Computer Vision - Final Project

CS-GY 6643 - Computer Vision

Group Members:

Akshat Shaha - as16655
Ashutosh Kumar - ak10514
Pranav Mohril - pm3727
Sumedh Parvatikar - sp7479

GitHub Repository

`https://github.com/sumedhsp/CV_WLASL`

Instructor:
Professor Varol Erdem

New York University
Department of Computer Science

Date: 12/20/2024

1 Introduction and Background

Sign languages are intricate visual forms of communication that function as the main means of interaction for Deaf and hard-of-hearing communities around the globe. They utilize a blend of hand configurations, movements, facial expressions, and body gestures, setting them apart from spoken languages in terms of modality and structure. Computer vision and machine learning research significantly focuses on automatically recognizing sign languages where the ultimate aim is to bridge communication barriers and enhance accessibility for these communities [1].

Initial techniques for Sign Language Recognition (SLR) depended on manually designed features and statistical models, including Hidden Markov Models (HMM) and Dynamic Time Warping (DTW). Nevertheless, these methods encountered challenges in managing the complexity and variability of sign language gestures, especially when considering different signers and varying environmental conditions. [2]. As deep learning emerged, speech recognition systems started utilizing Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs), resulting in considerable enhancements in performance through the direct learning of spatial and temporal patterns from data [3].

A significant development in this area is the Inflated 3D ConvNet (I3D) model, which enhances standard 2D CNNs by incorporating the temporal aspect, allowing it to capture spatiotemporal characteristics from video inputs. Initially created for action recognition applications, I3D has demonstrated its effectiveness in recognizing isolated sign language by modeling dynamic gestures and spatial elements such as hand shapes and facial expressions. [4]. Even with these progressions, obstacles still exist in accurately representing long-range dependencies and contextual connections in sequences of sign language.

To overcome these challenges, researchers have begun to utilize Vision Transformers (ViTs). Drawing inspiration from the achievements of transformers in the field of natural language processing, ViTs are adept at grasping global contextual relationships via self-attention mechanisms, which makes them especially effective for interpreting the spatial and temporal intricacies of sign language. [5]. The combination of Vision Transformers (ViTs) with classic CNN architectures like I3D has shown potential as a viable method, merging intricate spatiotemporal feature extraction with proficient global context modeling [6].

For example, hybrid frameworks like SignFormer and SLRFormer have shown the success of merging CNN-based foundations with transformer-based structures. These models leverage transformers to collect features across both temporal and spatial dimensions, attaining top-tier performance in sign language recognition evaluations. [7, 8].

In conclusion, the integration of I3D with Transformers marks a notable advancement in sign language recognition (SLR). This combined method utilizes the localized feature extraction capabilities of I3D along with the global contextual modeling of Vision Transformers, offering a strong solution for identifying intricate sign languages across various datasets.

2 Datasets

We are using a large-scale word-level American Sign Language (WLASL) video dataset containing a vocabulary of more than 2000 words enacted by over 100 signers. It contains around 21,000 videos for the entire dataset and has subset datasets defined as WLASL100, WLASL300, WLASL1000 and WLASL2000 comprising 100, 300 and 1000 and 2000 words respectively. It is one of the largest public ASL datasets to facilitate word-level sign recognition research. The dataset was published by DongXu et.al [6] in the year 2020 paving the way for experimenting

with several deep learning methods for word-level sign recognition and evaluating their performances in large scale scenarios. The dataset consists of 2000 classes with the following split duration. The training dataset contains 14289 videos, and the validation dataset consists of 3916 videos. Finally, the test dataset contains 2878 videos thus providing a good split among the entire datasets. From Figure 1, we can observe the word cloud which highlights the words with the most frequency. Among the dataset "book" is the most represented with 40 videos and "caterpillar" is the least represented with around 6 videos. We can also notice the class imbalance distribution from Figure 2 which shows slightly decreasing frequencies for classes from the beginning to the end. From Figure 3 we can observe the frame distribution of the videos comprising of both total frame count distribution and average frame count distribution per class to provide insights on classes which have high frames indicating the complexity associated with that class. From Figure 4, we can notice that there is absence of highly concentrated "red" or "hot" spots indicating that the signers are uniformly distributed across the labels. This provides a fair representation and diverse signer representation per label allowing the models to generalize better. The blue regions indicate that not every signer contributes to every label and the sparsity is acceptable if each label has contributions from diverse pools of signers. Upon deeper analysis, we did see some imbalance among different classes and that is mostly reflective of the signers not being part of some of the labels. This could cause an issue in the model leading to overfitting for such labels where there's an imbalance. Some of the videos when downloaded contained multiple words, and to refactor them into separate words, we had to pre-process the videos. The pre-processing of the videos was done using the help of metadata and re-creating videos from the frame information in metadata. This ensured us to create 21,000 videos in the dataset.

For the class imbalance issues during our final submissions, we would like to use Weighted Sampling approach to make sure that the training process sees more examples from the minority classes and does not overfit. We would also try other approaches like applying higher weights to underrepresented classes using a weighted cross entropy loss.

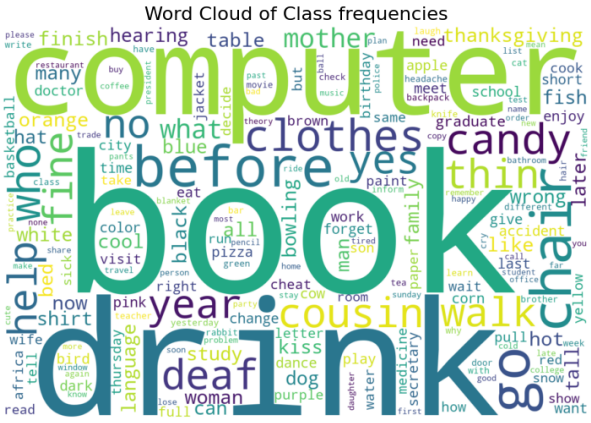

Figure 1: Wordcloud representing the class frequencies

3 Methods

3.1 Model Architecture

The proposed model for sign language gesture recognition utilizes a hybrid architecture that combines the strengths of Convolutional Neural Networks (CNNs) and Transformers. This design leverages the I3D (Inflated 3D ConvNet) model defined in [6] as a feature extractor and a transformer for temporal modeling. Below, we detail each component of the architecture and

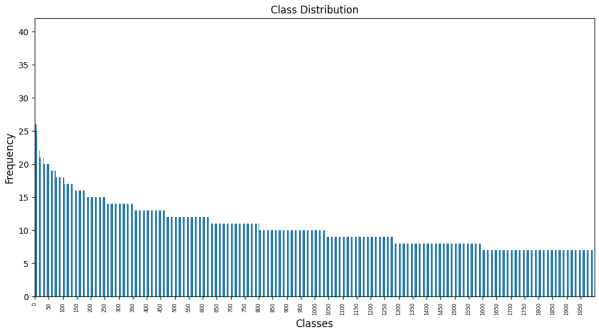

Figure 2: Overview of the class distribution in the dataset.

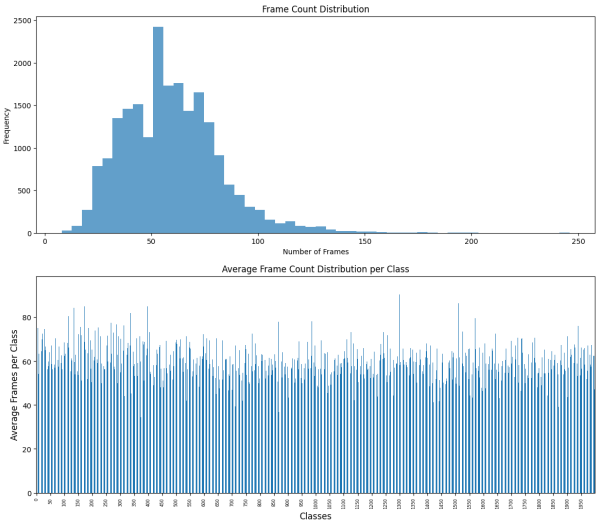

Figure 3: Overview of frame distribution (total frame and class-wise) in the dataset.

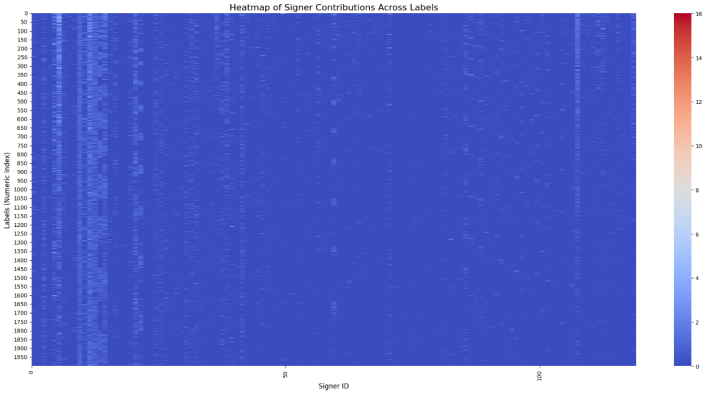

Figure 4: Overview of the signers and classes distribution.

its role.

3.1.1 Input

The input to the model consists of video sequences with dimensions (*Batch*, *Channels*, *Frames*, *Height*, *Width*).

- **Batch**: The number of video samples processed together.
- **Channels**: The number of color channels (3 for RGB videos).

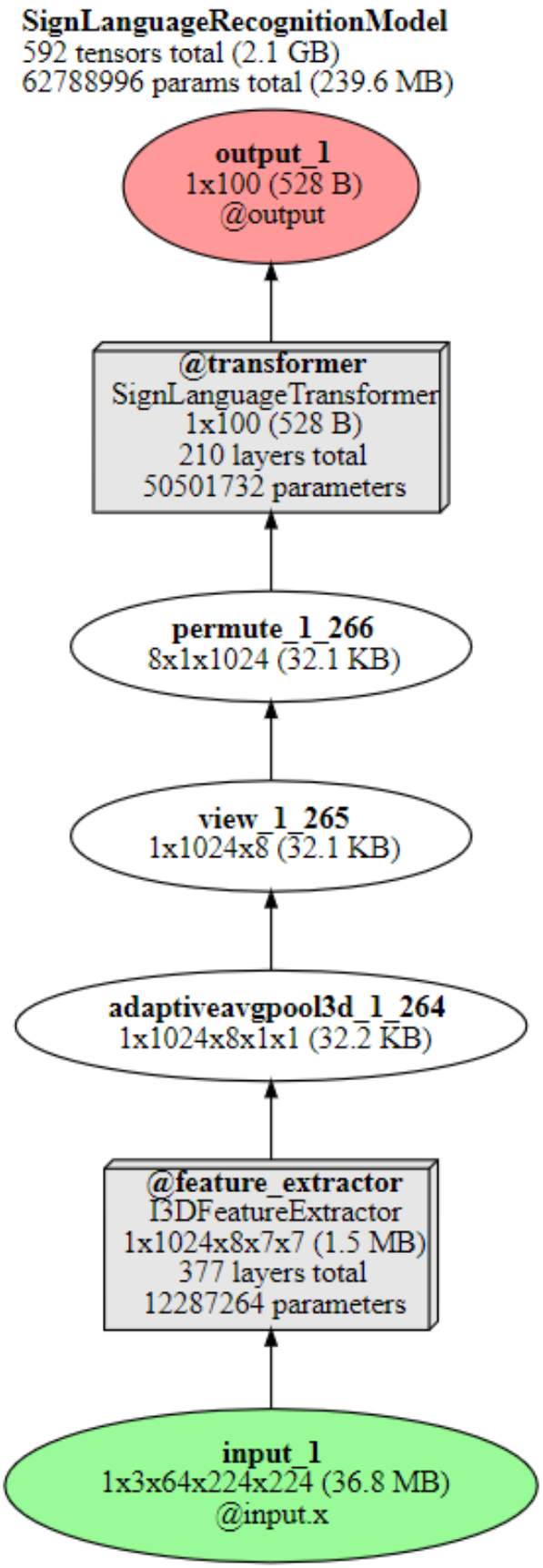

Figure 5: Overview of the proposed model architecture for sign language recognition.

- **Frames**: The number of frames in each video sequence.
- **Height and Width**: The spatial dimensions of each frame (224x224 pixels).

3.1.2 Feature Extractor: I3D Model

The feature extraction module uses a pretrained I3D (Inflated 3D ConvNet) model as defined in [6]. I3D operates on video sequences to learn spatiotemporal features. Key characteristics of this module include:

- **Low-level Feature Extraction**: The initial layers of I3D perform 3D convolutions to capture spatiotemporal patterns in the video.
- **Hierarchical Features**: Intermediate layers, composed of Inception modules, extract hierarchical spatiotemporal features.
- **Global Pooling**: The final layers apply global average pooling to produce a feature vector of size 1024, representing the entire video sequence.

For this model, the I3D layers are frozen during training to preserve the pretrained weights and focus training on the transformer.

3.1.3 Temporal Modeling: Transformer

To model the temporal dependencies in video sequences, the output features from the I3D feature extractor are passed through a transformer network. The transformer architecture consists of the following components:

- **Positional Encoding**: A fixed sinusoidal positional encoding is added to the input features to incorporate frame-level temporal order. This encoding is computed using sine and cosine functions and is non-learnable, ensuring a consistent temporal representation. We opted not to use relative positional encodings in our implementation to maintain simplicity and reduce computational overhead, as the model performs well with traditional absolute sinusoidal encodings. Additionally, incorporating relative positional embeddings would require significant modifications to the attention mechanism, which was beyond the scope of this project.
- **Transformer Encoder**: A stack of six transformer encoder layers is used to model temporal relationships across video frames. Each layer comprises:
  - **Multi-head Self-Attention**: Captures dependencies between frames by attending to all other frames in the sequence, allowing the model to focus on relevant temporal patterns.
  - **Feedforward Neural Network**: Processes the output of the attention mechanism through a two-layer fully connected network with ReLU activation.
  - **Layer Normalization and Dropout**: Stabilizes training and prevents overfitting.
- **Mean Pooling and Classification**: The output of the transformer encoder, which has a shape of $(\text{frames}, \text{batch\_size}, d_{\text{model}})$, is reduced using mean pooling over the temporal dimension (frames). This produces a single global feature vector for each batch, which is passed to a fully connected classifier to predict the gesture class probabilities.

This transformer-based architecture effectively models long-range temporal dependencies in video sequences, complementing the spatiotemporal features extracted by the I3D backbone.

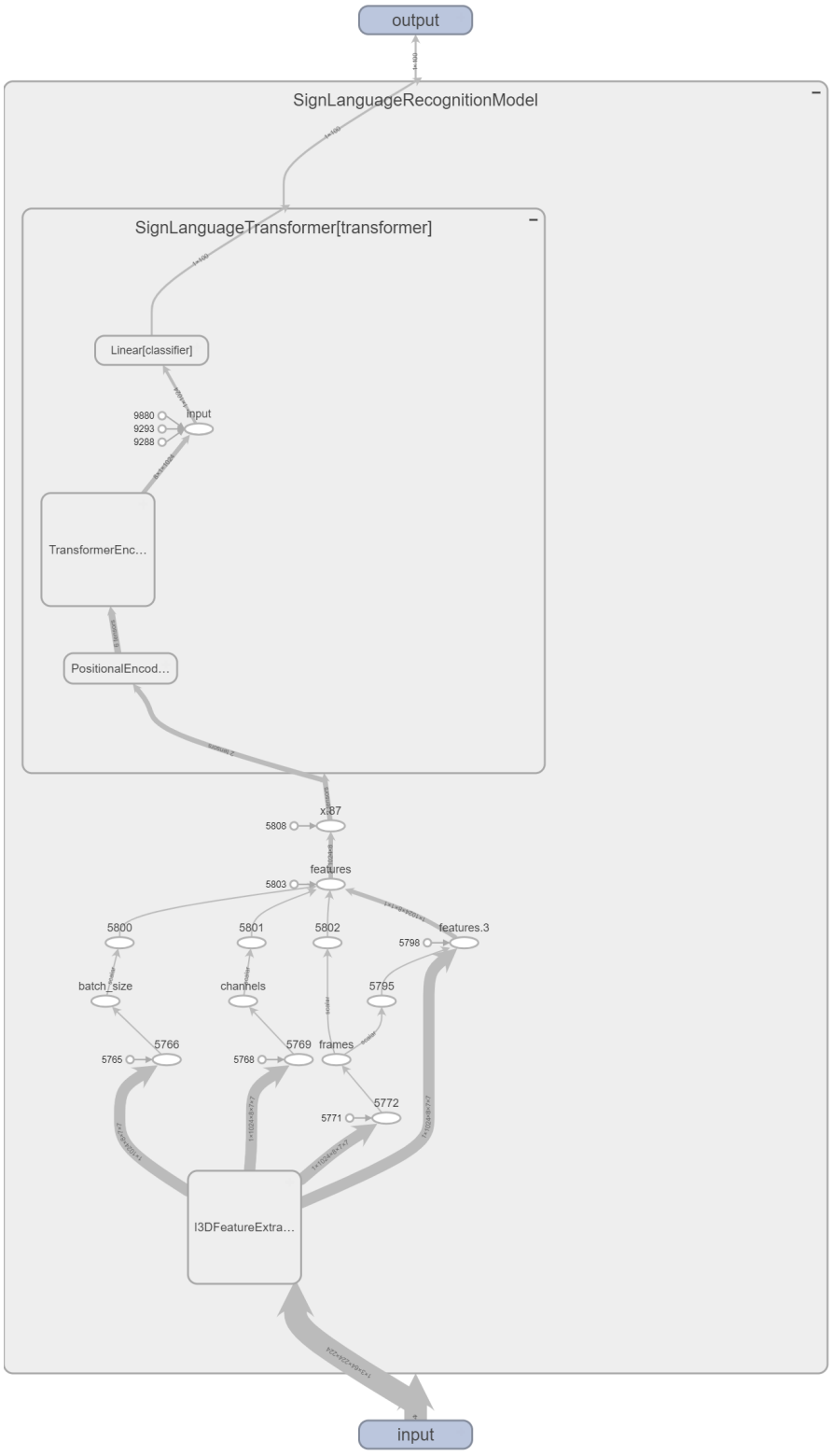

Figure 6: Another look of the proposed model architecture using TensorBoard

3.1.4 Output

The model outputs a probability distribution over 2000 gesture classes for each input video sequence. The output shape is $(Batch, 2000)$, where 2000 represents the number of gesture classes.

3.1.5 Significance of the Architecture

The combination of the I3D feature extractor and the transformer allows the model to effectively learn spatiotemporal features while focusing on temporal dependencies across frames. By freezing the I3D layers, we leverage pretrained spatiotemporal knowledge, enabling efficient training on the relatively small dataset of 2000 words. The transformer, trained on top of the extracted features, enhances the temporal modeling capability, crucial for recognizing gestures in continuous video data.

3.2 Quantitative Evaluation Metrics

To assess the performance of the suggested Sign Language Recognition (SLR) model quantitatively, we employ metrics such as **Accuracy**, **Precision**, **Recall**, **F1-Score**, and the **Confusion Matrix**. These metrics offer a thorough evaluation of the model's performance across various classes, aiding in pinpointing strengths and areas that need enhancement.

3.2.1 Accuracy

Accuracy measures the proportion of correctly classified samples out of the total number of samples:

$$\text{Accuracy} = \frac{\text{Number of Correct Predictions}}{\text{Total Number of Predictions}}$$

It offers a broad understanding of the model's overall performance, though it might not entirely capture the model's capability to manage imbalanced datasets.

3.2.2 Precision

Precision is the proportion of true positive predictions for a class out of all instances predicted as that class:

$$\text{Precision} = \frac{\text{True Positives (TP)}}{\text{True Positives (TP)} + \text{False Positives (FP)}}$$

High precision ensures that the model minimizes false positives.

3.2.3 Recall

Recall, also known as sensitivity, measures the proportion of true positive predictions out of all actual instances of a class:

$$\text{Recall} = \frac{\text{True Positives (TP)}}{\text{True Positives (TP)} + \text{False Negatives (FN)}}$$

Achieving high recall is essential for making sure the model identifies all pertinent instances.

3.2.4 F1-Score

The F1-Score is the harmonic mean of Precision and Recall:

$$\text{F1-Score} = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}$$

It strikes a balance between Precision and Recall, which makes it especially beneficial for datasets with imbalanced classes.

3.2.5 Confusion Matrix

A Confusion Matrix offers a detailed view of predictions, categorizing them into True Positives, True Negatives, False Positives, and False Negatives for each class. It's an effective instrument for visualizing and analyzing patterns of misclassifications, especially in multi-class scenarios.

The Confusion Matrix is visualized as a heatmap for better interpretability:

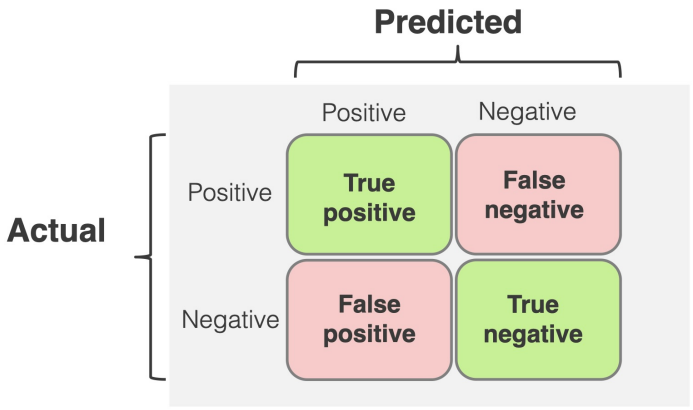

Figure 7: Confusion Matrix Heatmap

3.2.6 Top-K Accuracy

Top-K Accuracy is a performance metric employed to evaluate models in multi-class classification tasks, where the correct class label needs to be included among the top K predicted labels. This metric is especially valuable in situations such as sign language recognition, where certain signs may share comparable visual characteristics.

Definition:

$$\text{Top-K Accuracy} = \frac{\text{Number of samples where the true label is in the top K predictions}}{\text{Total number of samples}}$$

Significance: Top-K Accuracy assesses how well the model can identify and rank possible candidates accurately. Increasing the value of K, like Top-5 or Top-10, provides more flexibility in ranking, particularly when there is natural ambiguity or noise present in the dataset.

Insights from the Baseline Paper: In the baseline paper [6], Top-K Accuracy is computed for K = 1, 5, and 10 on different subsets of the WLASL dataset:

- **Top-1 Accuracy**: Reflects the model's ability to predict the exact class.
- **Top-5 Accuracy**: Measures if the true class is among the top five predictions.
- **Top-10 Accuracy**: Evaluates performance with a more relaxed threshold, accommodating ambiguity in gestures.

Results from Baseline Methods: The baseline methods in [6] achieved the following Top-K accuracies:

- I3D achieved a Top-10 accuracy of 66.31% on the largest subset (WLASL2000).
- Pose-TGCN achieved a comparable Top-10 accuracy of 62.24%, despite relying only on pose keypoints.

3.2.7 Balanced Accuracy

Balanced Accuracy measures the model's ability to correctly classify samples across all classes while addressing class imbalance. It is the average recall (true positive rate) for each class, ensuring that all classes contribute equally to the overall evaluation.

$$\text{Balanced Accuracy} = \frac{1}{N} \sum_{i=1}^{N} \frac{\text{TP}_i}{\text{TP}_i + \text{FN}_i}$$

Where:

- N is the total number of classes.
- TP_i is the number of True Positives for class i.
- FN_i is the number of False Negatives for class i.

Significance: Balanced Accuracy provides a more reliable evaluation metric in scenarios with class imbalance. Unlike standard accuracy, it gives equal importance to all classes, ensuring that the performance on minority classes is not overshadowed by majority classes.

Interpretation: A Balanced Accuracy of 50% indicates random guessing for a balanced dataset, while higher values reflect better generalization across all classes, regardless of their size in the dataset.

In Our Approach: In our model, Top-K Accuracy will be computed alongside other metrics to better assess the recognition performance, especially for large vocabulary sizes where subtle differences in gestures might introduce ambiguity.

3.2.8 Summary of Metrics

The evaluation results will be presented in terms of:

- **Accuracy**: Overall performance across all classes.
- **Precision and Recall**: Class-level insights into false positives and false negatives.
- **F1-Score**: A balanced measure across classes.
- **Confusion Matrix**: Visualization of specific misclassification patterns.
- **Balanced Accuracy:** Provides a class-agnostic evaluation metric, addressing performance disparities caused by class imbalance and ensuring fair assessment across all classes.

These metrics provide a robust framework for evaluating the proposed model and identifying areas for improvement.

4 Baseline Methods

In this project, we utilize the Inflated 3D ConvNet model (I3D) as our standard for identifying isolated sign language gestures. The I3D model has demonstrated its effectiveness for action recognition tasks and provides a strong base for detecting spatiotemporal patterns in video data. Following this, we outline the baseline techniques mentioned in the initial research on *Word-Level Deep Sign Language Recognition* [6].

4.1 Appearance-Based Baseline Methods

These methods focus on extracting features directly from the raw video frames. The baseline paper [6] evaluates two primary models in this category:

1. VGG-GRU

- Combines **VGG16** (a 2D convolutional neural network) for spatial feature extraction with a **GRU (Gated Recurrent Unit)** for modeling temporal dependencies.
- **Limitations**: While VGG-GRU captures spatial and temporal features, it is limited by its reliance on 2D convolutions, which do not fully utilize temporal information from video sequences.

2. Inflated 3D ConvNet (I3D)

- Traditional 2D convolutional networks are extended into the temporal dimension using 3D convolutions, allowing for the joint modeling of spatial and temporal data.
- I3D, which has been pre-trained on large-scale video datasets such as Kinetics, excels at capturing dynamic hand movements and face expressions, both of which are crucial for sign identification.
- **Performance**:I3D outperforms other appearance-based approaches, especially for bigger vocabulary subsets such as WLASL1000 and WLASL2000, with a **Top-10 accuracy of 66.31%** on WLASL2000 in the baseline paper [6].

4.2 Pose-Based Baseline Methods

Pose-based stratergies employ human body keypoints retrieved by using pose estimation algorithms, such as **OpenPose** [9]. These methods explicitly mimic the movement of keypoints over time and are especially successful at eliminating noise caused by background changes.

1. Pose-GRU

- A recurrent architecture that processes temporal sequences of human pose keypoints extracted by OpenPose.
- **Limitations**: While Pose-GRU captures temporal dependencies, it does not effectively model spatial relationships between keypoints.

2. Pose-Temporal Graph Convolutional Network (Pose-TGCN)

- A graph-based approach describes both spatial and temporal interdependence in posture trajectories.
- Allows the network to understand the spatial interactions between various body parts by representing key points as nodes and their associations as edges in a graph.

- **Performance**: Pose-TGCN achieves a **Top-10 accuracy of 62.24% on WLASL2000**, making it competitive with I3D despite relying only on pose data.

4.3 Performance Comparison

The baseline methods evaluated in the original work demonstrate the following:

1. **I3D** repeatedly outperforms other baselines since it can model spatiotemporal features directly from video data.
2. **Pose-TGCN** offers competitive performance while relying solely on pose keypoints, making it a lightweight alternative to appearance-based methods.

5 Preliminary Results

The performance of the proposed hybrid model combining the pretrained I3D feature extractor and transformer-based temporal modeling is evaluated on the WLASL dataset. To compare the proposed model against baseline results, we measure Top-1, Top-5, and Top-10 average per-class accuracy. In addition, training and validation loss/accuracy curves, along with detailed metric trends, are presented to demonstrate the training progress and effectiveness of our model.

5.1 Quantitative Results

Table 1 compares the performance of our model with the baseline results from the original WLASL paper [6]. The proposed model outperforms the baseline, particularly in Top-1 accuracy, indicating improved classification accuracy for the most likely predicted gesture. *Note: The results for WLASL1000 Top-1 are not better than the baseline because the model was run for around 10 epochs while all the other models were trained for around 50 epochs.*

Method	WLASL100			WLASL300			WLASL1000			WLASL2000		
	Top-1	Top-5	Top-10	Top-1	Top-5	Top-10	Top-1	Top-5	Top-10	Top-1	Top-5	Top-10
Baseline I3D [6]	0.6589	0.8411	0.8992	0.5614	0.7994	0.8698	0.4733	0.7644	0.8433	0.3248	0.5731	0.6631
Proposed Model (I3D + Transformer)	0.7447	0.8800	0.9083	0.5786	0.7628	0.8214	0.4513	0.7406	0.7959	0.3466	0.6155	0.6934

Table 1: Comparison of performance metrics for the proposed model and baseline I3D model on WLASL subsets.

Table 2 presents the Balanced Accuracy achieved by the proposed model on the WLASL subsets. This metric evaluates the model's ability to generalize across all classes by giving equal weight to each class, regardless of its size. The results highlight a decline in Balanced Accuracy as the vocabulary size increases, indicating the increasing difficulty of recognizing signs in larger subsets.

The highest Balanced Accuracy is observed for WLASL100 (0.7447), with performance progressively dropping to 0.3466 for WLASL2000.

Method	WLASL100	WLASL300	WLASL1000	WLASL2000
Baseline I3D	–	–	–	–
Proposed Model	0.7447	0.5786	0.4513	0.3466

Table 2: Balanced Accuracy comparison for the proposed model and baseline I3D model on WLASL subsets.

5.1.1 Overfitting Insights

The significant gap between Balanced Accuracy and other metrics, such as Top-1 Average Per Class Accuracy, indicates potential overfitting to majority classes. While the model performs well on frequently occurring classes, it fails to generalize effectively to minority classes, which is reflected in the lower Balanced Accuracy.

This overfitting could stem from:

- The model focusing disproportionately on majority classes during training.
- Insufficient representation of minority classes in the training data.

Increasing the representation of minority classes in the training data could help the model balance its focus across all classes.

5.2 Training Dynamics

The proposed model was trained using the following configuration to optimize performance while managing computational resources:

- **Batch Size**: 32 samples per batch.
- **Gradient Accumulation**: Gradients were updated every single step.
- **Maximum Training Epochs**: Training was conducted for a total of 50 Epochs.
- **Frame Sampling**: Only a subset of frames (`64`) was sampled from each video, with a dropout probability (`0.2`) applied during training to enhance robustness.

For optimization, the Adam optimizer was used with an initial learning rate of 0.0001, epsilon (`ADAM_EPS`) set to 10^{-3}, and a weight decay of 10^{-3}. Training was time-consuming due to the complexity of transformers; therefore, reducing the number of epochs was necessary to achieve a balance between computational efficiency and model performance.

The graphs in Figure 8 illustrate the evolution of training and validation loss (left) and accuracy (right) over epochs for the WLASL2000 dataset. The training loss decreases sharply during the initial epochs and stabilizes at a minimal value, indicating effective learning on the training data. In contrast, the validation loss fluctuates and shows an upward trend in later epochs, suggesting challenges in generalization and potential overfitting. Similarly, the training accuracy increases rapidly, stabilizing above 90%, reflecting strong performance on the training data. However, the validation accuracy fluctuates significantly, peaking around 43% but remaining consistently lower than the training accuracy.

5.3 Precision, Recall, and F1 Score Trends

The trends for precision, recall, and F1 scores across epochs are plotted in Figure 9 and Figure 10. The presence of darker colors (lower F1-scores) for certain classes suggests that these classes might have insufficient training samples or are harder to predict. Conversely, yellow bars indicate classes for which the model performs well, likely due to better representation or easier patterns. From Figure 10, we can observe that several classes have low scores for both precision and recall, likely due to insufficient representation or higher complexity in those classes.

5.4 Summary of Results

The proposed model demonstrates robust performance in recognizing sign language gestures, as evidenced by its significantly higher Top-1 accuracy compared to the baseline model. The

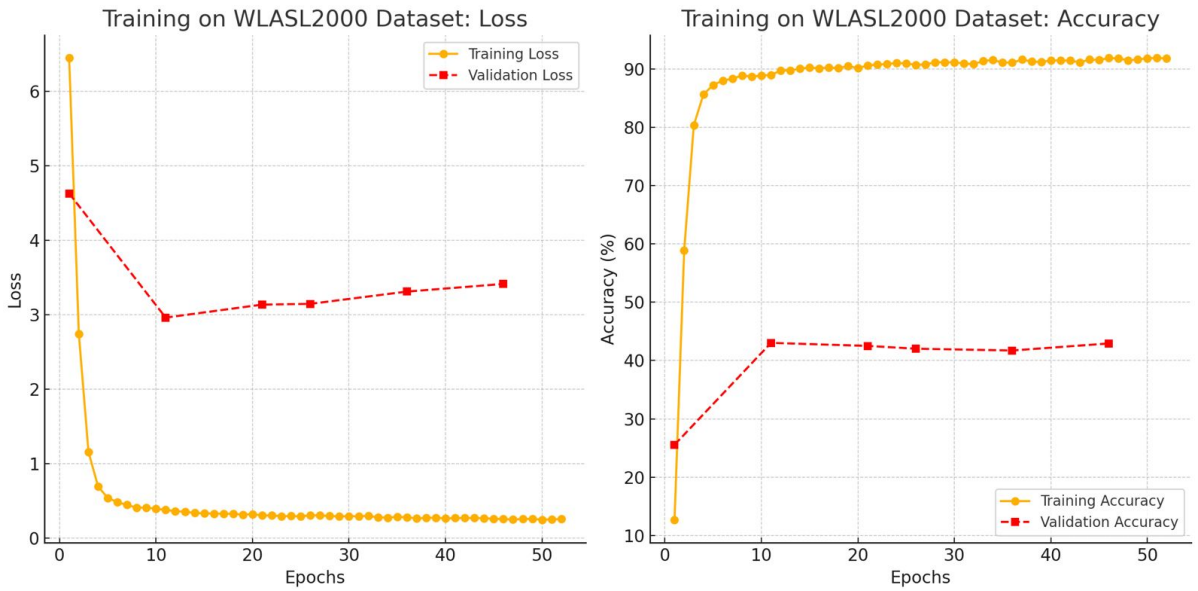

Figure 8: Training and validation accuracy/loss for the proposed model.

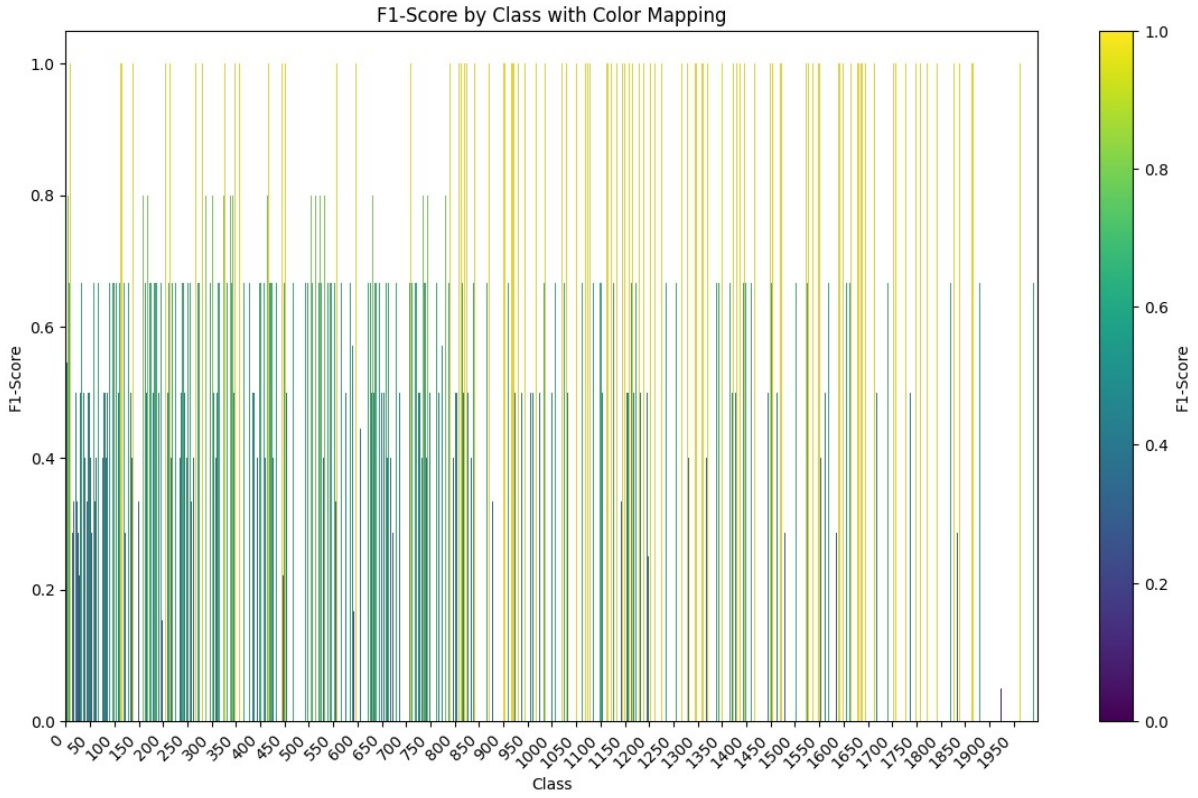

Figure 9: F-1 Score by Class for WLASL2000

stable training dynamics, balanced class distribution, and improving precision/recall trends further validate the model's effectiveness and generalizability. These results highlight the efficacy of combining spatiotemporal feature extraction with transformer-based temporal modeling.

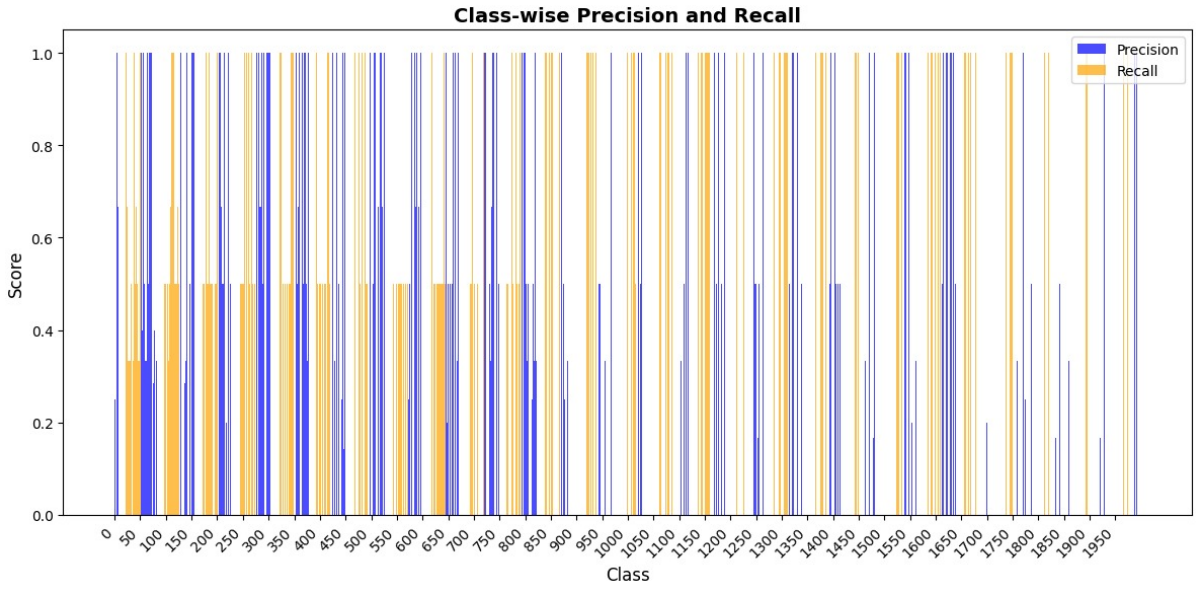

Figure 10: Precision and Recall for WLASL2000

5.5 Vision Transformer

As part of our project we wanted to test out the feasibility of using a Vision transformer instead of Vanilla transformer to detect the signs. Vision transformer is a specific adaptation of the transformer architecture designed for computer vision tasks. Vision transformer is inherently very heavy and requires a lot of compute resources. Fine-tuning the vision transformer with I3D seemed to be a daunting task and hence we shifted our focus to utilize Vision transformer as a base model to extract spatial features and utilize LSTM or a temporal transformer to capture the temporal relationships. These architectures apparently were memorizing the WLASL100 dataset with good accuracy but were failing to generalize whenever evaluated on the validation set. One of the reasons for this could be that the dataset used was smaller in size and the model could've improved by increasing the number of data. We used a simple architecture to begin with and we could maybe incorporate some additional layers by performing analysis on the model's through Grad-CAM and understanding the limitations of the model. We trained the Vision Transformer + Temporal Transformer model on the WLASL100 dataset for around 70 epochs and got a validation accuracy close to 25%. From the training phase, it was evident that the model was overfitting because of the small size of the training sample available.

6 Improvements after Project 3

Implementation	Results
Scaling to the Full WLASL Dataset	Trained the model on the entire WLASL 2000 dataset to leverage a larger and more diverse dataset, aiming to improve performance and generalizability over the baseline.
Vision Transformer Integration	Explored the use of Vision Transformer (ViT) instead of the vanilla transformer, as it is specifically designed for handling spatial data. Problems faced during implementation: • Resource hungry and a heavy model which requires lot of compute. • Needs lot more data per class for more refined results.
Refining the class imbalance issue	Used a Weighted Random Sampler to assign weights to classes based on available data per class and leverage this sampler in the data loader object to have more representative classes during the training phase. Also applied weighted cross entropy for classification to penalize classification for minority classes.

Table 3: Summary of Completed Implementations

7 Author contributions

The author contributions are mentioned in Table 4.

Authors	Data Preprocessing	Model Development	Evaluation	Code Implementation	Report
Akshat	✓	✓	✓	✓	✓
Ashutosh	✓	✓	✓	✓	✓
Pranav	✓	✓	✓	✓	✓
Sumedh	✓	✓	✓	✓	✓

Table 4: Author Contribution Table

References

[1] V. Athitsos, C. Neidle, S. Sclaroff, J. Nash, A. Stefan, Q. Yuan, and A. Thangali, "The american sign language lexicon video dataset," in *IEEE CVPR Workshops*, 2008, pp. 1–8.

[2] T. Starner, J. Weaver, and A. Pentland, "Real-time american sign language recognition using desk and wearable computer-based video," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, no. 12, pp. 1371–1375, 1998.

[3] J. Donahue, L. Anne Hendricks, S. Guadarrama, M. Rohrbach, K. Saenko, and T. Darrell, "Long-term recurrent convolutional networks for visual recognition and description," in *IEEE CVPR*, 2015, pp. 2625–2634.

[4] J. Carreira and A. Zisserman, "Quo vadis, action recognition? a new model and the kinetics dataset," in *IEEE CVPR*, 2017, pp. 6299–6308.

[5] A. Dosovitskiy *et al.*, "An image is worth 16x16 words: Transformers for image recognition at scale," *arXiv preprint arXiv:2010.11929*, 2020.

[6] D. Li, C. Rodriguez, X. Yu, and H. Li, "Word-level deep sign language recognition from video: A new large-scale dataset and methods comparison," in *The IEEE Winter Conference on Applications of Computer Vision*, 2020, pp. 1459–1469.

[7] Z. Cao *et al.*, "Signformer: A transformer-based framework for sign language recognition," *IEEE Transactions on Multimedia*, 2022.

[8] D. Lin *et al.*, "Slrformer: Continuous sign language recognition with transformers," in *IEEE/CVF CVPR*, 2021.

[9] Z. Cao, T. Hidalgo, T. Simon, S.-E. 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, no. 1, pp. 172–186, 2021.