Computer Vision - Project 3

CS-GY 6643 - Computer Vision

Group Members:

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

Instructor:
Professor Varol Erdem

New York University
Department of Computer Science

Date: 12/06/2024

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, WLASL500 and more comprising 100, 300 and 500 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

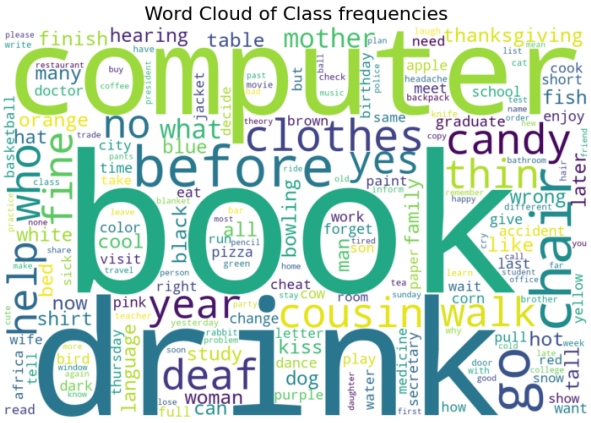

Figure 1: Wordcloud representing the class frequencies

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

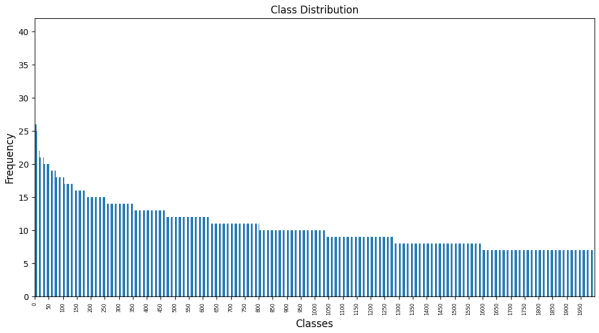

Figure 2: Overview of the class distribution in the dataset.

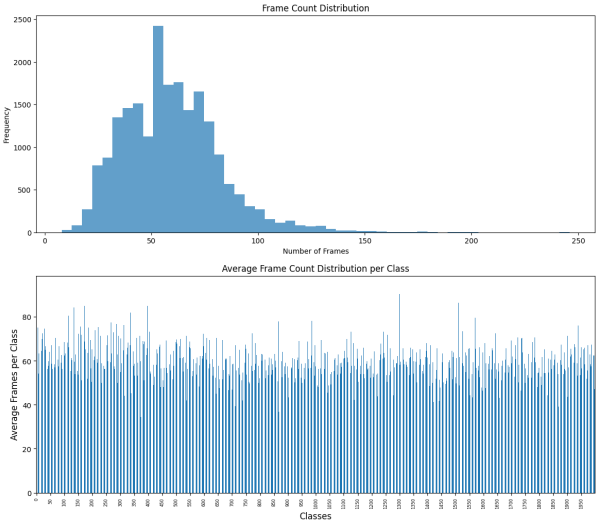

Figure 3: Overview of frame distribution (total frame and class-wise) in the dataset.

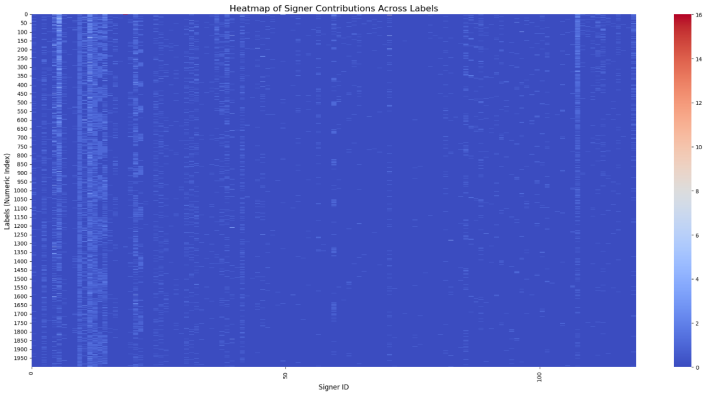

Figure 4: Overview of the signers and classes distribution.

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.

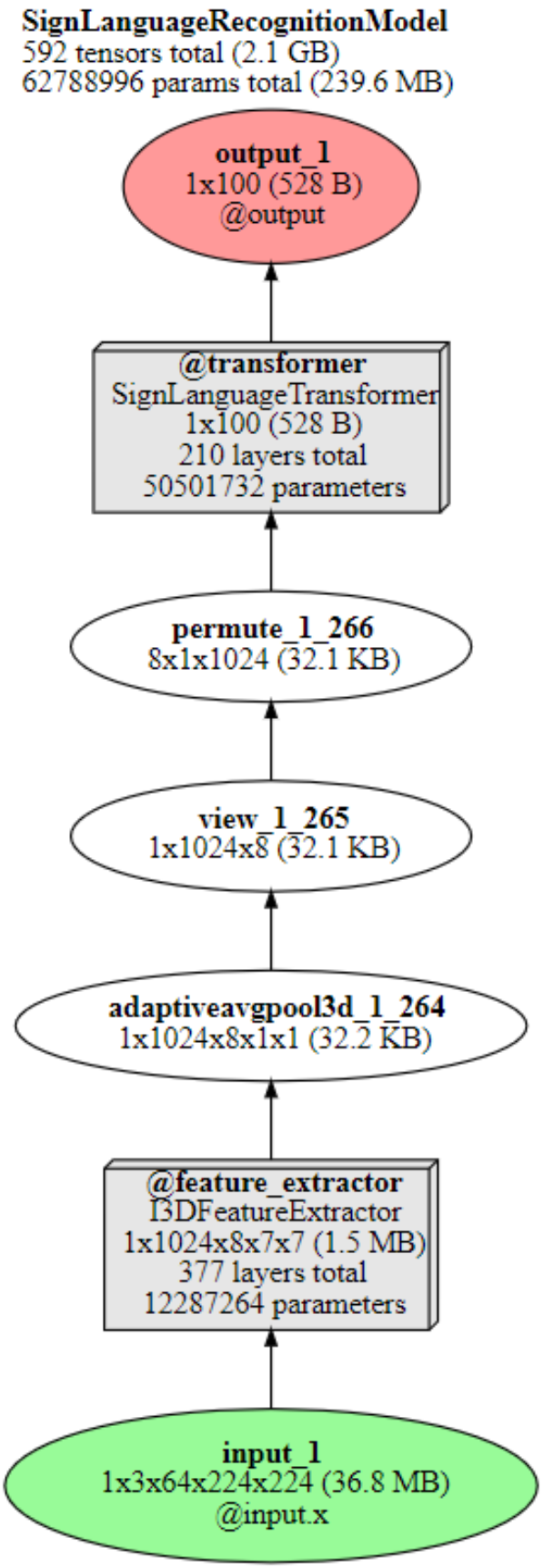

Figure 5: Overview of the proposed model architecture for sign language recognition.

- **Channels**: The number of color channels (3 for RGB videos).
- **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.
- **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.

3.1.4 Output

The model outputs a probability distribution over 100 gesture classes for each input video sequence. The output shape is $(Batch, 100)$, where 100 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 100 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:

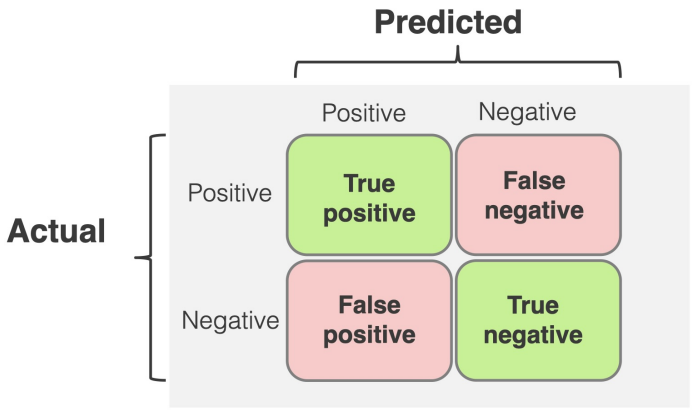

Figure 6: Confusion Matrix Heatmap

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.

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.

Visualization: The Top-K performance across subsets will be presented in tabular and graphical formats, allowing clear comparison with baseline methods.

3.2.7 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.

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. The **Top-1 Accuracy** of I3D on WLASL100 (a subset of the WLASL dataset with 100 classes) is reported as **65.89%** 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** 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, focusing on 100-word gestures. 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 significantly outperforms the baseline, particularly in Top-1 accuracy, indicating improved classification accuracy for the most likely predicted gesture.

Model	Top-1 Accuracy	Top-5 Accuracy	Top-10 Accuracy
Baseline (WLASL [6])	0.6589	0.8411	0.8992
Proposed Model (I3D + Transformer)	**0.7922**	**0.8800**	**0.9083**

Table 1: Comparison of accuracy metrics between the baseline and the proposed model.

5.2 Training Dynamics

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

- **Batch Size**: 12 samples per batch.
- **Gradient Accumulation**: Gradients were updated every 2 steps to simulate a larger batch size without requiring additional memory.
- **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 0.001, epsilon (`ADAM_EPS`) set to 10^{-8}, and a weight decay of 10^{-5} to prevent overfitting. 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 training and validation curves for the proposed model are shown in Figure 7. The top graph highlights the evolution of training and validation losses across epochs, while the bottom graph demonstrates the improvement in accuracy over time. Training loss decreases consistently over time, indicating that the model is effectively learning from the data. The validation loss has fluctuations across the checkpoints (epochs 1, 11, 21, etc.), which suggests that the model may be struggling to generalize at certain points and could be slightly over-fitting. Around epoch 41, the validation loss decreases notably before increasing again at epoch 51, indicating that there are periods of improvement followed by over-fitting. The below graph showcases the evolution of training and validation accuracy across epochs. The training accuracy consistently increases, indicating effective learning, reaching around 94-97%. The validation accuracy also improves but fluctuates across epochs, reaching its peak at epoch 41 (73.11%). The gap between training and validation accuracy suggests the model might be over-fitting, as the training accuracy is significantly higher than the validation accuracy.

5.3 Baseline Comparison and Model Robustness

The baseline performance of the I3D model for 100-word subset obtained by our own training is plotted in Figure 8. While the baseline model demonstrates reasonably good results, the proposed approach improves classification accuracy across all evaluated metrics, particularly in Top-1 accuracy.

5.4 Class Distribution Analysis

To ensure balanced training, we analyze the class distribution of the 100-word subset from the WLASL dataset. Figure 9 highlights the frequency of samples per class. While some classes exhibit slightly higher sample counts, the overall distribution remains reasonably uniform, ensuring unbiased training.

5.5 Precision, Recall, and F1 Score Trends

The trends for weighted average precision, recall, and F1 scores across epochs are plotted in Figure 10. These metrics indicate steady improvement in the model's ability to balance sensitivity and specificity over time, with all three metrics converging toward optimal values.

5.6 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 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.

6 Final Project Plan

By systematically scaling the model to larger datasets and integrating advanced architectures like Vision Transformers, we aim to push the boundaries of sign language recognition, addressing linguistic variability and computational challenges effectively.

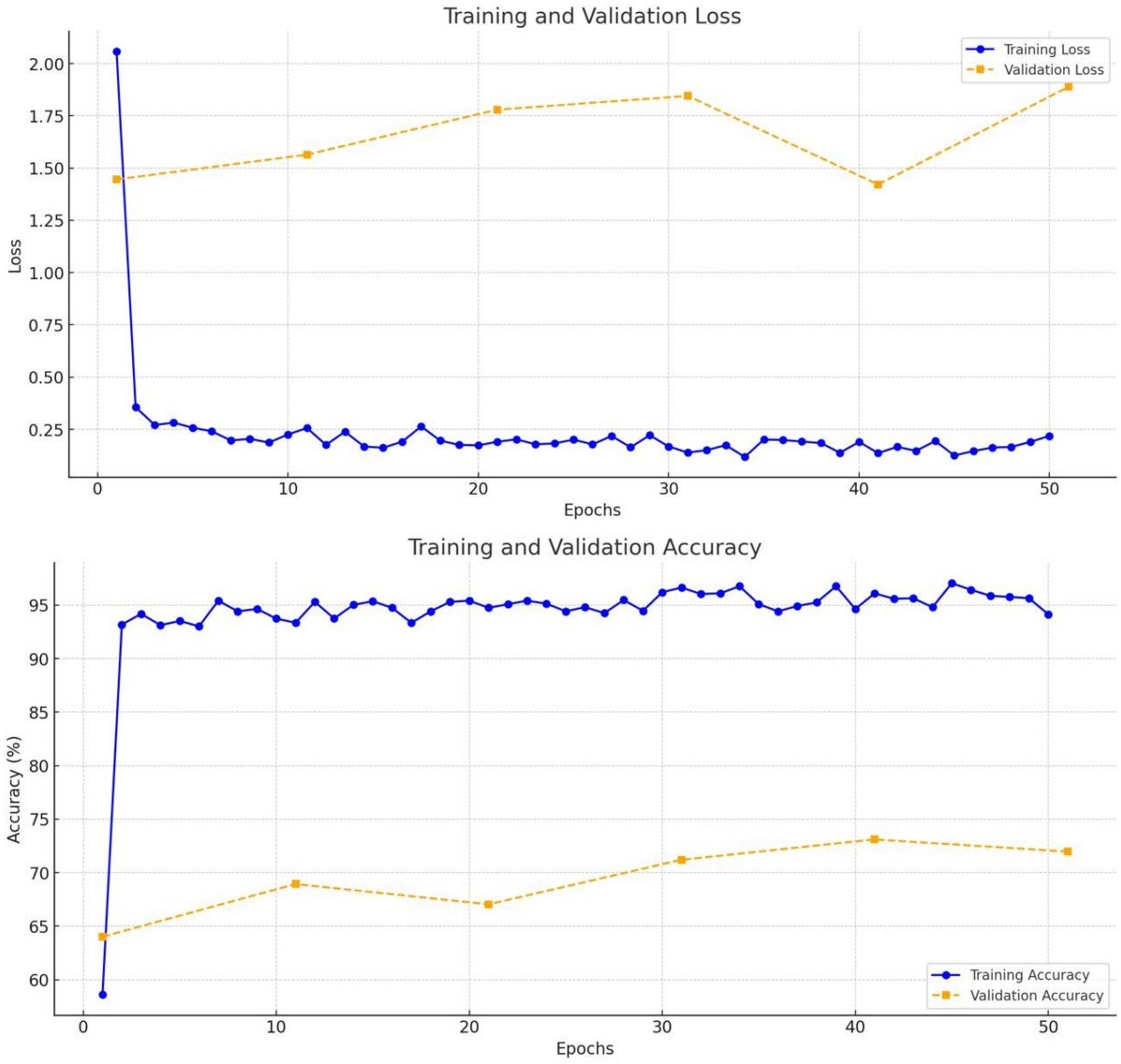

Figure 7: Training and validation loss/accuracy for the proposed model (I3D + Transformer).

6.1 Insights from Preliminary Experiments

The preliminary experiments provided key insights into the strengths and limitations of the proposed transformer-based model. While the transformer architecture significantly outperformed the baseline model, we observed several challenges:

- **Overfitting**: The validation loss showed fluctuations after certain epochs, indicating the need for stronger regularization and improved generalization strategies.
- **Increased Training Time**: The transformer-based model required longer training times and exhibited higher computational demands due to its complexity.
- **Memory Constraints**: Training transformers required significant memory resources, especially with larger batch sizes or longer video sequences.

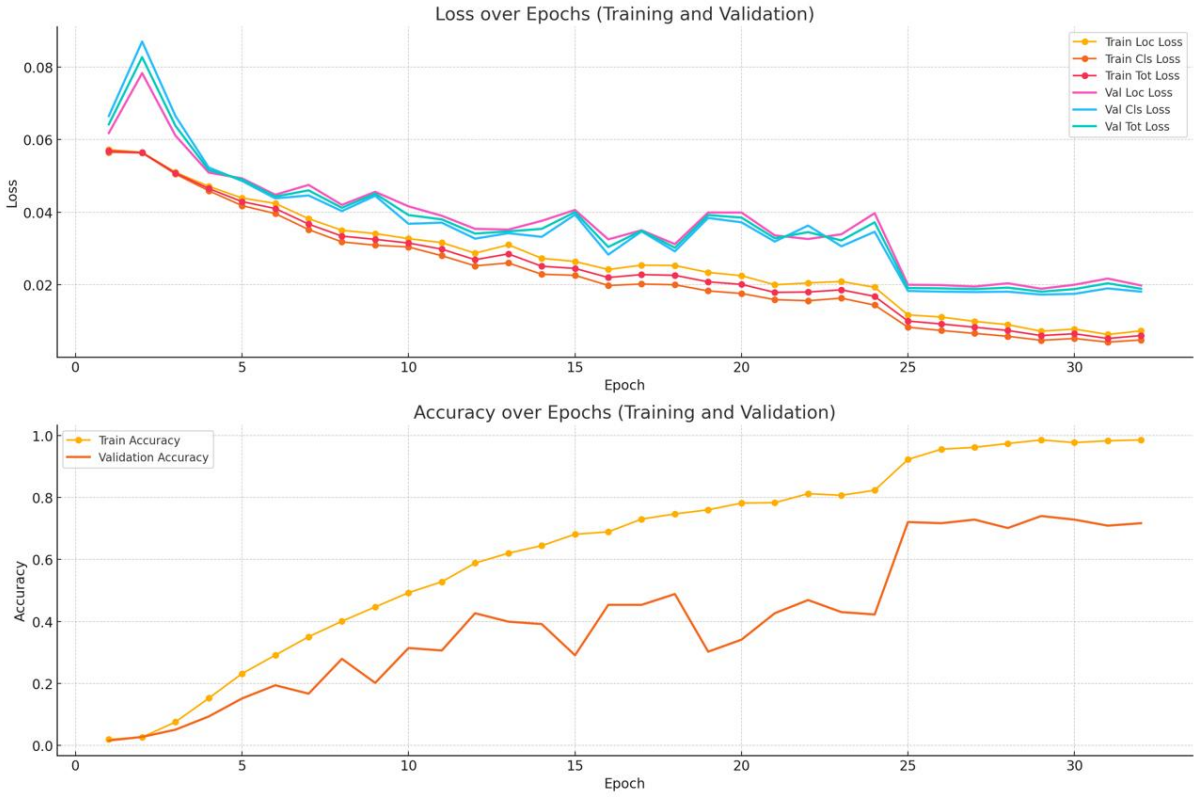

Figure 8: Training and validation loss/accuracy for the baseline model (WLASL).

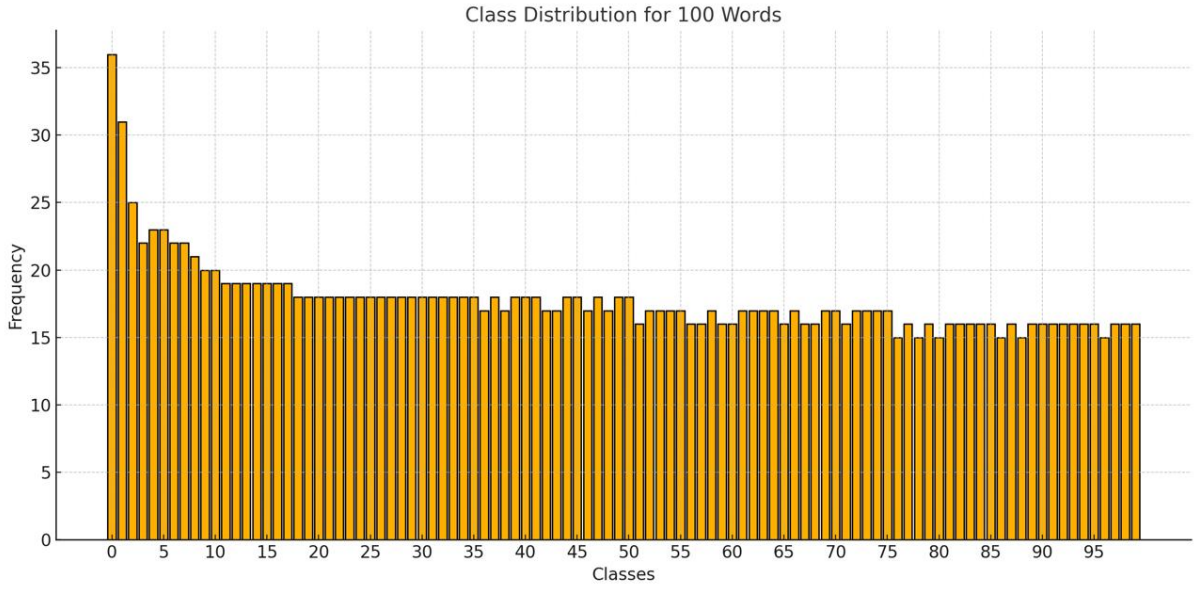

Figure 9: Class distribution of the 100 words in the WLASL dataset.

6.2 Proposed Adjustments for the Final Project

Based on these insights, the following adjustments will be made to refine the model for the final project:

- **Scaling to the Full WLASL Dataset**: We will train the model on the entire WLASL 2000 dataset to leverage a larger and more diverse dataset, with the goal of further improving

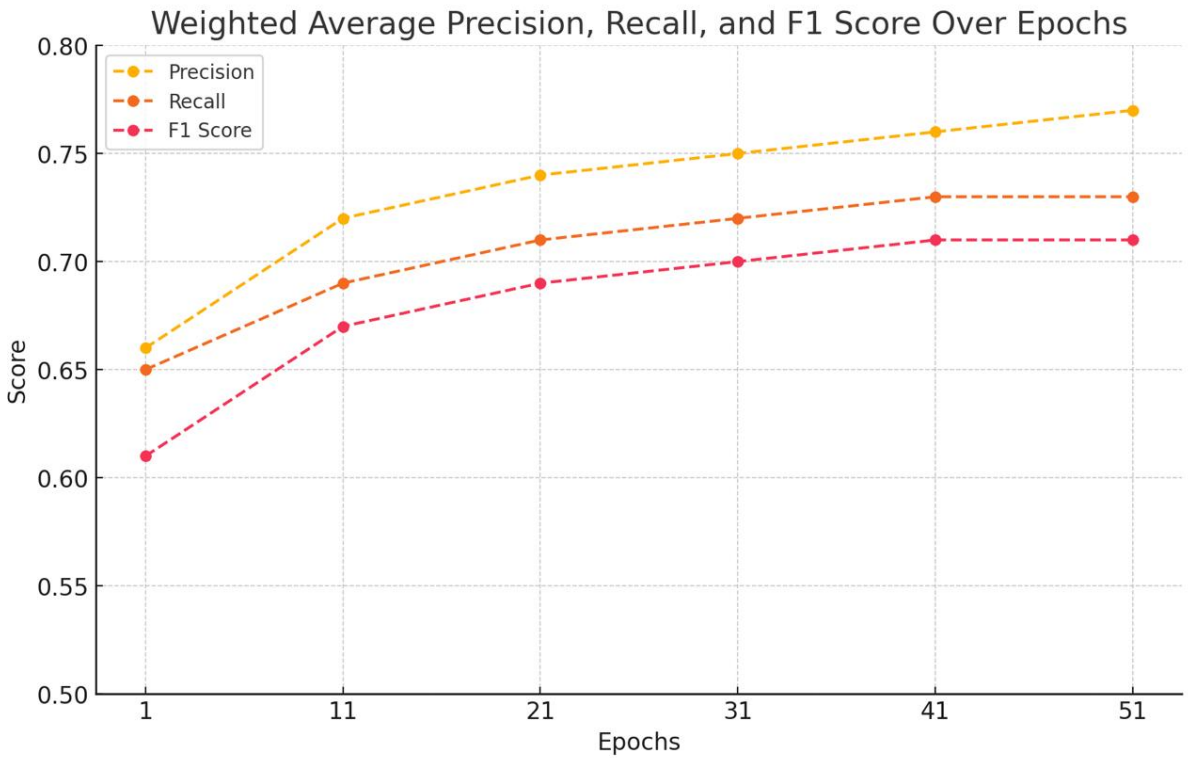

Figure 10: Weighted average precision, recall, and F1 score trends for the proposed model.

performance and generalizability over the baseline.

- **Vision Transformer Integration**: Instead of the vanilla transformer, we will explore the use of a vision transformer (ViT), which is specifically designed for handling spatial data. ViTs can effectively process image patches with self-attention, offering potential advantages such as:
  - Better spatial-temporal feature extraction for video data.
  - Improved performance on larger datasets compared to standard transformers.

6.3 Future Scope

- **Efficiency Optimization**:
  - Model Compression: Reduce the size of the model to make it lighter and more memory-efficient while retaining accuracy.
  - Inference Speed: Focus on optimizing the model for faster inference, which is critical for real-time applications.
- **Real-time Inference Pipeline**: In the final step, we plan to develop a pipeline to capture live data (e.g., from a webcam or similar input device) and enable real-time inference using the trained model. This will make the system suitable for practical deployment scenarios, such as live sign language recognition.

7 Author contributions

The author contributions are mentioned in Table 2.

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

Table 2: 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.