

# **SmartHome Gesture Control Application Project Part 2 Report**

**Author: Casey Tom**

**Class: CSE 535 Mobile Computing**

**Semester: Fall 2024 Session A**

**Date: 10/1/2024**

## **Approach to the Problem**

### **1. Understanding the Problem:**

- The task involves trying to identify inputted videos (test data) of hand gestures against videos of hand gestures previously recorded (training data).

### **2. Data Preparation:**

- Two dictionaries, `train_dict` and `test_dict`, were created to map gesture names to numbers for training and test data respectively.

### **3. Feature Extraction:**

- The `get_feature` function was implemented to extract features from the middle frame of each gesture video. This function utilizes `HandShapeFeatureExtractor` to extract features from the grayscale image of the frame.

### **4. Training Data Processing:**

- An empty list `train_data_list` was initialized to store training data.
- Each file in the training data folder was looped through, extracting gesture information from the file name and extracting features from the middle frame of the video.
- An instance of `TrainInfo` containing test gesture names, gesture number, and file gesture version number was created along with the extracted feature data and appended to `train_data_list`.

### **5. Test Data Processing and Gesture Recognition:**

- An empty list `out_data` was initialized to store the output data.
- Each file in the test data folder was looped through, extracting gesture information from the file name and extracting features from the middle frame of the video.
- The extracted features were compared with the training data using cosine similarity to identify the gesture with the highest similarity.
- The recognized gesture was appended to `out_data`.

## 6. Output Generation:

- The recognized gestures were written to a CSV file named Results.csv.

## Solution to the Problem

### 1. Define the TrainInfo Class:

- This class holds information about each gesture, including its name, version, number, and extracted features.

### 2. Create Dictionaries for Gesture Mapping:

- train\_dict and test\_dict map gesture names to numbers for training and test data respectively.

### 3. Implement the get\_feature Function:

- This function extracts the middle frame from the input video file, reads it as a grayscale image, and extracts features using HandShapeFeatureExtractor.

### 4. Process Training Data:

- Initialize an empty list train\_data\_list.
- Loop through each file in the training data folder.
- Extract gesture information from the file name.
- Extract features from the middle frame of the video using get\_feature function.
- Create an instance of TrainInfo with the extracted data and append it to train\_data\_list.

### 5. Process Test Data and Recognize Gestures:

- Initialize an empty list out\_data.
- Loop through each file in the test data folder.
- Extract gesture information from the file name.
- Extract features from the middle frame of the video using get\_feature function.
- Compare the extracted features with the training data using cosine similarity.
- Identify the gesture with the highest similarity ( minimum return values ) and append the result to out\_data.

### • Write Output to CSV:

- Write the recognized gestures to a CSV file named Results.csv.