CSE 6367 - Assignments - Assignment 8

In this assignment, you will implement dynamic space-time warping (DSTW). As in the previous assignment, your method will be trained and tested on the data from this directory. The zipped file is 1.8GB.

There are four videos in our dataset. For every video there is a corresponding text file, that specifies for every gesture in that video the class, start frame, and end frame of the gesture.

Task 1 (30 points): Implement a function `dstw_classifier(filename, start_frame, end_frame)`, that takes as input three arguments: the filename where an AVI video is saved, and the start and end frames of the gesture we want to recognize. The function returns as output a number between 0 and 9, representing the classification of the gesture using dynamic space-time warping. The hand candidates for the gesture we want to classify should be computed on the fly, using the `detect_hands` function available from the course website. Any other data that your function needs should be precomputed, and stored in a file called dstw_data.mat, that your `dstw_classifier` function loads. You should submit the dstw_data.mat file as well, and clearly describe in your README file what data was saved there and how that data was computed.

Task 2 (20 points): Evaluate DSTW as follows:

- Measure classification accuracy when the test data is the 20 gestures from the "short_sleeve" sequences, and the training data for each gesture is the 20 gestures from the two "model" sequences. In this case, find hand locations in the test video using the detect_hands function.

For DSTW, you should evaluate the accuracy that you obtain using different values of K, between 1 and 15.

Task 3 (30 points): Implement the gesture spotting version of DSTW: implement a function dstw_spot(input_video), that takes as input a filename where an AVI video is stored. The function returns a matrix of size Px3, where P is determined by your algorithm. This matrix, at each row, contains three numbers: [start_frame, end_frame, gesture_class]. In other words, for each gesture that occurs in the video, according to your algorithm, you should indicate the start frame, end frame, and gesture class for that gesture.

The candidate hand locations for each frame of the entire video should be computed using the `detect_hands` function available from the course website. Any other data that your function needs should be precomputed, and stored in a file called dstw_spotting_data.mat, that your `dstw_spot` function loads. You should submit the dstw_spotting_data.mat file as well, and clearly describe in your README file what data was saved there and how that data was computed.

Obviously, for the purposes of this task, YOU SHOULD NOT ASSUME that the system knows when each test gesture starts and ends. The system only knows when the training gestures start and end.

Task 4 (20 points): Evaluate the gesture spotting version of DTW and DSTW (for values of K between 1 and 15) as follows:

- Measure accuracy using DTW (not DSTW) when the test data is the 20 gestures from the "model" sequences, and the training data for each gesture is the 10 gestures from the other "model" sequence. In this case, find hand locations using the green_hands function.

- Measure accuracy using DSTW (for values of K between 1 and 15) when the test data is the 20 gestures from the "short_sleeve" sequences, and the training data for each gesture is the 20 gestures from the two "model" sequences. In this case, find hand locations in the test video using the detect_hands function.

In a readme.txt file, you should specify your measure of accuracy. Note that measuring the accuracy of a gesture spotting system is not quite as straightforward as measuring the accuracy of a gesture recognition system that knows when each gesture starts and ends in a test video, and that is why you should clearly describe your choice of how to evaluate accuracy.

How to submit

Submisions are only accepted via Blackboard. The submission should include a zip file containing your Matlab code and data and a README.txt file. The zip file should be named `assignment8.zip`. Your solution should definitely contain:

- A README.txt file, which should contain the name and UTA ID of the student, in addition to any additional comments/instructions useful for running the code and understanding the underlying ideas.
- The code that you wrote for solving each task
- Any auxiliary code that you have used. If you used code posted on the class website, you need to include that code in your submission.
- Your results for tasks 2 and 4.

We try to automate the grading process as much as possible. Not complying precisely with the above instructions causes a significant waste of time during grading, and thus points will be taken off for failure to comply, and/or you may receive a request to resubmit.

Submission checklist

- Was the name of the submitted zipped file assignment8.zip?
- Did you include a README.txt file, as specified?
- Was the submission zipped? We will not accept .rar, .tar, .gz, or any other filetypes, and we will not accept submissions where multiple files are submitted separately.