SmartHome Gesture Control Application
Part 2

Purpose

Students will work to develop a python application classifying Smart Home gestures. Students will gain hands-on experience in training and testing a CNN model for hand gestures. This project, and its successful completion, provides an excellent opportunity to gain further exposure to understand machine learning models.

Objectives

Students will be able to:

- Develop a python application that classifies specific gestures
- Train and test a CNN model

Technology Requirements

- TensorFlow
- Python 3.6.9
- OpenCV for Python
- Keras

Project Description

For Part 2, students will develop a python application for hand gesture recognition. To complete this project, students will use the practice gesture videos generated by them in project Part 1 and test gesture videos provided in the test.zip in the instructions and the source code provided.

Directions

Functionality of the application

Task 1: Generate the penultimate layer for the training videos.

Steps to generate the penultimate layer for the training set:

1. Extract the middle frames of all the training gesture videos.
2. For each gesture video, you will have one frame extract the hand shape feature by calling the get_Intsance() method of the HandShapeFeatureExtractor class. (HandShapeFeatureExtractor class uses CNN model that is trained for alphabet gestures)
3. For each gesture, extract the feature vector.
4. Feature vectors of all the gestures is the penultimate layer of the training set.

Task 2: Generate the penultimate layer for the test videos

Follow the steps for Task 1 to get the penultimate layer of the test dataset.

Task 3: Gesture recognition of the test dataset.

Steps:

1. Apply cosine similarity between the vector of the gesture video and the penultimate layer of the training set. Corresponding gesture of the training set vector with minimum cosine difference is the recognition of the gesture.
2. Save the gesture number to the Results.csv
3. Recognize the gestures for all the test dataset videos and save the results to the results.csv file.

Gesture Name	Output Label
0	0
1	1
2	2
3	3
4	4
5	5
6	6

7	7
8	8
9	9
Decrease Fan Speed	10
FanOn	11
FanOff	12
Increase Fan Speed	13
LightOff	14
LightOn	15
SetThermo	16

Submission Directions for Project Deliverables

You will need to submit the following project deliverables:

- Save the python application as main.py
- Save the training gesture videos under "traindata" folder.
- README file
- Save the results (recognized gestures) to results.csv file.
- A report evaluating your application that discusses the following points:
    - An explanation of how you approached the given problem
    - Solution for the problem

The five deliverables above are to be submitted in a single zip file. Please note that the zip file should only contain these deliverables. Please follow the folder hierarchy listed below for your submission:

Submission Project Folder

|---- traindata

|--- main.py

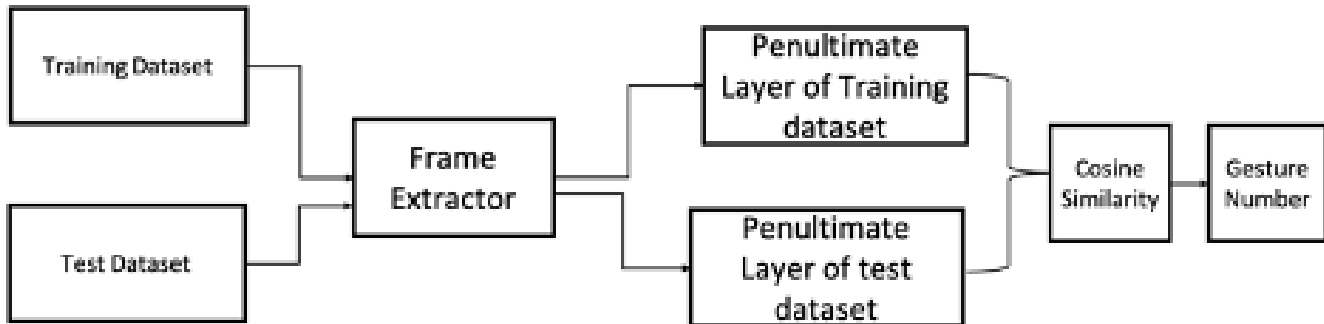|-- cnn_model.h5

|-- handshape_feature_extractor.py

|-- any additional files you may create

|---- README.md

|---- Report.pdf

The name of the zip file that is submitted should be "CSE535Project2_(full name).zip".

Workflow for the project:

Evaluation

Students will be evaluated based on the following criteria:

Compilation errors: 0

Application Compiles: 85

Total score: 85+ Accuracy Score

Accuracy score is computed by comparing Gesture true labels and user output labels.