Video Analytics: **Human Action Recognition (HAR)**

Team Members:

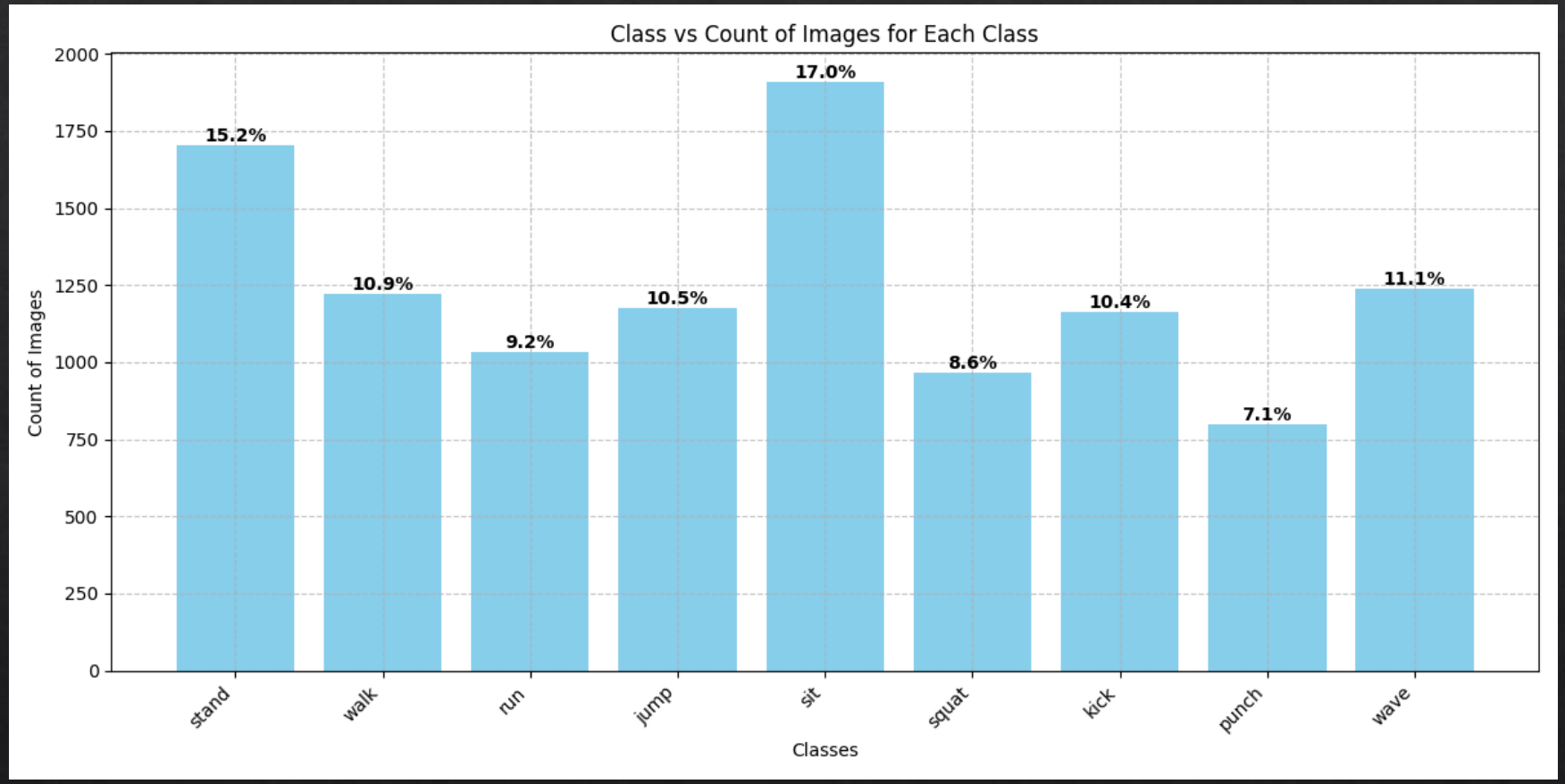
Unnath Chittimalla, Velidanda Krishna Sai

Why HAR?

- **Importance of HAR (Human Action Recognition)**
- **Applications in Real World:**
  - Smart surveillance (detecting suspicious activity)
  - Healthcare (monitoring elderly movements)
  - Sports analytics (tracking athlete performance)
  - Human-computer interaction (gesture-based controls)
  - AR/VR (gesture-based interactions)
- **Challenges in HAR**
- **Variability in Actions** (e.g., different styles of walking or jumping)
- **Occlusions & Camera Angles** (Partial visibility can affect accuracy)
- **Multi-Person Tracking Issues** (Handling overlapping people)
- **Computational Complexity** (Real-time processing on edge devices)

Current Dataset

- Class-Wise Data Analysis
- Training/Testing Split: 70 train / 30 test
- ~11k Total Images
- We used stratified approach while early stop training to ensure validation of all classes equally

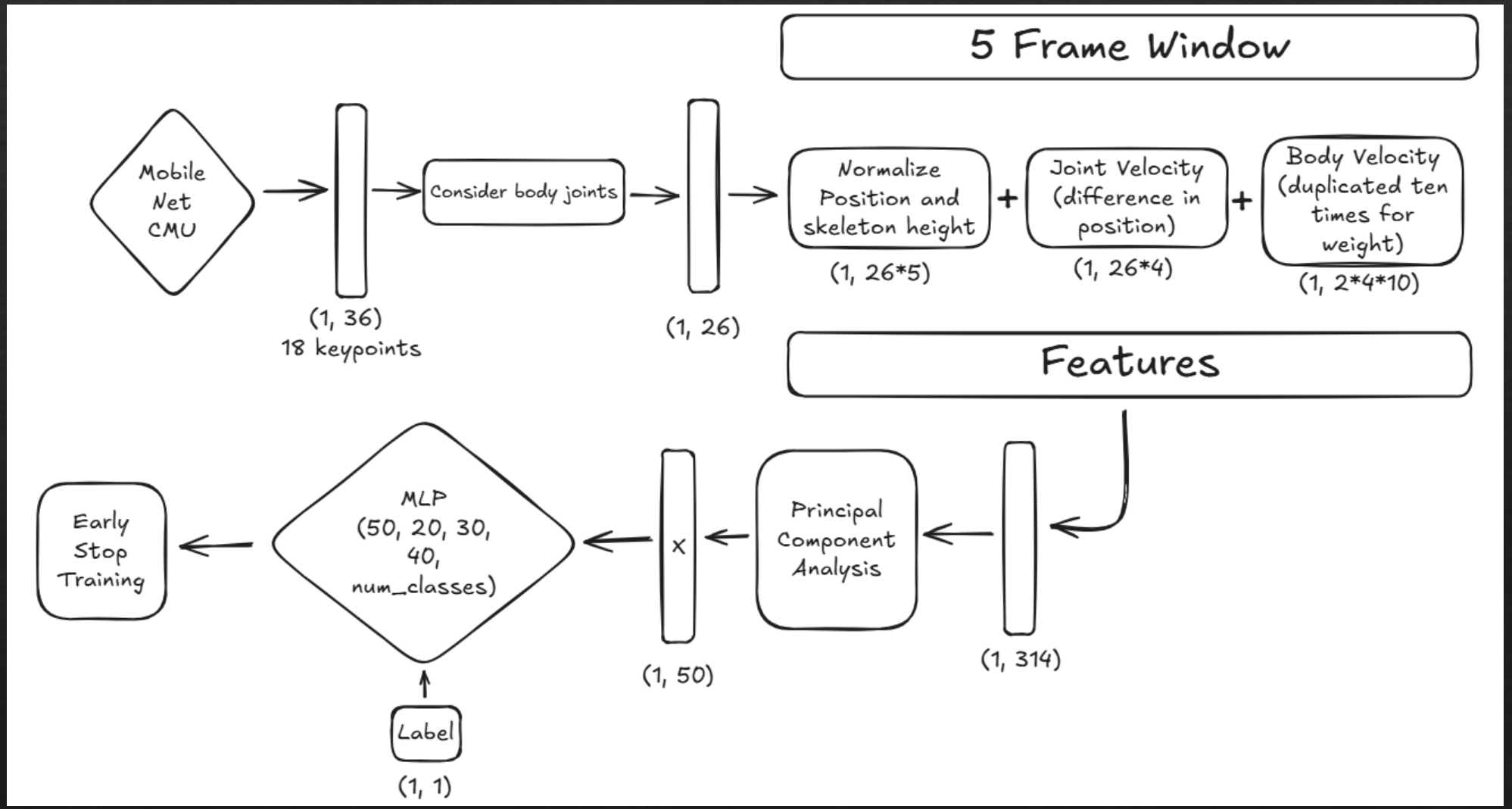


Methodology: **PoseNet + Tracking + MLP**

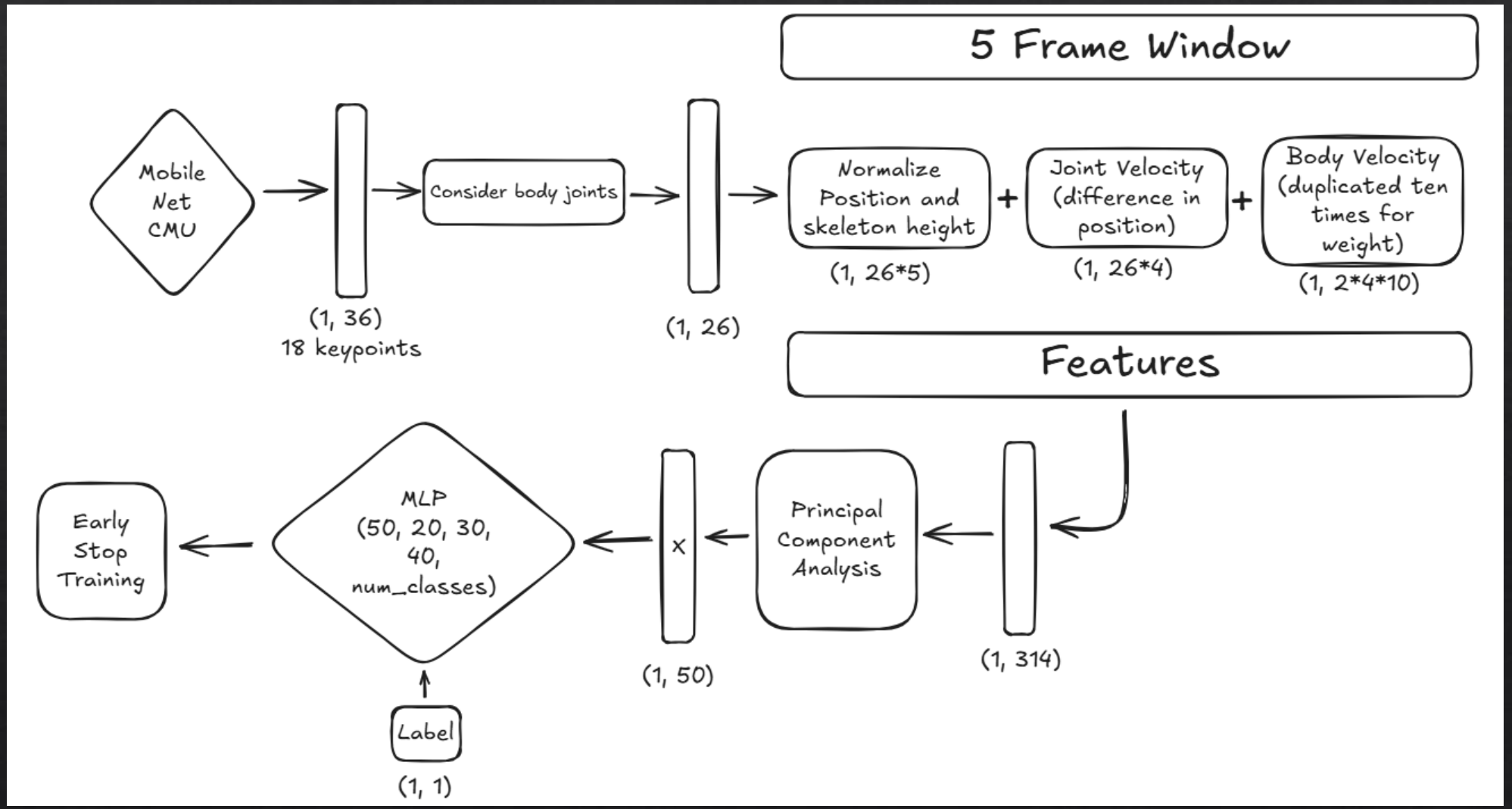
- We use a modular approach that extends from a single person pose and action recognition method.

- Workflow diagram explaining the following.
  - Role of OpenPose (MobileNet-CMU)
    - Detect persons using a person detector CNN like Faster-RCNN/YOLO.
    - Perform Pose Estimation and extract the 18 skeleton keypoints.
    - Get vector of shape (1, 36) where even index is x-position, odd index is y-position.
  - Processing and Feature Extraction
    - Perform Pose Estimation and extract the 18 skeleton keypoints.
    - Get tensor of shape (1, 36) where even index is x-position, odd index is y-position. Then use tracking to distinguish different people, store person id, video idx, frame idx, label, path to image so now we have tensor of shape (1, 36+5)
    - Filter out good skeletons, normalize joints position, calculate body and joint velocities using window of 5 frames. Also add noise if needed. Feature Vector shape is (1, 314 because 5*26(position) + 26*4(differences) + 2*4(neck velocity*10 times repeat))
  - MLP Layer (3 Hidden Layers with 20, 30, 40 nodes in respective layer)
    - PCA (Principal Component Analysis) is used to reduce the features from (samples, 314) to (samples, 50)
    - This is fed into the MLP and output nodes are equal in number to required classes.
    - Early stopping is used to automatically stop training when loss/accuracy converges.

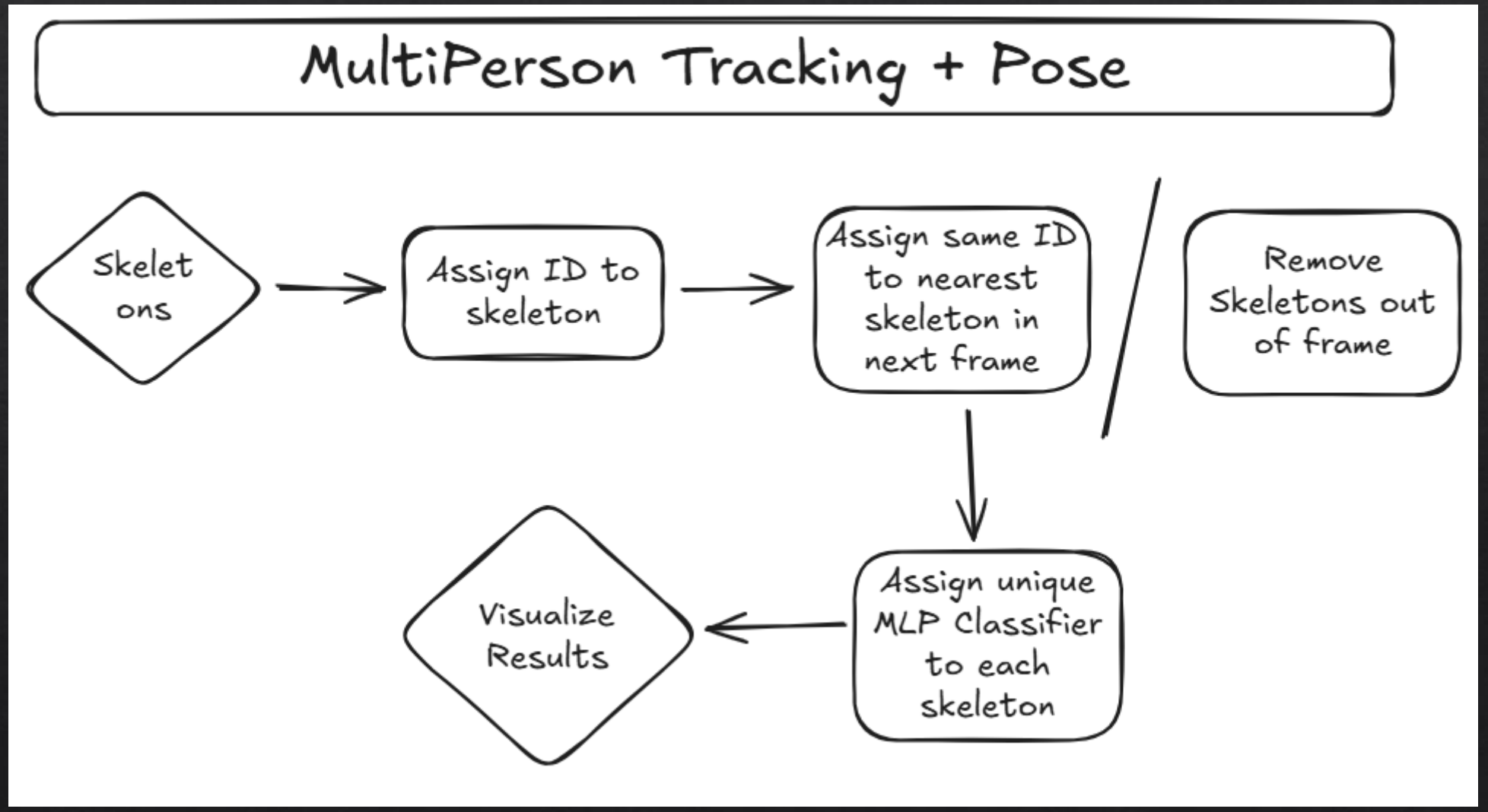
Feature Extraction and MLP



Feature Extraction and MLP



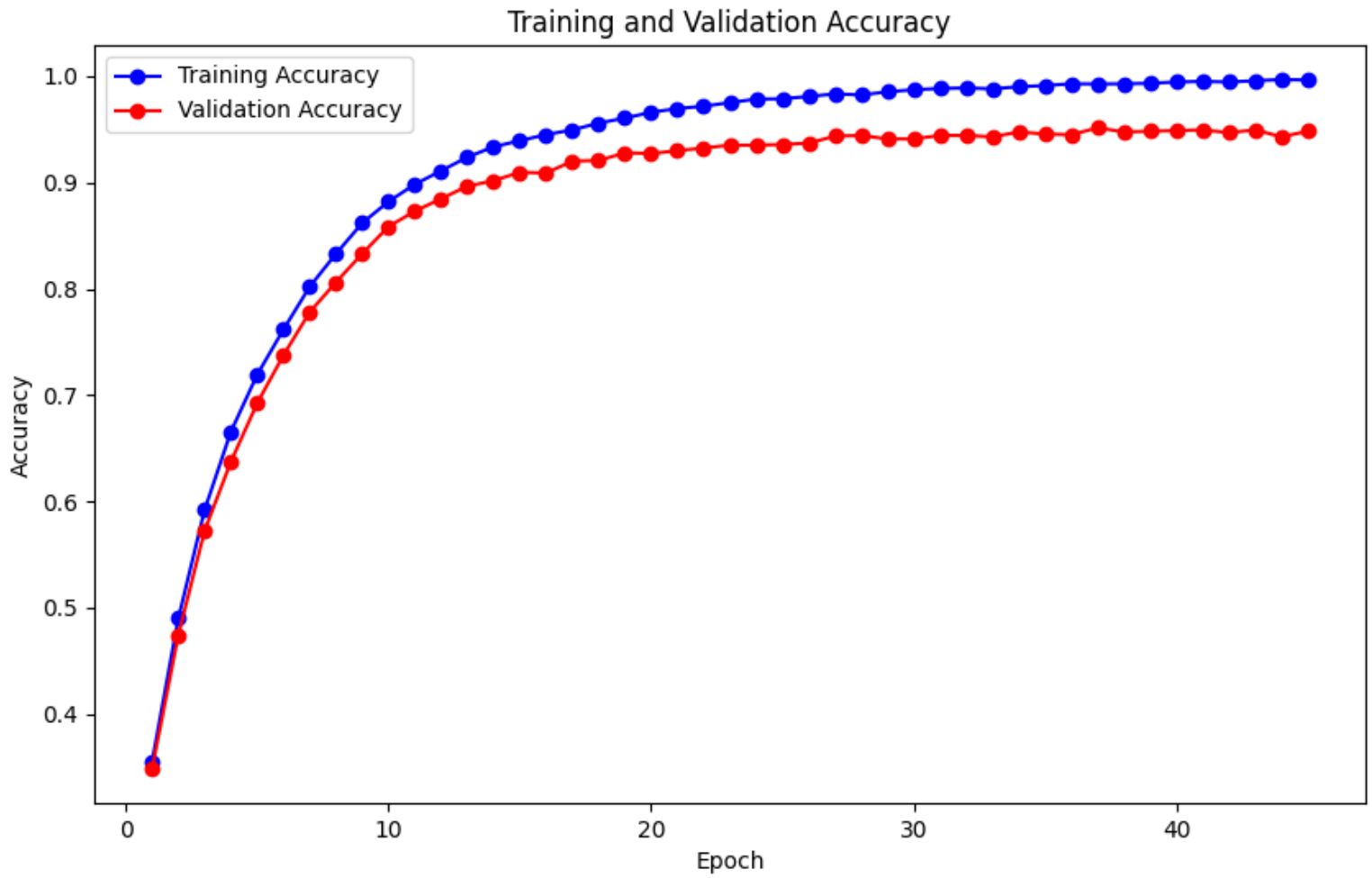
MultiPerson Tracking



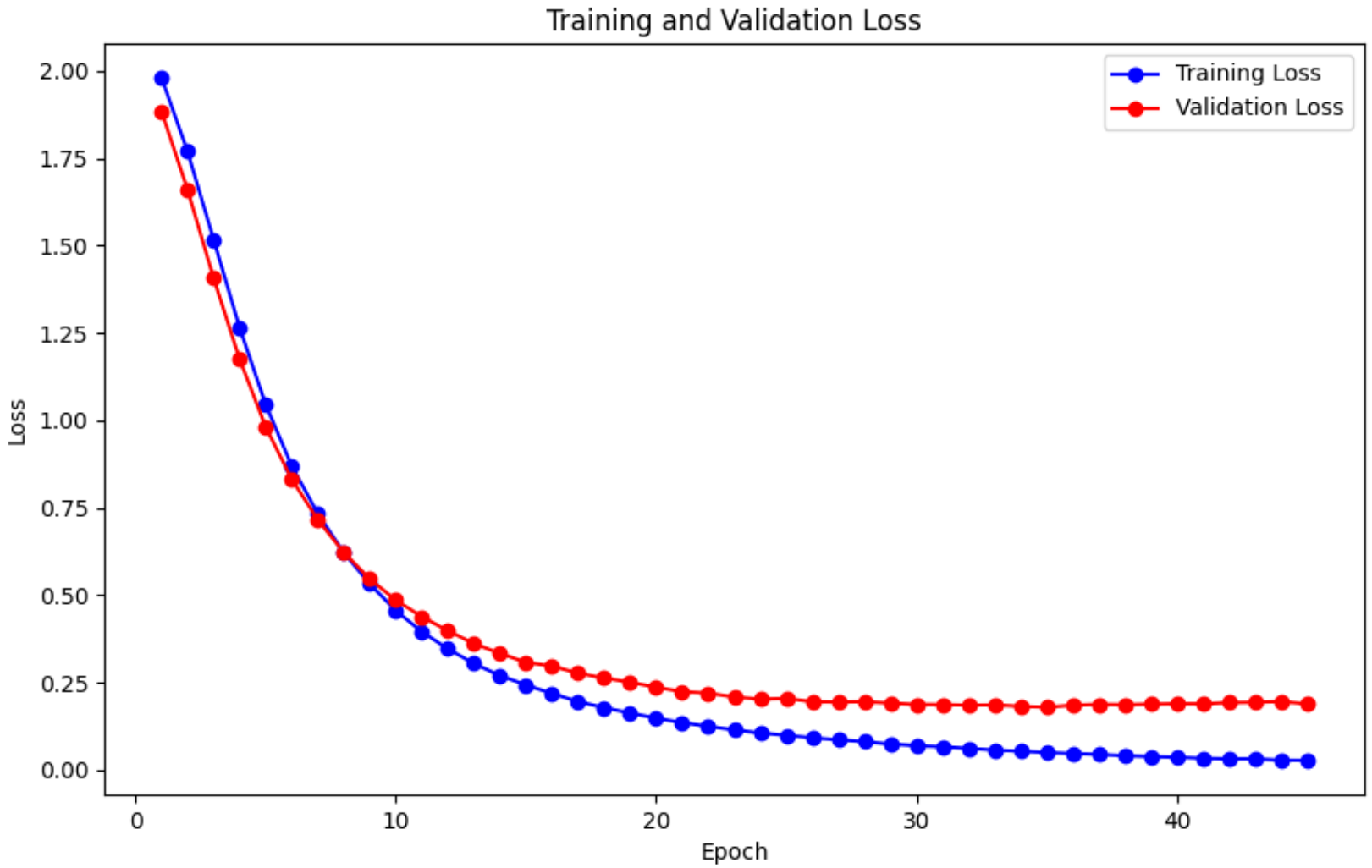
Results - Summarized

```
Epoch 045: Train loss 0.0270, Train acc 0.9966, Val loss 0.1888, Val acc 0.9483, Val prec 0.9468, Val rec 0.9460
Early stopping triggered.

Evaluating on test set ...
              precision    recall  f1-score   support

       stand       0.94      0.95      0.94       505
        walk       0.90      0.93      0.91       364
         run       0.94      0.95      0.95       307
        jump       0.99      0.97      0.98       331
         sit       0.98      0.98      0.98       551
       squat       0.97      0.95      0.96       287
        kick       0.91      0.88      0.90       319
       punch       0.94      0.93      0.94       238
        wave       0.95      0.93      0.94       369

    accuracy                           0.95      3271
   macro avg       0.94      0.94      0.94      3271
weighted avg       0.95      0.95      0.95      3271
```

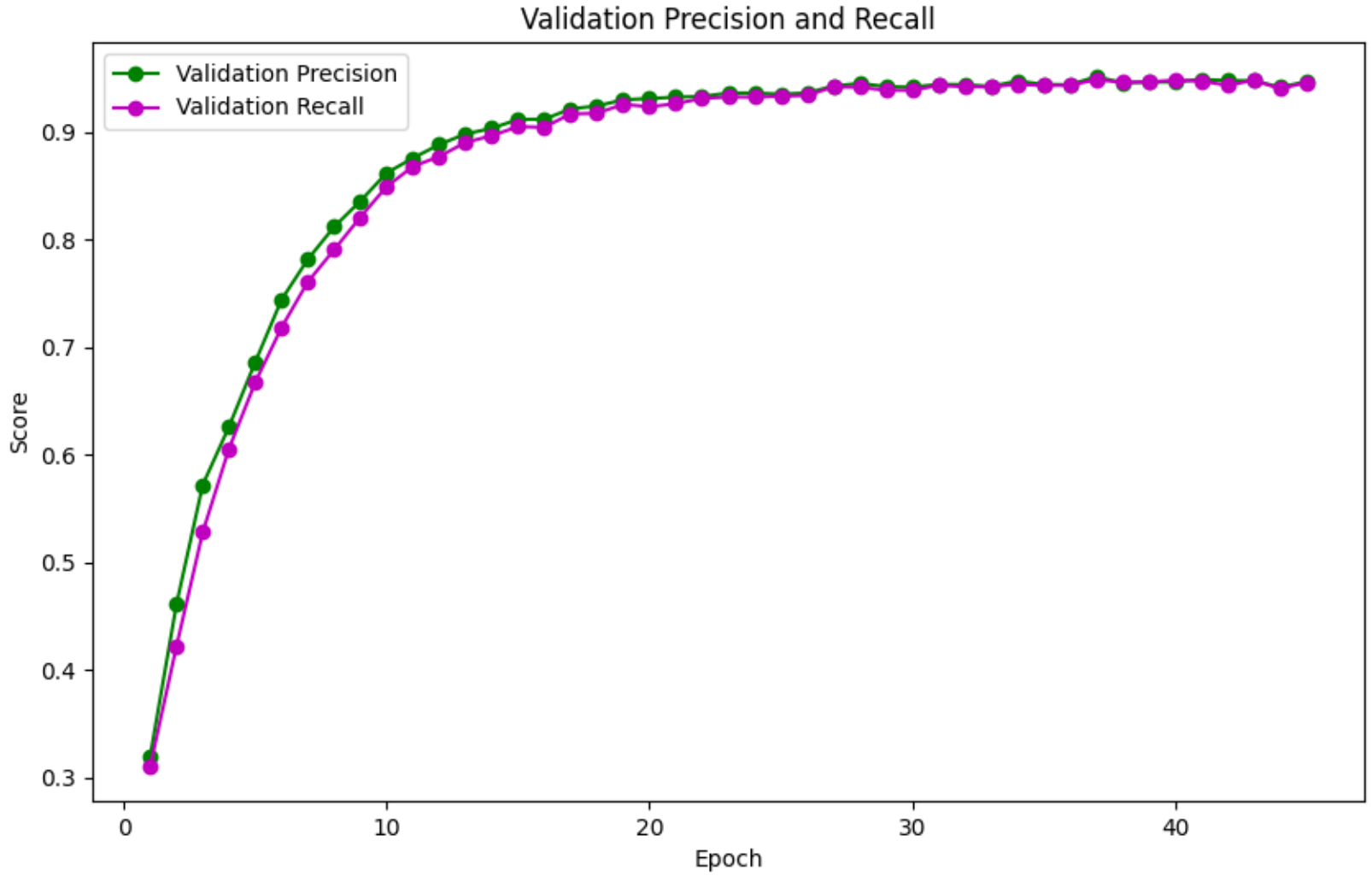
Results – Training/Validation Accuracy



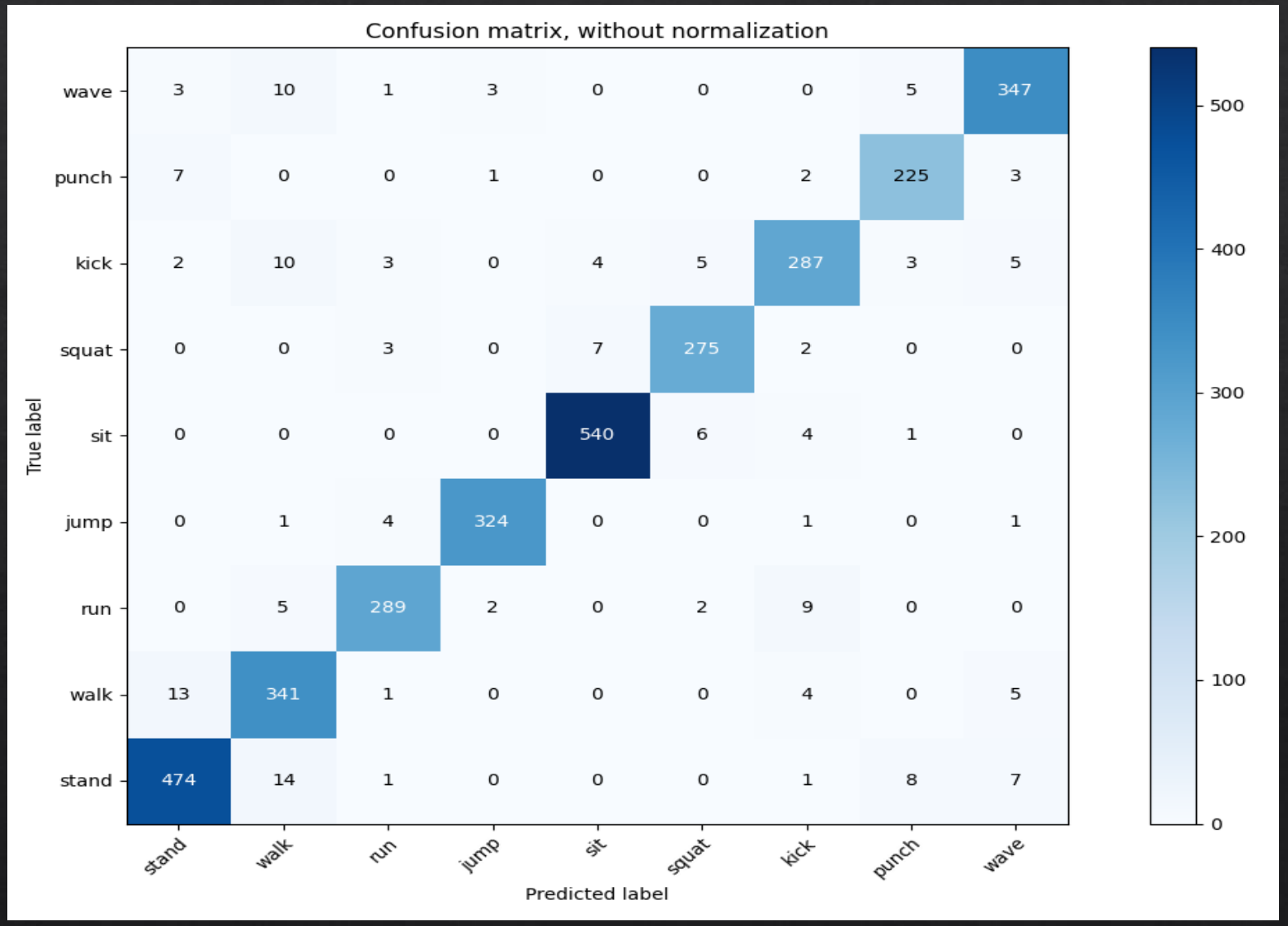
Results – Training/Validation Loss



Results – Precision/Recall



Results – Confusion Matrix



Methodology: **ViViT on Kinetics400**

- We build upon google's model on huggingface meant for the kinetics400 dataset.
- Workflow diagram explaining the following.