SmartHome Gesture Control Application - Part 2 Report

Overview

In Part 2 of the SmartHome Gesture Control Application project, we focused on determining specific gestures from a set of provided gesture videos. The process involved feature extraction, cosine similarity comparison, and outputting the recognized gestures. The following steps describe our methodology:

1. Training Set Preparation

We used the training set from Part 1 to prepare a feature vector set for all videos, ensuring a comprehensive representation of each gesture:

a. **Extract Middle Frames**: For each video in the training set, the middle frame was extracted. This frame was considered representative of the gesture to provide consistent input to our model.

b. **Feature Extraction**: The middle frame image was processed through a pre-trained Convolutional Neural Network (CNN) model to extract a feature vector. This vector served as the unique fingerprint of each gesture.

c. **Consolidate Vectors**: All extracted feature vectors were consolidated, along with their corresponding gestures, to create a comprehensive feature vector set. This set served as a basis for recognizing gestures in the test videos.

2. Gesture Recognition for Test Data

The next step involved determining gestures for the provided test videos by comparing them against the prepared feature vector set using cosine similarity:

a. **Extract Middle Frame**: For each test video, the middle frame was extracted to create a representative image of the gesture in the video.

b. **Extract Feature Vector**: The middle frame image was processed through the provided CNN model to generate a feature vector.

c. **Cosine Similarity Comparison**: Cosine similarity was applied between the test video feature vector and the training vector set. The vector from the training set with the minimum cosine difference was determined to be the closest match.

d. **Output the Result**: The output label of the matching vector was then written to the `results.csv` file, along with the video filename and gesture name.

3. Mutation of Testing Data

A mutation mechanism was implemented to improve the accuracy of recognition in cases where a gesture video did not match the expected label:

- **Data Mutation**: When a test video was not recognized correctly (i.e., the label did not match expectations), the vector data from that video was tagged with the correct label and added back into the feature vector set.

- **Rerun Cosine Similarity**: The cosine similarity calculation was rerun, incorporating the newly labeled vector to improve recognition accuracy over time. This iterative approach ensures that the model becomes more robust as it adapts to unrecognized or misclassified gestures.

Summary

The implemented solution effectively extracts and recognizes gestures from test videos by leveraging a CNN model and cosine similarity calculations. The use of middle frames ensures a standardized approach, while the data mutation mechanism facilitates model improvement, even during testing. The results, including recognized gesture names and their corresponding output labels, are saved in `results.csv` for further analysis.