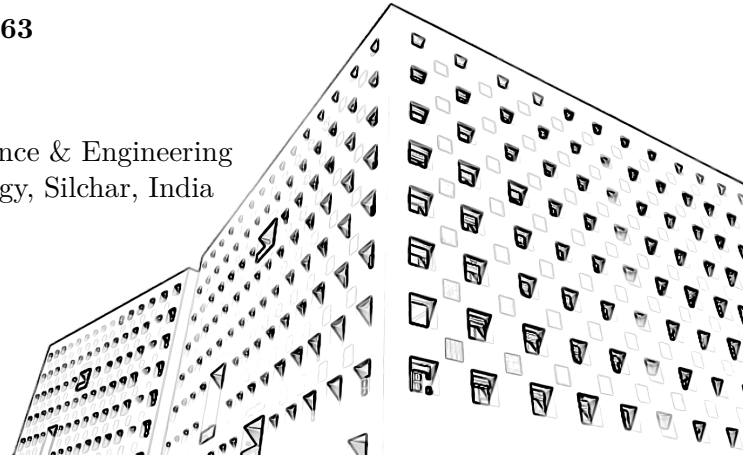
Indian Sign Language Recognition using Convolutional Vision Transformers

ICNSoC-2025 Paper ID: 663

Presented By:
Ankan Dutta
Department of Computer Science & Engineering
National Institute of Technology, Silchar, India

Co-authors:
Spandan Priyam Chetia
Mwkthangsa Daimari
Satyajit Swain
Suganya Devi K



OUTLINE

1 Introduction

Indian Sign Language

1 Introduction

Sign language is the primary mode of communication for deaf and hard-of-hearing individuals.

Indian Sign Language (ISL) is widely used by the Indian Deaf community, but lacks substantial computational and linguistic resources.

ISL consists of unique hand shapes that employ both hands for the alphabets and are different from other standard languages like American Sign Language (ASL).

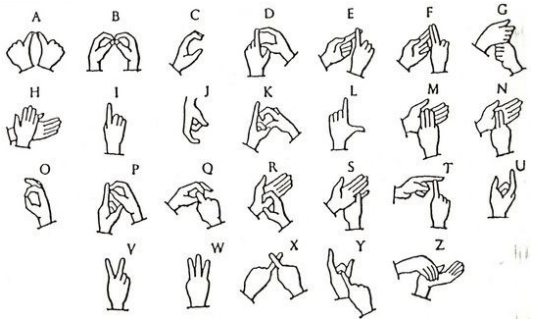


Figure 1.1:Indian Sign Language

Sign Language Recognition Overview

1 Introduction

Sign language recognition (SLR) is a challenging task that involves translating hand gestures into text or spoken language.

Current state-of-the-art methods often employ deep learning models, including Convolutional Neural Networks (CNN) for feature extraction and Recurrent Neural Networks (RNN) or Hidden Markov Models (HMM) for sequence modeling.

This work presents a hybrid model that combines CNN for hand feature extraction and Transformers for capturing long-range dependencies in sequential data, improving the system's ability to recognize continuous sign language.

OUTLINE

2 Motivation

Need for ISL

2 Motivation

ISL is crucial to enable communication between individuals with hearing and speech impairments. It is estimated that ISL is used by at least 15 million deaf signers in India. Despite its widespread use, ISL remains underrepresented in technological solutions.

ISL is a fully developed language with its own grammar, syntax, and regional variations, making it as complex and rich as spoken languages. This complexity poses challenges for computational recognition and underscores the need for advanced assistive technologies.

Bridging this gap is essential to promote inclusivity in education, employment, and daily life. By advancing assistive technologies, this work promotes accessibility and inclusion, contributing to a more equitable society.

Research Questions

1. How to design a robust system that achieves competitive accuracy with hand-only input?
2. Can transformers be leveraged to better model the sequential nature of sign language, thereby improving the system's performance on challenging datasets?

OUTLINE

3 Related Works

Related Works

3 Related Works

Step	Paper	Methodology	Findings and Takeaways
1	Vision Transformers and Transfer Learning Approaches for Arabic Sign Language Recognition. Applied Sciences.[1]	The paper investigates two primary approaches for Arabic Sign Language (ArSL) recognition: 1. Transfer learning using pretrained deep learning models (e.g., VGG, ResNet, MobileNet, Xception, Inception, DenseNet, BiT, ViT, Swin) 2. Deep learning approaches using CNN architectures developed from scratch. The study evaluates these models on a dataset containing 54,049 images of 32 Arabic alphabets	The study highlights the effectiveness of transfer learning combined with vision transformers for Arabic Sign Language recognition, demonstrating superior performance over CNN-based models. Transfer learning approaches, particularly those utilizing pretrained models, are more robust and efficient for sign language classification tasks.

Step	Paper	Methodology	Findings and Takeaways
2	Sign Language Transformers: Joint End-to-End Sign Language Recognition and Translation.[2]	The authors propose a transformer-based architecture that jointly learns Continuous Sign Language Recognition (CSLR) and Translation (SLT) using Connectionist Temporal Classification (CTC) loss. This end-to-end model combines two components: 1. Sign Language Recognition Transformer (SLRT) for predicting gloss sequences 2. Sign Language Translation Transformer (SLTT) for generating spoken language translations The system learns from sign language videos without requiring intermediate steps or ground-truth alignment, and is evaluated on the RWTH-PHOENIX-Weather-2014T dataset, achieving state-of-the-art performance in both tasks.	1. Proposes a unified model for simultaneous recognition and translation. 2. Outperforms previous models, doubling BLEU-4 scores for some tasks.(e.g., 21.80, compared to 9.58 from previous models in the Sign2Text task.)

Step	Paper	Methodology	Findings and Takeaways
3	SIGNFORMER: DeepVision Transformer for Sign Language Recognition.[3]	The authors propose a transformer-based architecture for static Indian sign language recognition. The methodology uses multi-head self-attention in the encoder phase, dividing sign language images into patches, which are passed through 6 transformer encoder layers. The classification is handled by a multi-layer perceptron (MLP) with data augmentation applied to the input images.	Achieves a recognition accuracy of 99.29% on static sign language using just 5 training epochs. The model is efficient with fewer layers and performs well under various augmentation methods, demonstrating high accuracy with a minimal training process. This approach can potentially be extended to recognize continuous sign language.

Step	Paper	Methodology	Findings and Takeaways
4	Deep Sign: Enabling Robust Statistical Continuous Sign Language Recognition via Hybrid CNN-HMMs.[4]	The authors used a hybrid CNN-HMM approach to recognize continuous sign language. They: 1. Preprocessed video data by tracking the signer' s hand. Combined CNN for feature extraction with HMM for sequence modeling. 2. Pretrained the CNN on ImageNet and fine-tuned on sign language datasets. 3. Evaluated performance on three datasets using Word Error Rate (WER). 4. Optimized the model through iterations and experimented with different CNN architectures (GoogLeNet, AlexNet, and LeNet).	1. The hybrid CNN-HMM approach significantly reduces WER. GoogLeNet outperformed other architectures. 2. Pretraining on external data improved model accuracy. 3. Further improvements can come from including non-manual features and exploring end-to-end training.

Step	Paper	Methodology	Findings and Takeaways
5	A New Benchmark on American Sign Language Recognition using Convolutional Neural Network. [5]	1. **Data Pre-processing**: Grayscale conversion, normalization, and resizing of images to 64×64 pixels. 2. **Model**: CNN (SLRNet-8) with 6 convolution layers, 3 pooling layers, ReLU activation, batch normalization, max-pooling (2×2), Global Average Pooling (GAP), and dropout (0.5 probability). 3. **Training**: Data augmentation (random rotation, zooming, shifting), Adam optimizer (learning rate: 0.001), 10-fold cross-validation, and early stopping (30 epochs). 4. **Testing**: 99.9% accuracy on four ASL datasets with separate training and testing sets.	1. **High Accuracy**: The CNN model (SLRNet-8) achieved 99.9 2. **Efficient Pre-processing**: Grayscale conversion and image resizing allowed faster training without sacrificing recognition performance. 3. **Robust Architecture**: The use of ReLU, max-pooling, and dropout helped reduce overfitting and improved generalization. 4. **Data Augmentation**: Techniques like rotation and zooming improved model robustness, allowing it to handle variability in input images effectively.

Step	Paper	Methodology	Findings and Takeaways
6	Real-Time Sign Language Gesture (Word) Recognition from Video Sequences Using CNN and RNN. [6]	The authors utilised a dataset of Argentinean Sign Language (46 gesture categories). Two approaches were used to train the model on the temporal and the spatial features, and both differ by the way inputs given to RNN to train it on the temporal features. 1. **Prediction Approach**: This approach captures temporal dependencies by treating each frame as a sequential step, enhancing recognition accuracy through contextual learning. 2. **Pool Layer Approach**: This method uses a pooling mechanism to aggregate spatial features extracted by Convolutional Neural Networks (CNNs) from video frames.	1. Achieved a recognition accuracy of 95.2%, demonstrating the effectiveness of the combined CNN and RNN approach. 2. Successfully handles the complexities of sign language, including hand gestures, facial expressions, and body language, while overcoming traditional RNN limitations with LSTM units. 3. Highlights the potential of deep learning technologies to enhance accessibility for the deaf and hard-of-hearing communities through automated sign language translation.

Step	Paper	Methodology	Findings and Takeaways
7	Indian Sign Language recognition system using SURF with SVM and CNN. [7]	The authors propose an Indian Sign Language (ISL) recognition system using: 1. SURF (Speeded Up Robust Features) for feature extraction. 2. Bag of Visual Words (BOVW) model to map features to sign language alphabets and digits. 3. SVM and CNN for classification. 4. A custom dataset with 36,000 images was created from videos captured via webcam, and the system output labels in both text and speech.	1. **Hybrid Approach**: Combining SURF with SVM and CNN offers a robust and efficient ISL recognition method. 2. **High Accuracy**: The system achieves over 99% accuracy for both alphabets and digits. 3. **Custom Dataset**: No standardized ISL dataset was available, so a custom one was developed for the project.

Step	Paper	Methodology	Findings and Takeaways
8	Korean Sign Language Recognition Using Transformer-Based Deep Neural Network. [8]	1. A multi-branch architecture combining Convolutional Neural Networks (CNNs) and Transformer-based deep neural networks. 2. Use of a KSL dataset and a lab-created dataset with 77 Korean sign language labels. 3. Multi-branch approach with CNN for local feature extraction and Transformer for long-range dependency extraction	1. Achieved 89% accuracy on the KSL dataset and 98.3% accuracy on the lab dataset. 2. Successfully integrated CNN and Transformer to address issues of illumination and background complexity. 3. Proposed a novel grain module to reduce image size and enhance feature extraction efficiency.

Step	Paper	Methodology	Findings and Takeaways
9	Sign Language Recognition with Transformer Networks. [9]	The authors applied Transformer networks, utilizing multi-head attention mechanisms for isolated sign language recognition. They used OpenPose for human keypoint estimation and combined it with end-to-end feature learning via Convolutional Neural Networks (CNNs). Four methods were evaluated: 1. PoseLSTM 2. Pose Transformer Network (PTN), 3. Video Transformer Network (VTN) 4. Multimodal Transformer Network (MTN). The dataset used was the Flemish Sign Language (VGT) corpus, which included 100 classes and 18,730 samples.	1. The MTN method outperformed others, achieving 74.7% accuracy on the test set for a vocabulary of 100 classes. 2. Transformer networks were more effective than LSTMs for isolated SLR. 3. Adding keypoints from OpenPose improves accuracy when combined with learned features from CNNs. 4. The results can contribute to developing suggestion tools to aid sign language corpus annotation.

Step	Paper	Methodology	Findings and Takeaways
10	Attention is All You Sign: Sign Language Translation with Transformers. [10]	1. Transformer-based approach combined with the Spatial-Temporal Multi-Cue (STMC) network for end-to-end Sign Language Translation (SLT). 2. Explores Gloss2Text and Sign2Gloss2Text tasks. 3. Implements weight tying, transfer learning, and ensemble learning. 4. Experiments conducted on PHOENIX-Weather 2014T and ASLG-PC12 datasets.	1. Achieved state-of-the-art BLEU score improvements: 5+ on ground truth glosses and 7+ on predicted glosses for PHOENIX dataset. 2. End-to-end translation using predicted glosses outperforms ground truth glosses. 3. Introduced novel STMC-Transformer architecture, suggesting further improvements with joint training of SLR and translation systems.

OUTLINE

4 Addressable Research Gaps

Improved Scalability

4 Addressable Research Gaps

- **Gap:** In CNN models, performance saturates beyond a certain model size.
- **Solution:** Convolutional Vision Transformers (CVT) benefit from the scalability of transformers, enabling the handling of extremely large models and datasets (e.g., JFT-300M).

Performance on Smaller Datasets

4 Addressable Research Gaps

- **Gap:** Vision Transformers (ViT) underperform on small datasets due to lack of inductive biases.
- **Solution:** CVT incorporate CNN-like biases (locality and hierarchy), achieving competitive results with less data.

Positional Encoding Inefficiency

4 Addressable Research Gaps

- **Gap:** ViT rely on positional encodings, which are inefficient for variable resolutions.
- **Solution:** CVT embed spatial relationships directly via convolutions, eliminating the need for positional encodings.

OUTLINE

5 Objectives

Objectives

5 Objectives

The primary objective of this research is to develop an efficient and accurate recognition system for ISL using CVT. The study seeks to:

- Leverage CVT ability to capture spatial and hierarchical features for improved gesture recognition.
- Enhance the accuracy and robustness of ISL recognition systems under diverse conditions.
- Contribute to the development of assistive technologies that promote inclusivity and accessibility for individuals with hearing and speech impairments.

OUTLINE

6 Methodology

Dataset Collection

6 Methodology

Phase 1: Character-Level Data

- Dataset curated using Kaggle and Hugging Face repositories.
- Additional samples captured using laptop webcam.
- Each alphabet (A-Z) has 30–50 images, totaling hundreds of samples for varied backgrounds and lighting.

Phase 2: Word-Level Data

- Based on gestures referenced from the official ISL portal.
- Includes 150 frequently used ISL words (e.g., *you*, *from*, *agree*, *warn*).
- Each word class contains 30–50 samples sourced from Kaggle, Roboflow, Mendeley Data, and webcam captures.

Dataset Structure and Format

6 Methodology

- The dataset is organized into:
  - **Training Set (70%)**: Used for training the model.
  - **Validation Set (20%)**: Used for tuning the model.
  - **Testing Set (10%)**: Used to evaluate generalization.
- Image Format: All samples are resized to 224×224 pixels and stored in JPEG format.
- Image variation includes lighting, hand orientations, and gesture articulation.

Dataset Preprocessing

6 Methodology

Basic Preprocessing:

- Resize images to 224×224 pixels.
- Normalize pixel values for consistency.

Advanced Augmentation Techniques:

- **Color:** Brightness, contrast, hue, saturation, CLAHE, channel shuffling.
- **Noise/Blur:** Gaussian noise, motion blur, ISO noise.
- **Geometric:** Random shifts, scaling, rotations, perspective warp.
- **Lens Simulation:** Optical distortion to mimic real-camera anomalies.

Example of Collected Dataset

Figure 6.1:Example of Collected ISL Gesture Samples

Table I: File Count for ISL Letters and Digits

Label	Count	Label	Count	Label	Count
1	220	2	225	3	220
4	215	5	210	6	240
7	230	8	200	9	200
A	210	B	230	C	225
D	220	E	240	F	210
G	235	H	225	I	220
J	215	K	215	L	240
M	210	N	225	O	230
P	220	Q	225	R	220
S	230	T	240	U	220
V	220	W	225	X	220
Y	230	Z	220		

Dataset Summary

6 Methodology

Table II: Word-Level Class Distribution

Word	Count	Word	Count	Word	Count	Word	Count
afraid	220	pray	220	pain	200	about	210
agree	225	secondary	230	up	220	always	205
assistance	210	skin	215	present	200	again	215
bad	200	small	205	warn	200	better	225
become	220	specific	210	spouse	205	bring	210
college	200	stand	240	wear	210	between	220
doctor	200	teach	230	welcome	215	before	200
from	200	there	240	wish	225	beyond	205
ghost	215	group	220	with	220	behind	210
mother	200	today	200	without	230	beside	200
old	200	toward	220	work	220	across	215
you	200	above	210	arrive	205	adapt	200
again	210	afford	205	allow	215	borrow	200

Table II: Word-Level Class Distribution (cont.)

Word	Count	Word	Count	Word	Count	Word	Count
participate	220	permit	205	prefer	200	recall	205
reduce	210	remove	200	reply	205	remain	200
share	215	steal	200	suppose	205	survive	210
treat	200	unite	205	upload	200	chase	205
divide	200	dream	210	escape	205	invent	200
isolate	210	justify	205	locate	200	oppose	205
repair	200	revive	210	salute	205	survive	210
travel	200	erode	205	evolve	210	rejoice	200
venture	205	wonder	200	witness	205	yield	200
catch	220	choose	210	doubt	205	enjoy	200
enter	200	excuse	205	follow	210	guide	200
happen	215	include	210	laugh	200	manage	205
offer	200	measure	215	notice	200	occupy	210

Model Architecture

6 Methodology

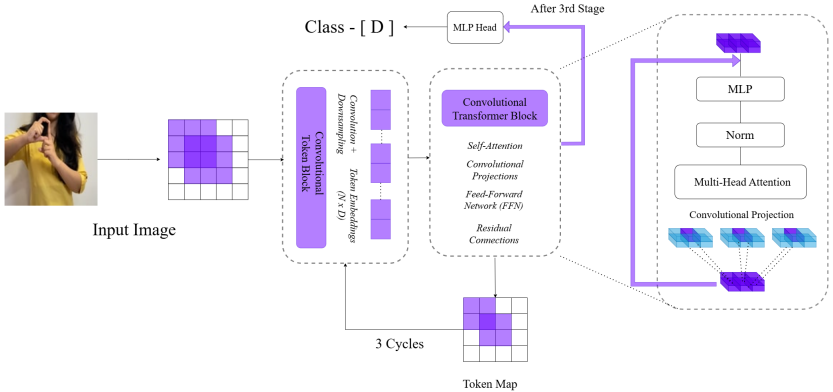


Figure 6.2:Convolutional Vision Transformer Architecture

CVT Architecture Overview

6 Methodology

- **Convolutional Token Embedding:** Overlapping Conv layers extract local spatial features and reduce resolution.
- **Hierarchical Transformer Blocks:** CVT blocks combine depthwise separable convolutions with multihead self-attention for efficiency and locality.
- **Multi-Stage Processing:** Three stages progressively reduce spatial size (from 56×56 to 7×7) while increasing feature dimension.
- **Global Context Capture:** Self-attention layers model long-range dependencies across the entire feature map.
- **Classification Head:** Global average pooling aggregates tokens, followed by an Multi Layer Perceptron (MLP) to predict final class probabilities.

CVT Model Summary

6 Methodology

- **Input:** 224×224 RGB image
- **Convolutional Token Embedding:**
  - Overlapping conv layers (7×7, stride 2 → 3×3, stride 1)
  - Downsamples spatial map and produces tokens (N × D)
- **Three Hierarchical Stages:**
  - Stage 1: 3 CVT blocks on 56×56 tokens (embed 256, heads 4)
  - Stage 2: 4 CVT blocks on 28×28 tokens (embed 384, heads 6)
  - Stage 3: 6 CVT blocks on 14×14 tokens (embed 768, heads 8)
- **CVT Block Structure:** Conv projections → Multi-head self-attention → Feed-forward network → Residual + LayerNorm
- **Classification Head:** Global average pooling over final tokens → single MLP → Softmax

Landmark-Based Preprocessing

6 Methodology

1. **Sharpening:**
   - Unsharp mask to emphasize hand edges.
2. **Landmark Detection:**
   - MediaPipe HandLandmarker extracts 21 hand keypoints.
   - Skeleton image generated using OpenCV.

Before and After Landmark Transform

Figure 6.3:Before Landmark Transform

Figure 6.4:After Landmark Transform

Training Strategy

6 Methodology

- **Base Model:** Pre-trained `microsoft/cvt-13` from Hugging Face
- **Freezing Strategy:** Early layers frozen initially; progressively unfrozen after 5 epochs
- **Training Split:** 70% training, 20% validation, 10% testing
- **Loss:** Binary Cross-Entropy
- **Optimizer:** AdamW with ReduceLROnPlateau scheduler

Hyperparameters:

- Batch Size: 64
- Epochs: 100
- Initial LR: 1×10^{-6}
- Min LR: 1×10^{-9}

Training Enhancements

6 Methodology

1. **Callbacks**
   - **Model Checkpointing:** Save top 2 models with lowest validation loss.
   - **Early Stopping:** Stops after 10 epochs of no improvement.
2. **Hardware and Runtime**
   - **GPU:** NVIDIA T4 (Google Colab)
   - **Framework:** PyTorch Lightning + Python 3.8

Python and Key Libraries

6 Methodology

Implementation Language: Python 3.8

Key Libraries Used:

- **PyTorch Lightning:** Structured training loop and callback management.
- **OpenCV:** Image I/O, resizing, sharpening, and geometric transforms.
- **MediaPipe:** Real-time hand keypoint extraction using HandLandmarker.
- **scikit-learn:** Evaluation metrics and supporting utilities.

Supported Formats: JPEG (lossy) and PNG (lossless)

Operations with OpenCV:

- Reading/writing images and converting to grayscale
- Color space transformations (RGB, HSV, etc.)
- Unsharp Masking: Enhances contours for landmark clarity
- Geometric transforms: Rotation, scaling, and perspective warp

Evaluation Metrics

1. **Accuracy (Multiclass)**

$$\text{Accuracy} = \frac{\text{Correct Predictions}}{\text{Total Predictions}} \tag{1}$$

2. **Weighted Precision & Recall**

$$\text{Precision}_{\text{weighted}} = \frac{\sum_c N_c \, \text{Precision}_c}{\sum_c N_c}, \tag{2}$$

$$\text{Recall}_{\text{weighted}} = \frac{\sum_c N_c \, \text{Recall}_c}{\sum_c N_c} \tag{3}$$

3. **Weighted F1 Score**

$$\text{F1}_{\text{weighted}} = \frac{\sum_c N_c \, \text{F1}_c}{\sum_c N_c} \tag{4}$$

OUTLINE

7 Results and Evaluation

Model Comparison (Phase 1)

7 Results and Evaluation

The CVT model was trained on the Phase 1 dataset that includes classes for all the digits (1–9) and alphabets (A–Z). Table III shows the performance of the CvT model along with the state-of-the-art (SOTA) deep learning models, such as ResNet, ViT, InceptionV3, and VGG16 with respect to the evaluation metrics mentioned earlier.

Table III: Performance evaluation on SOTA models (Alphabet Level)

Metric	ResNet	ViT	InceptionV3	VGG16	CVT
Precision	81.92%	91.85%	85.62%	83.44%	**92.17%**
Recall	81.33%	90%	84.10%	82.91%	**91.14%**
F1-score	80.50%	89.45%	83.85%	81.68%	**90.92%**
Accuracy	81.67%	90.94%	84.9%	83.13%	**91.66%**

Model Comparison (Phase 2)

7 Results and Evaluation

Additional word-level classes (such as “you” , “who” , “agree” , etc.) were included in Phase 2. Two experiment runs were performed: one with raw images and another using landmark extraction and filtering. Table IV compares performance using the full dataset. The CVT model showed improved accuracy when pre-processing was applied.

Table IV: Performance Metrics With and Without Preprocessing

Metric	Base Dataset	Filtered + Landmark Dataset
Precision	85.87%	90.98%
Recall	86.21%	91.13%
F1-score	85.54%	90.91%
Accuracy	86%	91.05%

Accuracy Curve

7 Results and Evaluation

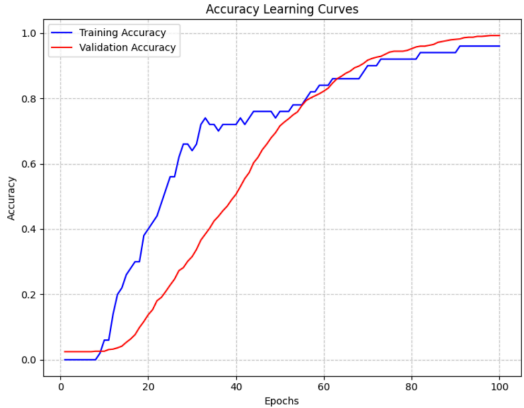


Figure 7.1:Training and Validation Accuracy over Epochs

Loss Curve

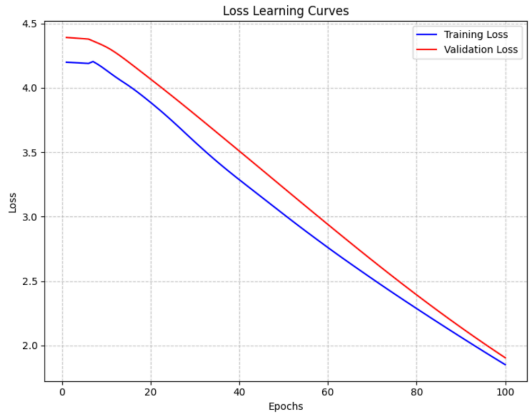


Figure 7.2:Training and Validation Loss over Epochs

Sample Prediction Output

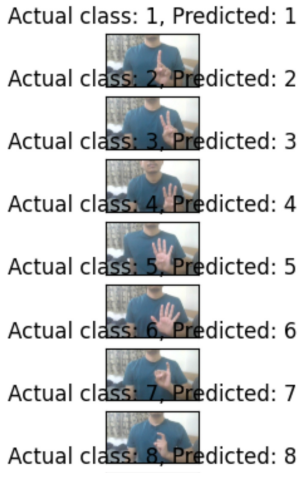


Figure 7.3:Sample Prediction Output on Numbers ISL

Discussion

7 Results and Evaluation

- **Phase 1 (Alphabet Recognition)**
  - CVT achieved the highest accuracy (91.14%) and F1-score (90.92%), outperforming other models.
  - ViT performed well (90% accuracy), while ResNet, InceptionV3, and VGG16 showed lower performance.
  - Transformer-based models (CVT, ViT) proved more effective for individual hand gesture recognition.

- **Phase 2 (Word & Alphabet Classification)**
  - Adding word-level classes increased complexity, reducing the accuracy (86%) on raw images.
  - Pre-processing (image filtering + landmark extraction) significantly improved the accuracy (91.05%).

Summary and Key Insights

7 Results and Evaluation

- **CVT Effectiveness**: CVT outperforms other models by combining convolutional and transformer features for better spatial-contextual learning.
- **Preprocessing Impact**: Filtering and landmark extraction boost accuracy from 86% to 91.05%, improving word recognition.
- **Augmentation Benefits**: Techniques like flipping and color jittering enhance generalization and reduce overfitting.
- **Optimized Training**: Early stopping and checkpointing ensure efficient training and prevent overfitting.

OUTLINE

8 Design and Development Approach

Architecture

8 Design and Development Approach

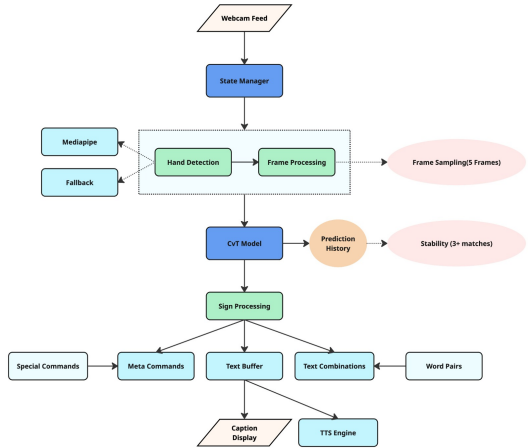


Figure 8.1:Architecture of Application

System Overview: Input and Detection

8 Design and Development Approach

1. Live Feed Acquisition

- Real-time video stream captured using OpenCV.
- GUI displays frame rate, system status, and prediction overlay.

2. Hand Detection & Region Isolation

- Primary detection via MediaPipe HandLandmarker.
- Fallback to Haar Cascade + skin color segmentation if MediaPipe fails.
- Bounding boxes and landmarks are drawn for feedback and ROI extraction.

System Overview: Model Prediction and Smoothing

3. Gesture Prediction

- Convolutional Vision Transformer for classification.
- Combines local spatial (CNN) and global contextual (Transformer) features.
- TensorRT-accelerated inference using FP16 and channels_last format.

4. Gesture Stability & Prediction Smoothing

- Uses a prediction buffer to store recent outputs.
- Accepts a prediction only if it appears at least 3 times in the last 5 frames.
- Helps reduce flickering and noise in output.

System Overview: Meta Commands and Interaction

8 Design and Development Approach

5. Meta Command Execution

- Recognized gestures can trigger system-level actions:
  - `_BACKSPACE` → Deletes last word.
  - `_SPEECH` → Converts text to speech.
- Interactive prompt for compound gestures:
  - e.g., “old” + “mother” → “grandmother”

6. Text Aggregation

- Recognized words/letters are appended to a running text field.
- Maintains coherence of phrases and phrases.

System Overview: Output and Deployment Optimization

8 Design and Development Approach

7. Text-to-Speech Integration

- Offline text-to-speech synthesis using `pyttsx3`.
- Activated by a gesture command or GUI button.

8. Deployment Optimizations

- TensorRT inference using `torch_tensorrt` on GPU.
- FP16 precision + channels_last layout for memory efficiency.
- Supports quantization-aware inference for lower-latency processing.

OUTLINE

9 References

References

Nojood M. Alharthi and Salha M. Alzahrani.
Vision transformers and transfer learning approaches for arabic sign language recognition.
Applied Sciences, 13(21), 2023.

Necati Cihan Camgoz, Oscar Koller, Simon Hadfield, and Richard Bowden.
Sign language transformers: Joint end-to-end sign language recognition and translation.
In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

References

9 References

Deep Kothadiya, Chintan Bhatt, Tanzila Saba, and Amjad Rehman.
Signformer:deepvision transformer for sign language recognition.
IEEE Access, PP:1–1, 01 2023.

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, 12 2018.

References

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.
04 2020.

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.
01 2018.

Shagun Katoch, Varsha Singh, and Uma Shanker Tiwary.
Indian sign language recognition system using surf with svm and cnn.
Array, 14:100141, 04 2022.

References

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.

Mathieu De Coster, Mieke Van Herreweghe, and Joni Dambre.
Sign language recognition with transformer networks.
In Nicoletta Calzolari, Frédéric Béchet, Philippe Blache, Khalid Choukri, Christopher Cieri, Thierry Declerck, Sara Goggi, Hitoshi Isahara, Bente Maegaard, Joseph Mariani, Hélène Mazo, Asuncion Moreno, Jan Odijk, and Stelios Piperidis, editors, *Proceedings of the Twelfth Language Resources and*

References

Evaluation Conference, pages 6018–6024, Marseille, France, May 2020. European Language Resources Association.

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


राष्ट्रीय प्रौद्योगिकी संस्थान, सिलचर
NATIONAL INSTITUTE OF TECHNOLOGY SILCHAR

Thank You!