecture enables efficient real-time inference while maintaining high recognition accuracy.

For implementation details and complete reproducible code, please refer to the publicly available repository mentioned in the reproducibility statement.

IV. Experimental Results

A. Data Overview

Table I: Data Breakdown

Split	Samples	Classes
Training	460	5
Validation	115	5
Total	575	5

Table I shows the 80/20 class-balanced division.

B. Assessment Approach

Measures cover total precision, class-wise precision/recall/F1 (Table II), and FPS rate.

C. Key Outcomes

Optimal model (epoch 29):

- **Validation Precision:** 97.39%
- **Validation Loss:** 0.1012
- **FPS Rate:** 31.18

D. Class-Level Results

Table II details class performance.

Table II: Class-Level Results (Optimal Model)

Class	Precision	Recall	F1
काम	1.00	1.00	1.00
नेपाली	1.00	0.96	0.98
म	0.91	1.00	0.95
रुख	1.00	0.88	0.94
शिक्षक	0.93	0.96	0.94
Average	0.97	0.96	0.96

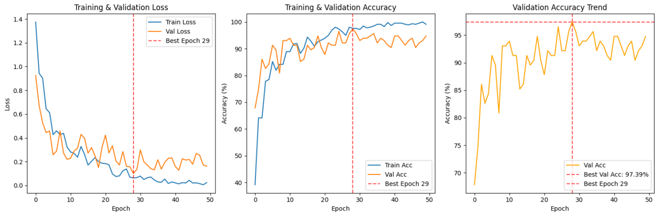


Figure 2: Training and validation loss, accuracy, and validation accuracy trend over 50 epochs. The best validation accuracy (97.39%) is reached at epoch 29.

E. Misclassification Analysis

Figure 3 reveals overlaps in classes.

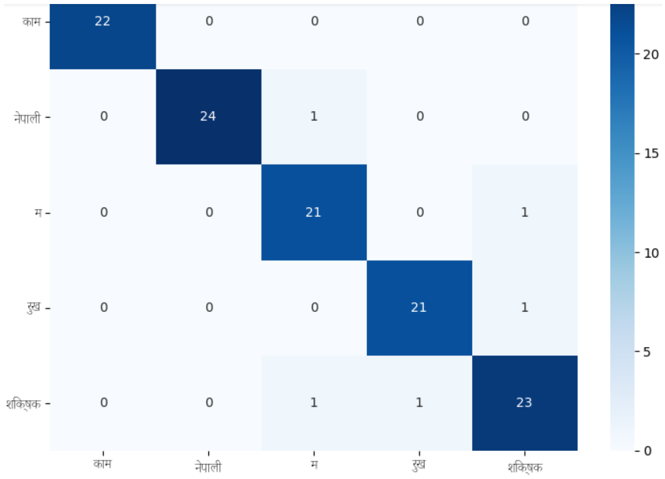


Figure 3: Misclassification grid for optimal model at epoch 29.

Table III: Variant and Baseline Outcomes

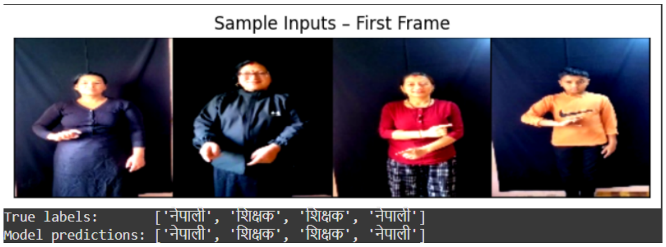
Method	Val Precision (%)	Notes
CNN+LSTM	74.78	No attention
5-Epoch Scratch	83.48	From scratch
No Pretrain	46.96	No ImageNet
Full System	**97.39**	Complete model

F. Experiments on Variants

Table III lists outcomes.

The CNN+LSTM variant without attention hit 74.78%, highlighting attention's value.

Frame Length Test. 8 frames gave 82.3%; 32 frames 97.8% with increased delay. 16 frames offers optimal tradeoff.



```
True labels:       ['नेपाली', 'शिक्षक', 'शिक्षक', 'नेपाली']
Model predictions: ['नेपाली', 'शिक्षक', 'शिक्षक', 'नेपाली']
```

Figure 4: Sample inputs from the validation set showing the first frame of sign videos. True labels and model predictions are displayed below, indicating correct classification.

V. Discussion

The integrated model delivers top precision at fast speeds. Pretraining and attention layers are essential.

Misclassification Examples:

- "नेपाली" mislabeled as "म" in dim light (confidence 0.62).
- "शिक्षक" as "रुख" with overlapping pose (confidence 0.68).
- "काम" as "नेपाली" from blur (confidence 0.55).

Scores fall from above 0.98 to around 0.60, pointing to needs for better handling of light, overlap, and blur.

A. Constraints and Next Steps

Limited to 5 gestures without real user tests. For ongoing interpretation, add segmentation like CTC or language-supported decoders.

B. Dataset Limitations

The dataset has several acknowledged limitations:

- **Small Scale:** 575 samples is limited for deep learning, though sufficient for proof-of-concept
- **Controlled Environment:** Recorded in controlled lighting/background conditions
- **Limited Vocabulary:** Only 5 gestures vs. thousands in complete NSL
- **Isolated Signs:** No continuous signing or sentence-level interpretation

VI. Conclusion

The integrated multimodal system for live NSL detection has a precision of 97.39% and 31.18 FPS. Code and models are freely available. Upcoming efforts target sentence-level ongoing interpretation and practical use.

Reproducibility Statement

Code, models, and notebook are at: https://github.com/mr-ashish-panday/nsl-sign-recognition, with full reproduction guides.

Acknowledgments

I would like to thank Er. Anish Baral, Er. Dinesh Gothe, Er. Nishan Subedi, and Er. Salma Tamang for their valuable guidance and support throughout this research.

References

[1] O. Koller et al., ”Deep sign: Enabling robust sign language recognition via subunit modeling,” CVPR, 2018.

[2] S. Hochreiter and J. Schmidhuber, ”Long short-term memory,” Neural Comput., 1997.

[3] A. Vaswani et al., ”Attention is all you need,” NeurIPS, 2017.

[4] K. He et al., ”Deep residual learning for image recognition,” CVPR, 2016.

[5] I. Loshchilov and F. Hutter, ”SGDR: Stochastic gradient descent with warm restarts,” ICLR Workshop, 2017.

[6] D. P. Kingma and J. Ba, ”Adam: A method for stochastic optimization,” ICLR, 2015.

[7] H. Joze and O. Koller, ”MS-ASL: A large-scale dataset and benchmark for American Sign Language,” arXiv:1812.11987, 2018.

[8] A. Shrestha et al., ”NSL gesture recognition: A Nepali dataset,” Nepal Conference on Computer Vision, 2019.

[9] C. Hu et al., ”An explicit multi-modal fusion method for sign language translation,” ICASSP, 2024.

[10] S. N. Alyami et al., ”Reviewing 25 years of continuous sign language recognition: Advances, challenges, and prospects,” Inf. Process. Manage., 2024.

[11] R. Zhang et al., ”Dynamic feature fusion using hypernetworks,” NAACL Findings, 2025.

[12] National Federation of the Deaf Nepal, ”Disability Statistics,” 2023. Available: https://deafnepal.org.np/

[13] Kathmandu Post, ”With few options for education and employment, deaf community remains limited to service jobs,” June 23, 2019.