CSE 535: MOBILE COMPUTING

MidTerm – SmartHome Gesture Control Application(Part 2)

Submitted By

Sandhya Balu

1223737762

CONTENTS

1. OBJECTIVE
2. PREREQUISITES
3. APPROACH TO THE PROBLEM
4. SOLUTION TO THE PROBLEM

Objective

A python application to classify smart home gestures using CNN model. To achieve this we use practice gesture videos generated by SmartHome mobile application in project part 1.

Prerequisites

- TensorFlow
- Python 3.6.9
- OpenCV for Python
- Keras

Approach to the problem

Task 1: Generating the penultimate layer for training and test gesture videos

- Extracting the middle frame of each training data using the get_instance method of HandShapefeature Extractor class which extracts the frames from the gesture video.
- Extract the feature vector of the middle frame for each gesture which is the penultimate layer of the training data set.
- Similarly extract the penultimate layer for test data set.

Task 2 : Recognising gesture of the train dataset

- Applying the cosine similarity between training gesture vector and the test penultimate layer and the vector in the training penultimate layer with minimum cosine difference is to be considered as the gesture.

Solution to the problem

Following steps are used to achieve the solution:

- Both the training and test gesture videos are saved in two different folders and imported into main file.

- Training gesture videos are processed using cv2 library
- Each video is passed to a frameExtractor() method of HandShapeExtractor class to get the middle frame.
- Middle frame is then pre-processed to generate a feature vector using getVectorFromFrame() method.
- Feature vectors of all training videos middle frames are stored into a csv file as trainingpenultimate.csv.
- Similarly feature vectors of all test gesture videos middle frames are stored into a csv file as testpenultimate.csv
- Once a penultimate layer for both training and test data set are extracted, the vector of the test gesture video and the penultimate layer of the training set are compared using cosine similarity using scipy.spatial library.
- The vector in the training set which has smallest cosine difference is considered to be the gesture of the test video.
- All the test dataset gestures are to be recognised and the results are saved in results.csv file.