Tracking ASL Gestures Using Autoencoders

Katyani Mehra
Department of Computer Science
Rochester Institute of Technology
Rochester, NY 14623
km7872@rit.edu

Abstract

Tokenization is a critical preprocessing step for language models, as it enables the breakdown of language into smaller units, facilitating tasks such as translation, glossing, and recognition. However, manual tokenization and glossing of ASL videos is a time-consuming process, often taking hours even for short clips. This project explores the use of Vector Quantized-Variational Autoencoders (VQ-VAE), to track ASL gestures over time. Since ASL is a continuous language, a single frame or body part does not always correspond to a single token. Rather, each token may span multiple frames and involve multiple body parts. By passing ASL videos through the VQ-VAE we obtained a mutual information score of 0.98, indicating a strong dependency between gestures and tokens. This project demonstrates the successful implementation of a functional ASL tokenizer.

1 Introduction

Sign languages such as American Sign Language (ASL) differ from spoken languages as they rely on visual elements such as handshape, location, and movement [1], rather than auditory elements like speech. These elements, commonly referred to as tokens are used in language recognition, translation, and generative tasks with the help of machine learning.

Consider the phrase *long time*, as signed by an ASL user compared to existing systems. While current systems typically break this phrase down into two separate signs, one for *long* and another for *time*, ASL expresses *long time* as a single, continuous gesture. The key difference is that existing systems treat *long* and *time* as distinct, independent concepts having separate signs. In contrast, True ASL combines both concepts into one unified gesture. This limitation in current systems is largely due to insufficient sign language tokenization.

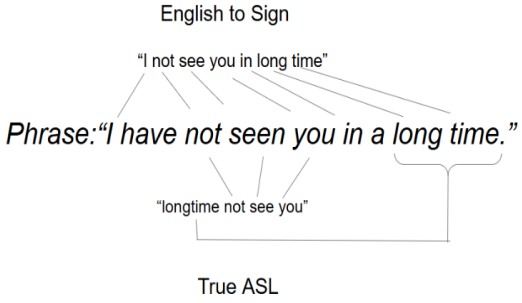


Figure 1: Current systems fail to capture ASL grammar

To address these challenges, we propose using Vector Quantized-Variational Autoencoders (VQ-VAE) to track ASL gestures over time. Although VQ-VAEs are often advertised as generative models, their core function is to compress information into a compact representation [7]. This compressed representation can later be decoded to reconstruct the original input. In our approach, we apply this autoencoder technique to ASL videos, aiming for the compressed output to yield discrete tokens that effectively capture the distinctive features of each gesture.

For our experiments, we used the MS-ASL dataset [4], which includes ASL phrases performed by various signers. The dataset includes 1229 video samples for training, 498 for validation, and 242 for testing.

To extract pose information from videos we initially used OpenPose [5]. However, it lacked 3D pose data due to the absence of depth information in our dataset. Therefore, we switched to MediaPipe Holistic [6], which provides 3D pose estimation and captures over 500 key points per frame, tracking movements of the body, face, and hands. This pose data was first normalized to account for variation among signers and then fed into our VQ-VAE model.

We evaluated our approach using Mutual Information (MI) score, to measure the dependency between gestures and the tokens generated by the model. Our results yielded an average MI score of 0.98 for all body parts, indicating a strong correlation between the original ASL gestures and the learned tokens.

2 Approach

2.1 Overview

In this paper, we propose tokenizing ASL videos using a Vector Quantized-Variational Autoencoder (VQ-VAE), which extends the traditional autoencoder by incorporating a codebook to map input data to a set of discrete, quantized representations. The process begins by passing videos from the MS-ASL dataset through a pose estimation model to extract key pose points. These pose points are then used to train the VQ-VAE. During training, the model learns to encode input data into discrete tokens by referencing entries in the codebook. Once trained, this codebook serves as a set of tokens that effectively capture the essential features of ASL gestures.

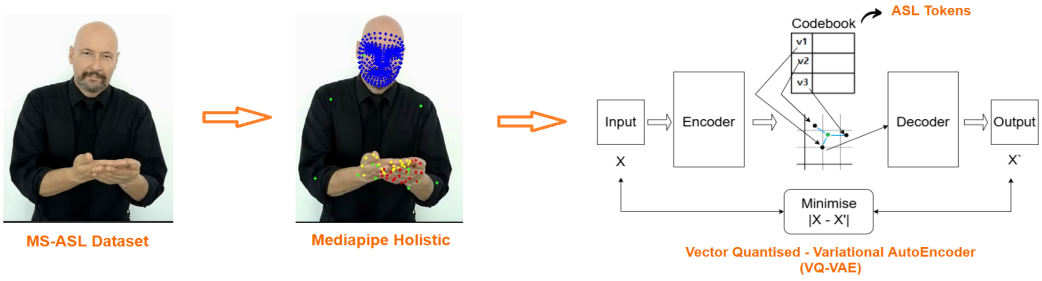


Figure 2: ASL Tokenization Pipeline

2.2 Dataset

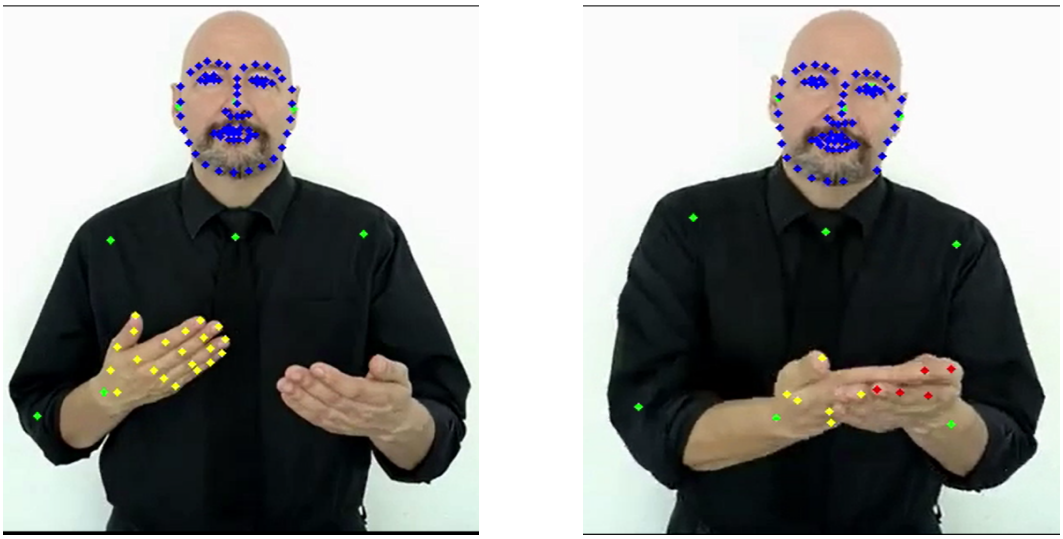
While many ASL datasets exist, we required one that captures continuous ASL gestures, which are essential for modeling natural sign language. The MS-ASL dataset meets this need, featuring 78 different signers and approximately 25,000 samples of ASL phrases. Each entry includes the video URL, the start and end times of the gesture sequence, and the corresponding gesture label.

However, a significant portion of the dataset was no longer accessible due to videos being removed or made unavailable on YouTube. As a result, we were able to retrieve approximately 1,900 ASL phrases across 456 videos. The difference between the number of phrases and videos stems from the fact that many videos contain multiple continuous phrases, reflecting the natural flow of ASL more accurately.

2.3 Pose Estimation

Rather than relying on raw pixel data, we use joint-based pose information as input to our VQ-VAE model. To extract these joint positions from ASL videos, we employed pose estimation models, which detect key body landmarks such as hands, face, and body. These models typically output a set of coordinates representing joint positions, often accompanied by confidence scores. Some models generate heatmaps for each joint, where the peak of each heatmap indicates the most likely joint location.

Our goal was to use a single model capable of capturing all relevant joints (hands, body, and face), minimizing the complexity of combining outputs from multiple specialized body detectors. We experimented with two models: **OpenPose** and **MediaPipe Holistic**.



(a) Failure to detect left hand

(b) Hand Keypoints missing due to occlusion

Figure 3: Challenges in OpenPose: Hand detection failure and missing keypoints during occlusion

OpenPose provides approximately 137 2D keypoints, processes videos frame-by-frame and outputs a JSON file for each frame, including x, y coordinates and confidence scores for detected joints. However, we encountered several issues: In some frames, hand keypoints were missing. Initially, we thought this was due to occlusion, but we later observed this occurring in many other cases as well. This presents a challenge for models like VQ-VAE, which do not handle missing data effectively.

To overcome these limitations, we switched to **MediaPipe Holistic** that tracks 543 keypoints, including 3D coordinates for the face, hands, and body. MediaPipe showed better consistency in hand detection.

2.4 Normalization and Interpolation

To account for variations in size and position due to camera distance and frame placement, we normalize the landmarks to a standardized coordinate system. In this system, the left shoulder is mapped to 1, the right shoulder to -1, and the center of the shoulders is set to 0. As a result, all left limbs will have positive x-coordinates, while right limbs will have negative x-coordinates. The body is then scaled to a consistent size by calculating the distance between the left and right shoulders and adjusting the scale accordingly. This normalization process is applied to all videos. Figure 4 is an example showing two different individuals, both using the same coordinate system after normalization.

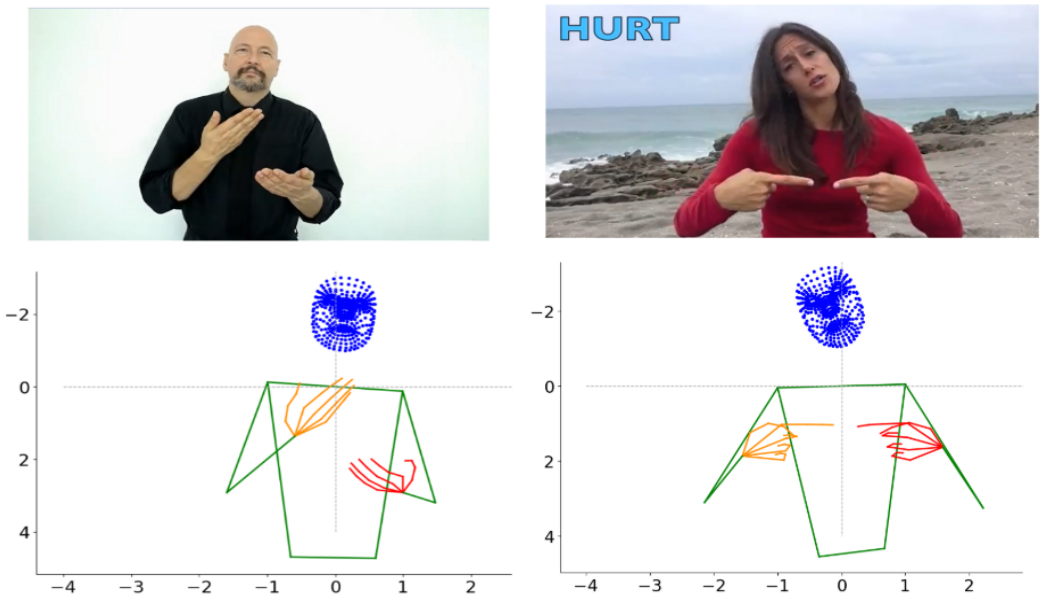


Figure 4: Normalized keypoints across two individuals

Additionally, we are not considering all the joints from the pose estimation model. Instead, we focus on 21 joints for each hand, 6 for the upper body, and the remaining 468 for the face, totaling 516 joints. These joints are passed frame-by-frame into the VQ-VAE model. We limit the body joints to 6 because we are not considering the lower half of the body. Figure 5 shows the joints used for the body and hand.

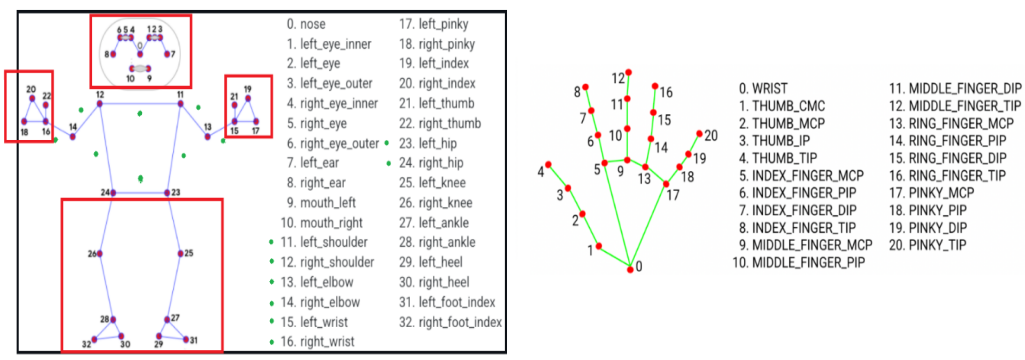


Figure 5: Mediapipe hand and body landmarks, with the landmarks in red squares being ignored

For OpenPose, we needed to find a way to handle missing data, particularly when some joints of a body part are not tracked across consecutive frames. In recent implementations of VQ-VAE, the authors addressed this issue by setting missing points to zero. However, in our case, zero corresponds to the center of the shoulders, which makes it an unsuitable placeholder for missing data. As a result, we opted to use linear interpolation to fill in the missing joint positions. Specifically, we interpolate across the video, but only for those frames where at least one joint of the corresponding body part is available. While we don't necessarily need this for MediaPipe data, it is a nice-to-have feature as part of our pipeline.

Even after interpolation, we still encounter some frames where not all 516 joints are detected. These frames could be filler frames, such as those at the start or end of a video, or in instructional videos where the word appears for a few seconds before a person shows up. Additionally, there are cases

where the pose estimation model simply fails to detect certain joints. These frames are irrelevant and problematic for the VQ-VAE model, so we choose to discard them.

2.5 Vector Quantized Variational Autoencoders

Imagine you have a 256 x 256 image - a traditional Autoencoder would compress this image into a smaller, simpler representation through the encoder. For example, this might be a 4 x 1 representation of the input image. This compressed version is then passed to the decoder, which attempts to reconstruct the original image. The quality of the reconstruction is assessed using a loss function Mean Squared Error (MSE), which calculates the difference between the original image (x) and the reconstructed image (x'). The model aims to minimize this difference during training.

In a Variational Autoencoder (VAE), instead of mapping the input to a single vector, the encoder creates a distribution of possible latent variables (usually a Gaussian distribution). The decoder then uses a sample from this distribution to reconstruct the input. The loss function in VAEs has two parts: the reconstruction loss (the difference between the original and the reconstructed image) and a regularization term (KL divergence), which ensures the learned distribution is close to a standard Gaussian.

VQVAE's take a different approach. Instead of using a continuous distribution, VQVAE uses a set of discrete vectors stored in a *codebook*. The encoder finds the closest vector in the codebook for each input, and the decoder uses that vector to recreate the input. The loss function in VQVAE includes two parts: the reconstruction loss (how well the input is reconstructed) and the commitment loss, that specifically prevents the encoder from shifting between codebook vectors too often.

The main advantage of VQ-VAE is its ability to work with discrete data, making it ideal for tasks like ASL gesture tracking, where gestures can be represented as distinct tokens or symbols. Another significant advantage is that VQ-VAE operates in an unsupervised learning manner. We do not explicitly label which gesture it is training on, instead, we allow the model to identify movements on its own.

2.6 Sample to Token mapping

After training our VQ-VAE, we obtain a codebook that represents the ASL tokens. We can then run sample gestures through the model to see which tokens they are mapped to. Figure below shows an example of this mapping.

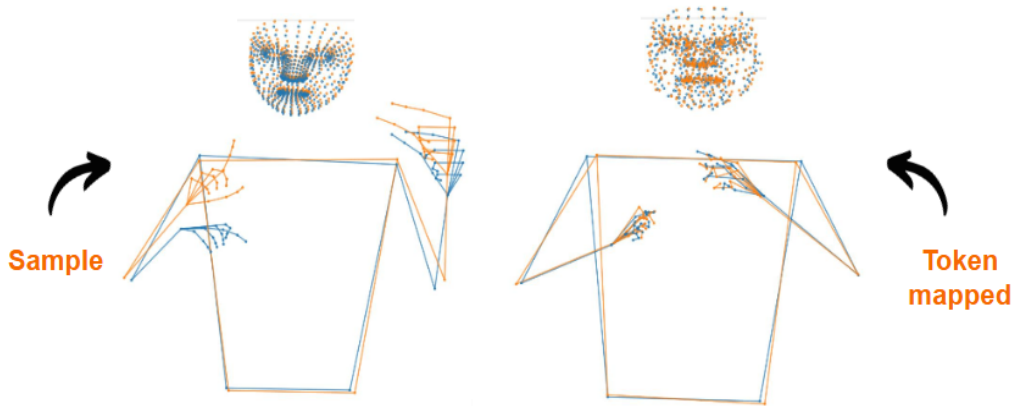


Figure 6: Sample to token mapping

2.7 How many Frames make a Token?

As mentioned earlier, one token can cover several frames. Since the videos in our dataset are mostly recorded at 30 frames per second (fps) and on average, people perform 2 to 3 gestures per second, so each gesture lasts around 10 to 15 frames, we decided to chose to use 7 frames, which is about

0.23 seconds. We split each video into 7 non overlapping frames each having 516 keypoints in x,y,z channels

2.8 What about Single-Hand gestures?

Many ASL gestures are performed using a single hand, either left or right, depending on the signer's dominant hand. This raises the question whether it would make sense to train a separate model specifically for single-hand gestures. It also leads to the idea of whether each body part should have its own VQ-VAE, since any of them could contribute to forming a token.

During our analysis, we observed that the number of facial landmarks (468) is significantly higher than the number of landmarks for the hands and body combined (21 for each hand and 6 for the upper body). This large imbalance caused the model to focus more on the face more than the hand.

To address this, we decided to split the training into three parts, creating a separate VQ-VAE for each body region: face, hands, and upper body.

3 Results

We developed a tokenizer for gesture sequences using a VQ-VAE-based model. To evaluate how meaningful the tokens are, we used Mutual Information (MI) score. MI measures how strongly the tokens are related to specific gestures. We use scikit-learn's `mutual_info_score` to compute the MI between gesture labels and tokens. The computed MI scores for different body parts are shown below:

- **Body**: 0.59
- **Face**: 1.27
- **Left Hand**: 1.04
- **Right Hand**: 1.02

These results show that MI scores for the face and hand landmarks are more than the body. This is expected since the body has fewer keypoints (only 6 per frame) and less variability in movement compared to the face and hands, which have more complex and dynamic gestures. The average MI score for all body parts is 0.98.

4 Future Work

This research is an important step forward, although much of it was unplanned, with decisions made along the way. Here are some areas of improvement that we would like to highlight:

- More 3D data: As we observed, the MS-ASL dataset was supposed to contain 25,000 gestures, but only about 1,900 were available for use. With more data, the tokens generated by the model would be more refined and just more.
- Experimenting with a different Pose Model: When analyzing a few samples processed through MediaPipe and OpenPose, we noticed that some joint positions would jump. This is likely due to both models processing videos frame-by-frame. Upon further analysis of bone lengths, we observed inconsistent bone tracking for many joints. Figure 7 is a video analysed for inconsistent bone tracking.

  The bone length calculation is as follows:

  $$\text{Length} = \|\text{xyz}[i] - \text{xyz}[j]\|$$

  where i and j represent two keypoints (joints).
- 516 Joints per Frame, can we do better?: Previous research has applied dimensionality reduction techniques like Principal Component Analysis (PCA) to hand keypoints, successfully reducing the dimensionality to just 6 or 7 components. Applying similar techniques to the face should help reduce the number of dimensions significantly.

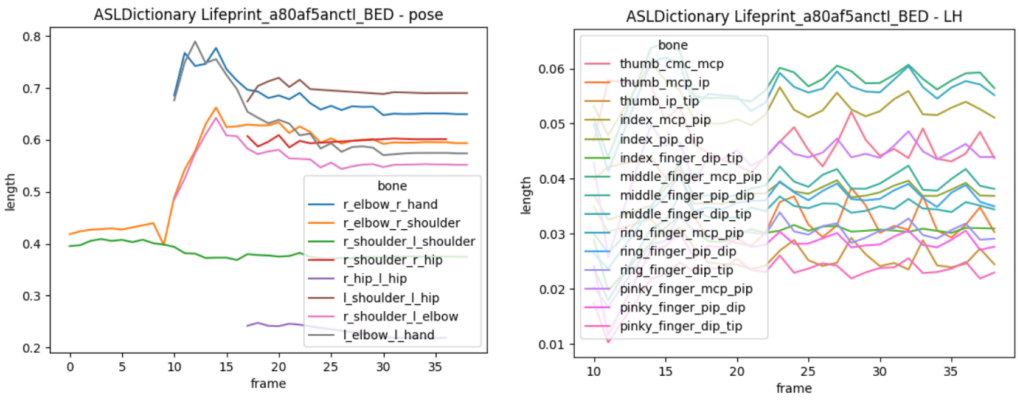


Figure 7: Bone lengths fluctuate across frames.

- 7 non overlapping frames: We split each video into 7 frames without any overlap. Each 7-frame chunk is treated as a possible token. In the future, we plan to explore overlapping windows, and a variable frame chunck depending on the fps of different videos.

5 Conclusion

This paper demonstrates that we can create a tokenizer using VQ-VAE, trained on pose information for ASL videos. We report an average mutual information score of 0.98, indicating a strong correlation between tokens created and the gesture. This work will inherently improve ASL translation, recognition, and generative models as they all rely on a tokens.

References

[1] A. Er-Rady, R. Faizi, R. O. H. Thami and H. Housni, "Automatic sign language recognition: A survey," 2017 International Conference on Advanced Technologies for Signal and Image Processing (ATSIP), Fez, Morocco, 2017, pp. 1-7, doi: 10.1109/ATSIP.2017.8075561.

[2] Kahlon, N.K., Singh, W., "Machine translation from text to sign language: a systematic review", Univ Access Inf Soc 22, 1–35 (2023). https://doi.org/10.1007/s10209-021-00823-1

[3] Orbay, A. and Akarun, L., "Neural Sign Language Translation by Learning Tokenization", arXiv.org, 2020. Available at: https://arxiv.org/abs/2002.00479

[4] Joze, H.R.V. and Koller, O., "MS-ASL: A Large-Scale Data Set and Benchmark for Understanding American Sign Language", arXiv:1812.01053 [cs] [Preprint]. Available at: https://arxiv.org/abs/1812.01053.

[5] Cao, Z. et al., "OpenPose: Realtime Multi-Person 2D Pose Estimation using Part Affinity Fields", arXiv:1812.08008 [cs] [Preprint]. Available at: https://arxiv.org/abs/1812.08008.

[6] Google AI, "Holistic landmarks detection task guide", 2024. Available at: https://ai.google.dev/edge/mediapipe/solutions/vision/holistic_landmarker.

[7] "Neural Discrete Representation Learning", [Online]. Available: https://proceedings.neurips.cc/paper/2017/hash/7a98af17e63a0ac09ce2e96d03992fbc-Abstract.html.