Preliminary Documentation for Gesture Recognition Project

Bora ILCI and Kaan Emre KARA

Group 10

Instructor: Valeriya Khan

May 12, 2025

1. Introduction

This project aims to develop a neural network-based model for hand gesture recognition. To ensure scientific rigor, the model's performance will be compared against state-of-the-art solutions, and comprehensive research on methods and datasets will be conducted.

2. Algorithm Descriptions

2.1 CNN + LSTM

Extract spatial features per frame using a 2D CNN (three 3×3 conv layers, batch normalization, ReLU, max pooling), then model temporal dynamics with an LSTM (128 hidden units) on 256-dimensional embeddings. Classification is performed with a softmax layer over 10 classes.

2.2 3D-CNN

Apply 3×3×3 convolutions on frame volumes to learn spatiotemporal features (C3D: 8 conv layers, 2 FC layers, softmax).

2.3 Transfer Learning

Use ResNet-18 pretrained on ImageNet to extract frame embeddings, aggregated via global average pooling or a lightweight LSTM.

3. Dataset Selection and Description

3.1 LeapGestRecog: GTI–UPM dataset from Kaggle with 10 gesture classes, ~2,000 sequences, ~100 grayscale frames at 84×84 resolution. Preprocessing includes frame extraction, resizing to 64×64, normalization to [0,1], and class-based organization.

3.2 Additional Data: SHREC hand gesture dataset (depth + RGB) for extensions; custom webcam collection (10 gestures × 50 sequences).

4. Libraries and Tools

PyTorch 1.13; OpenCV for preprocessing; NumPy & pandas for data handling; torchvision.transforms for augmentation; scikit-learn for data splitting and metrics; Matplotlib & Seaborn for visualization; TensorBoard for logging; flake8 for linting.

5. Experimental Plan

Data split: 70% train, 15% validation, 15% test with stratification.

Implement baselines: CNN+LSTM, 3D-CNN, optional ResNet-18 transfer learning.

Training: 30 epochs, batch size 16, Adam (LR=1e-3), CrossEntropyLoss, LR scheduler, early stopping (patience=5).

Hyperparameter search: LR {1e-2,1e-3,1e-4}, hidden sizes {64,128,256}, batch sizes {16,32}.

Benchmark: Compare F1, accuracy, training time, model size.

6. Visualization Methods

Plot learning curves (loss & accuracy vs. epochs); confusion matrix heatmap; bar charts for per-class precision, recall, F1; overlay predictions on sample frames; table of model parameters, FLOPs, and inference time.

7. Quality Measures

Accuracy: correct classifications / total samples. Precision: TP/(TP+FP). Recall: TP/(TP+FN). F1-Score: harmonic mean of precision and recall. Confusion analysis of misclassifications. Resource metrics: training time, memory usage, inference latency.