Indian Sign Language Recognition using Convolutional Vision Transformers

Ankan Dutta
Computer Science and Engineering Department
National Institute of Technology, Silchar
Assam, India
ankan21_ug@cse.nits.ac.in

Spandan Priyam Chetia
Computer Science and Engineering Department
National Institute of Technology, Silchar
Assam, India
spandan21_ug@cse.nits.ac.in

Mwkthangsa Daimari
Computer Science and Engineering Department
National Institute of Technology, Silchar
Assam, India
mwkthangsa21_ug@cse.nits.ac.in

Suganya Devi K
Computer Science and Engineering Department
National Institute of Technology, Silchar
Assam, India
suganya@cse.nits.ac.in

***Abstract*—For the deaf community in India, the Indian Sign Language (ISL) is an essential communication tool. In this work, we propose a recognition system based on Convolutional Vision Transformers (CVT) that leverages both the global contextual modeling of Vision Transformers (ViT) and the local feature extraction capabilities of Convolutional Neural Networks (CNN). To address challenges such as small datasets and class imbalance, our system employs extensive data augmentation, color space translation, and robust training techniques. Experimental results indicate that our CVT-based approach achieves an accuracy of 91.14% in the ISL dataset, outperforming both CNN-based ResNet and pure ViT models. The research results highlight the potential of the CVT model as a promising tool for advancing ISL recognition research.**

***Index Terms*—Indian Sign Language, Convolutional Neural Networks, Vision Transformer, Convolutional Vision Transformers, Data Augmentation.**

I. INTRODUCTION

Sign languages serve as a vital mode of communication for individuals with hearing and speech impairments, enabling them to convey thoughts and emotions through hand gestures, facial expressions, and body movements. Among various sign languages, Indian Sign Language (ISL) is widely used by the deaf and hard-of-hearing community in India. However, despite its importance, ISL remains under-researched and lacks widespread digital support, making communication challenging for the deaf community in India.

This paper addresses the challenges of ISL recognition, a crucial tool for the deaf community in India. The study proposes a recognition system using Convolutional Vision Transformers (CVT), which leverage both global contextual modeling and local feature extraction. This paper aims to bridge these gaps by presenting an ISL recognition framework leveraging deep learning and computer vision techniques. The proposed model utilizes MediaPipe HandLandmarker to detect hand landmarks and applies a pre-trained CVT-13 model for robust gesture classification. By combining edge enhancement techniques and skeleton-based landmark representation, the model improves accuracy while maintaining computational efficiency.

The primary research objectives are as follows:

- Develop an effective and precise ISL recognition system using CvT.
- To leverage CvT's capacity to capture both spatial and hierarchical features for improved gesture recognition.
- Improve understanding of the strengths and limitations of hybrid models in the context of ISL.
- Contribute theoretical insights that can guide future research in sign language recognition and related assistive technologies.

The paper is structured as follows: Section 2 reviews related work, Section 3 details the methodology, Section 4 presents results, and Section 5 concludes with future directions.

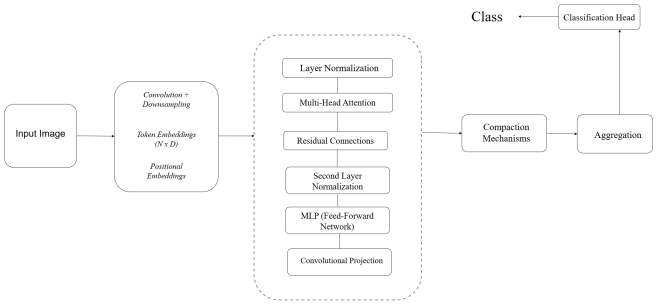

Fig. 1: Convolutional Vision Transformer model summary.

II. RELATED STUDY

Recent research on sign language recognition has explored a variety of deep learning techniques.

- **CNN-based Approaches:** The use of Convolutional Neural Networks (CNNs) for sign language recognition has been the subject of numerous studies. Rahman et al. [5], in particular, set a new standard for American Sign Language (ASL) recognition with their innovative CNN model. Through a thorough preprocessing pipeline that included rigorous data augmentation (such as rotations, scaling, and shifting) and normalization to address issues like limited training samples and class imbalance, they trained a deep CNN on four publicly available ASL datasets that covered both alphabets and numerals. In order to automatically learn hierarchical and discriminative feature representations, the suggested architecture consists of many convolutional layers with batch normalization and ReLU activation functions. In order to reduce overfitting, dropout layers were also used, which eventually improved recognition accuracy by about 9% compared to earlier techniques. Their assessment, which included cross-dataset testing and k-fold cross-validation, further demonstrated the method's generalizability and robustness. Furthermore, by combining CNNs with Recurrent Neural Networks (RNNs) to capture temporal dynamics, Masood et al. [6] expanded the framework to real-time gesture recognition, demonstrating the potential of CNN-based techniques to efficiently extract salient local features and model sequential dependencies.
- **Hybrid and SVM-based Systems:** There have been suggestions for hybrid techniques to address some of CNNs' drawbacks. For instance, Katoch et al. [10] created a system for recognizing Indian sign language that combines CNNs and Support Vector Machines (SVMs) with SURF feature extraction. To overcome issues like backdrop reliance and uneven hand segmentation, they use SURF to extract robust local descriptors that are invariant to rotation, scaling, and partial occlusion. Histograms of visual words are created to represent each image once the extracted SURF characteristics are quantized into a visual vocabulary. An SVM classifier uses these histograms as input to give a preliminary framework for recognition. Concurrently, a CNN learns hierarchical, discriminative features from the raw data in order to further enhance the feature representation. In comparison to conventional techniques, the system achieves improved recognition accuracy with the combined use of CNN for deep feature learning, SVM for efficient classification with limited data, and SURF for quick and reliable feature extraction. The advantages of combining several approaches to address the intricate heterogeneity present in Indian sign language are highlighted by this hybrid approach.
- **Transformer-based Methods:** The application of Transformer designs to sign language recognition has been examined in more recent research. Transformer networks, for instance, were used by De Coster et al. [8] to extract long-range dependencies from sign language data. Their method successfully models the intricate spatial and temporal dynamics seen in sign language by utilizing a multi-head self-attention mechanism with residual connections. In their tests on the Flemish Sign Language corpus, the suggested transformer-based approach showed promise as a tool for speeding up corpus annotation, achieving an accuracy of 74.7% on a vocabulary of 100 classes. Shin et al. [7] presented a transformer-based deep neural network specifically for Korean sign language recognition. The method is developed in Shin et al.'s work utilizing a multi-branch architecture that applies a convolutional layer-based transformer block after a "grain module" that uses successive 3×3 convolutions to minimize the input size and create a patch-like representation. This block efficiently captures both local and global dependencies by utilizing lightweight multi-head self-attention (LMHSA) with relative position biases. Lastly, a classification module that uses fully connected layers and global average pooling is used to aggregate the retrieved features. By combining local and global context, these improvements not only get over patch-only transformers' limits in maintaining local structure but also boost recognition performance overall.
- **Transformers for Translation:** Furthermore, Yin and Read [9] advanced sign language translation by utilizing transformer designs. Their study achieved notable improvements over earlier approaches by introducing an end-to-end SLT system that integrates a Transformer network with Spatial-Temporal Multi-Cue (STMC) characteristics. When tested on the PHOENIX-Weather 2014T and ASLG-PC12 datasets, they specifically indicated increases of more than 5 BLEU on ground truth glosses and 7–17 BLEU on forecasted glosses. To efficiently translate gloss sequences into spoken English, their system combines a two-layer Transformer with cutting-edge training methods like weight tying, transfer learning, and ensemble decoding. The results of this study show that transformer-based methods are effective not just at classification tasks but also at accurately translating sign language into natural language.
  - **Advancements in Sign Language Recognition Using Transformers** Sign language recognition (SLR) has seen significant advancements with the integration of deep learning and transformer-based architectures. Traditional approaches relied on convolutional neural networks and recurrent models, but recent studies demonstrate that Vision Transformers and hybrid models, such as Convolutional Vision Transformers, enhance feature extraction and temporal modeling [4]. Camgoz et al. introduced a transformer-based approach for end-to-end sign language recognition and translation [2], while SignFormer leverages deep vision transformers for improved sequence learning [3]. These innovations address challenges in large-scale SLR datasets by improving spatial-temporal representations, paving the way for more robust and real-time sign language understanding systems [1].

The evolution from traditional CNN methods to hybrid and Transformer-based models illustrates the research community's ongoing efforts to capture both local and global information in sign language data. Our proposed CvT model is designed to build on these insights by integrating the best of both worlds.

III. METHODOLOGY

Our methodology focuses on a CvT-based approach for ISL recognition, emphasizing research aspects such as data augmentation, image normalization, and optimal training parameter selection.

Convolutional Vision Transformer (CvT) is a hybrid model that combines the strengths of Convolutional Neural Networks (CNNs) and Vision Transformers (ViTs) for better image representation learning. It improves upon standard ViTs by incorporating convolutions in the tokenization and projection layers, enhancing spatial locality, translation invariance, and efficiency.

The CvT architecture follows a multi-stage hierarchical structure, where each stage consists of:

- **Convolutional Token Embedding** – Extracts feature patches using convolutional layers instead of linear projections.
- **Hierarchical Transformer Blocks** – Applies self-attention with convolutional projections to refine feature representations.
- **Classification Head** – Generates the final output prediction using global pooling and a fully connected layer.

CvT improves Vision Transformers (ViTs) by integrating convolutional layers for better spatial feature extraction and efficiency. It follows a hierarchical structure with three main components: 1. Convolutional Token Embedding

- Replaces ViT's fixed patch embedding with convolutional tokenization to retain local spatial features.
- Uses overlapping convolutions to generate tokens instead of rigid, non-overlapping patches.

2. Hierarchical Transformer Blocks

- Each stage consists of multiple transformer layers, including:
- Convolutional Projection Attention (CPA): Applies depthwise separable convolutions before self-attention to improve locality and efficiency.
- Feed-Forward Network (FFN): A two-layer MLP with GELU activation for feature transformation.
- Layer Normalization & Skip Connections: Ensures stable training and prevents gradient vanishing.

3. Multi-Stage Feature Extraction

- Progressively reduces resolution while increasing feature depth, similar to CNNs.
- Early layers capture low-level details (edges, textures), while deeper layers learn high-level features (shapes, objects).

4. Classification Head

- Uses Global Average Pooling (GAP) to reduce features to a single vector.
- A Fully Connected (FC) layer maps the extracted features to the final output classes.

Figure 2 shows architecture of Convolutional Vision Transformer.

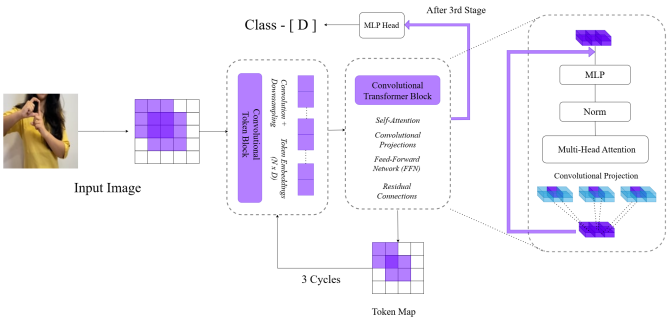

Fig. 2: Architecture of Convolutional Vision Transformer.

A. Dataset Details

Data was collected from multiple sources to ensure diversity and robustness:

- **Phase 1:** Data was obtained from Hugging Face and Kaggle, ensuring a variety of hand gestures, lighting conditions, and orientations. Additional samples were collected using a webcam, and each ISL alphabet (A-Z) had between 30 and 50 samples.
- **Phase 2:** The data set was expanded to include word-level classes (e.g., "you," "who," "agree"), with each new class containing approximately 50 samples sourced from Mendeley Data, Roboflow, and Kaggle.

For clarity, the file counts for our datasets are summarized in Table **??**.

Figure 8 shows a random grid of the collected hand gesture data.

B. Data Splitting

To evaluate the model effectively, the dataset was split into training, validation, and test sets. Initially, 20% of the total dataset was reserved as the test set. The remaining 80% was further divided into training and validation sets, with 80% allocated for training and 20% for validation.

Thus, the final distribution was as follows:

- **Test Set**: 20% of the total dataset
- **Training Set**: 64% of the total dataset (80% of the remaining 80%)
- **Validation Set**: 16% of the total dataset (20% of the remaining 80%)

This setup ensures that the model is trained and validated without exposure to the test data, which is used solely for final evaluation.

TABLE I: Dataset File Counts (Words)

Dataset Name	Number of Files
afraid	153
agree	154
assistance	109
bad	80
become	80
college	80
doctor	80
from	80
pain	80
pray	80
secondary	80
skin	80
smalls	80
specific	80
stand	80
today	80
warn	80
which	80
work	80
you	80

TABLE II: Dataset File Counts (Letters)

Dataset Name	Number of Files
A	25
B	38
C	34
D	34
E	46
F	21
G	29
I	30
K	36
L	36
M	26
N	24
O	38
P	30
Q	38
R	26
S	29
T	31
U	27
V	25
W	28
X	24
Z	27

Fig. 3: Random grid of collected hand gesture data.

C. System Configuration and Training Environment

All experiments were conducted on a system with the following specifications:

- **Processor:** Intel Xeon CPU @2.20Hz .
- **RAM:** 29 GB
- **GPU:** NVIDIA T4 * 2 ,4 CPU cores.
- **Operating System:** Windows 11 .
- **Programming Language:** Python 3.8+
- **Libraries and Frameworks:** OpenCV,Pytorch Lightning, Scilit Learn,Mediapipe Hand Landmarks.

Model training and evaluation were performed using Jupyter Notebook / Google Colab / VS Code (choose the platform you used). Random seeds were set where applicable to ensure reproducibility of results.

D. Evaluation Metrics

We evaluate our system using the following metrics:

1) **Accuracy (Multiclass):** Measures the proportion of correctly classified instances out of total instances.

$$\text{Accuracy} = \frac{\sum_{c=1}^{C}(TP_c + FP_c + FN_c)}{\sum_{c=1}^{C} TP_c}$$

2) **Weighted Precision:** Averages precision across all classes, weighted by the number of instances in each class.

$$\text{Weighted Precision} = \frac{\sum_{c=1}^{C} N_c \cdot \text{Precision}_c}{\sum_{c=1}^{C} N_c}$$

3) **Weighted Recall:** Averages recall across all classes, weighted by the number of instances in each class.

$$\text{Weighted Recall} = \frac{\sum_{c=1}^{C} N_c \cdot \text{Recall}_c}{\sum_{c=1}^{C} N_c}$$

4) **Weighted F1 Score:** Averages the F1 score across all classes, weighted by the number of instances in each class.

$$\text{Weighted F1 Score} = \frac{\sum_{c=1}^{C} N_c \cdot \text{F1}_c}{\sum_{c=1}^{C} N_c}$$

where C is the total number of classes, N_c is the number of true instances in class c, and Precision_c, Recall_c, and F1_c are the respective metrics for class c.

IV. RESULTS AND DISCUSSION

A. Data Preprocessing

Data preprocessing is divided into two stages:

Basic Preprocessing (Phase 1):

- **Resizing:** Standardize all images to 224 × 224 pixels.
- **Data Augmentation:** Enhance robustness using color jittering (brightness, contrast, etc.) and flipping (horizontal & vertical).
- **Batch size:** 64 images and labels.

Advanced Preprocessing (Phase 2):

- **Sharpening Without Mask:** Enhance edges and hand details with an unsharp mask to improve landmark detection.
- **Generating Landmarks:** Use MediaPipe HandLandmarker to extract key points from the hand and generate a skeleton image using OpenCV.

Figure 5 shows a Skeleton representation generated after applying landmark transformation.

Fig. 4: Before landmark transform.

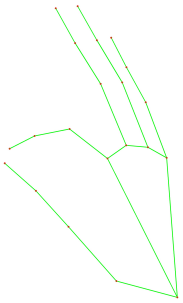

Fig. 5: After landmark transform.

Before Sharpen

After Sharpen

Fig. 6: Comparison of images before and after sharpening.

B. Model Setup and Training Process

Model Initialization:

- **Output Classes:** 26.
- **Base Model:** Pre-trained `microsoft/cvt-13` from the Hugging Face Model Hub.
- **Learning Rate:** 1e-4.

Training Step:

- **Batch Size:** 64 images and labels.
- **Process:**
  1) Data is passed through the model to generate logits.
  2) Cross-entropy loss is computed between the logits and true labels.

Optimization:

- **Optimizer:** AdamW with a learning rate of 1e-4.

C. Training Enhancements and Configuration

- **Callbacks:**
  - **Model Checkpointing:** Save the top 2 models (lowest validation loss) every 50 training steps.
  - **Early Stopping:** Halt training after 3 epochs without improvement in validation loss.
- **Trainer Configuration:**
  - **GPU:** Single Nvidia T4.
  - **Epochs:** Maximum of 30.

Figure 7 and 8 shows a loss learning curve and accuracy learning curve graph.

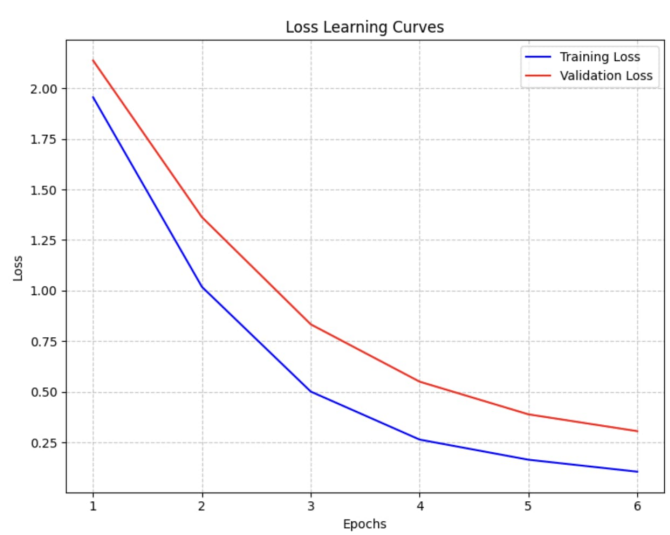

Fig. 7: Loss graph.

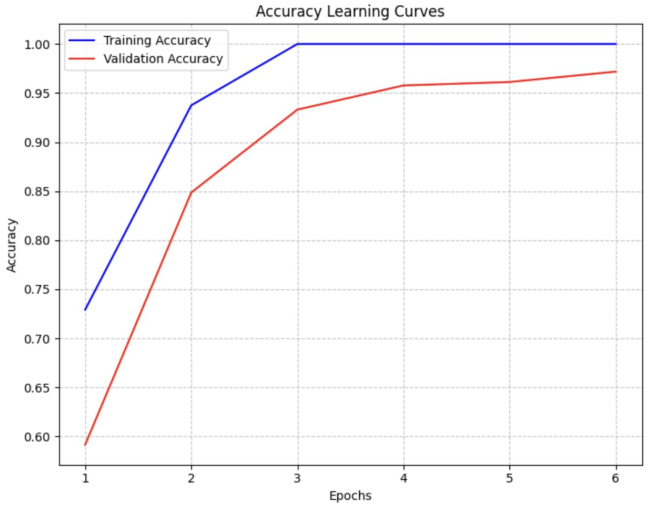

Fig. 8: Accuracy graph.

D. Model Comparison (Phase 1)

In Phase 1 (Alphabet Recognition), the dataset includes classes for all alphabets (A-Z) and digits (0-9). The models were trained on 80% of the data and evaluated on the remaining 20%. Table III shows the performance metrics for various models.

E. Ablation Study

Additional word-level classes (such as "you," "who," "agree") were included in Phase 2 (Word and Alphabet Classification).In these table , we have compare with letter and word dataset combined. There were two runs: one with raw photos and another with landmark extraction and filtering. The performance metrics are summarized in Table IV.

TABLE III: Performance Metrics for CvT, ResNet, ViT, InceptionV3, and VGG16 Models in ISL Recognition (Alphabet Level)

Metric	CvT	ResNet	ViT	InceptionV3	VGG16
Accuracy	91.14%	81.33%	90.00%	84.10%	82.91%
Precision	92.17%	81.92%	91.85%	85.62%	83.44%
Recall	91.14%	81.33%	90.00%	84.10%	82.91%
F1-score	90.92%	80.50%	89.45%	83.85%	81.68%

TABLE IV: Performance Metrics for Base Data and Filtered + Landmarks Data (Phase 2)

Metric	Base Data Performance	Filtered + Landmarks Performance
Accuracy	83.03%	88.35%
Precision	83.06%	88.69%
Recall	83.03%	88.35%
F1-score	83.01%	88.27%

F. Discussion

The experimental results reveal that the CvT model significantly improves the recognition accuracy compared to conventional CNN-based methods. By effectively integrating spatial and contextual features, the CvT model demonstrates a research-worthy balance between local feature extraction and global context modeling. The results of Phase 2 also highlight the impact of advanced pre-processing (landmark extraction and filtering) on overall performance. Despite these promising findings, challenges such as limited dataset size and class imbalance remain open research questions that warrant further exploration.

V. CONCLUSION AND FUTURE WORK

This paper presents a research-oriented exploration of ISL recognition using a Convolutional Vision Transformer (CvT) framework. Our work contributes to understanding how hybrid models can capture both local and global features effectively. The study demonstrates that careful design of data augmentation, pre-processing, and model training can lead to significant performance improvements. Future research should focus on addressing open challenges such as data scarcity and class imbalance, as well as exploring novel hybrid architectures and advanced data augmentation techniques to further enhance recognition performance.

Future Research Directions:

- Investigate advanced data augmentation and synthetic data generation to alleviate dataset limitations.
- Explore novel hybrid architectures that further integrate local and global feature extraction.
- Conduct in-depth theoretical analyses to understand the contributions of each model component to ISL recognition.

REFERENCES

[1] N. M. Alharthi and S. M. Alzahrani. *Vision transformers and transfer learning approaches for Arabic sign language recognition.* Applied Sciences, 13(21), 2023.

[2] N. C. Camgoz, O. Koller, S. Hadfield, and R. Bowden. *Sign language transformers: Joint end-to-end sign language recognition and translation.* In Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), June 2020.

[3] Deep Kothadiya, Chintan Bhatt, Tanzila Saba, and Amjad Rehman. *Signformer: deepvision transformer for sign language recognition.* IEEE Access, PP:1–1, January 2023.

[4] Oscar Koller, Sepehr Zargaran, Hermann Ney, and Richard Bowden. *Deep sign: Enabling robust statistical continuous sign language recognition via hybrid CNN-HMMs.* International Journal of Computer Vision, 126, December 2018.

[5] Md Moklesur Rahman, Md Islam, Md. Hafizur Rahman, Roberto Sassi, Massimo Rivolta, and Md Aktaruzzaman. *A new benchmark on American sign language recognition using convolutional neural network.* April 2020.

[6] Sarfaraz Masood, Adhyan Srivastava, Harish Thuwal, and Musheer Ahmad. *Real-Time Sign Language Gesture (Word) Recognition from Video Sequences Using CNN and RNN.* Pages 623–632, January 2018.

[7] Jungpil Shin, Abu Saleh Musa Miah, Md. Al Mehedi Hasan, Koki Hirooka, Kota Suzuki, Hyoun-Sup Lee, and Si-Woong Jang. *Korean sign language recognition using transformer-based deep neural network.* Applied Sciences, 13(5), 2023.

[8] Mathieu De Coster, Mieke Van Herreweghe, and Joni Dambre. *Sign language recognition with transformer networks.* In Proc. Twelfth Language Resources and Evaluation Conference, pages 6018–6024, Marseille, France, May 2020.

[9] Kayo Yin and Jesse Read. *Attention is all you sign: Sign language translation with transformers.* 2020.

[10] Shagun Katoch, Varsha Singh, and Uma Shanker Tiwary. *Indian sign language recognition system using SURF with SVM and CNN.* Array, 14:100141, April 2022.