

Lab on Design Implementation of DeepFace CNN HL

This lab is the last lab. Note the lab report has to be submitted together with the zip project file and video clip(s) of the project. All submission material has to be emailed in by the 17th of May (Sunday) 5:00 pm.

1. For the object tracker computation, you will need to
 - (1) down load the sample code object tracker (see the url link of the class PPT) and place it in a proper working directory and run the program to observer how the tracking function works.
 - (2) the object tracker program does tracks human face, and make sure your program realize that function for a single frame input; and
 - (3) make sure it works for video file input, then down load any arbitrary youtube video with human faces, and make sure the object tracker program works;
 - (4) then switch to live video usb camera input, make sure the tracking functions works for the face input from the usb camera.
2. Modify the object tracker computation, make it the first modified version, say, faceTracker1.py, so
 - (1) for any size rectangle and/or square bounding boxes human faces, perform tracking function and draw the roi bounding box.
 - (2) for each roi having a human face, make sure your program faceTracker1.py save it, either as a single frame image if the input is a single frame, or as a video file if the input is a file.
3. Create a new faceTracker2.py and this program will read the saved image or video clips back and normalize it a 228x228 square image or video, then save the normalized the image/video as a file.
4. Now try the deepface CNN, see the url link from the lecture ppt posted on line (github/hualili),
 - (1) down load the sample code and place it in a proper working directory and run the program to observer how the deep face program works.
 - (2) save the weights of the trained deep face program.
 - (3) modify the deep face program, make a new program named deepFaceTrained1.py, this program does not need to have a training part, it only needs to read the trained weights back to the program and start performing face detection functions;
5. Create a new deepFaceTrained2.py by allowing this program to read your save normalized 228x228 image or video files, and performing facial detection.
6. Now create a new program by integrating deepFaceTrained2.py and objectTracker2.py to realize
 - (1) the input of image or video from a file;
 - (2) the input of live image or video from usb camera;
 - (3) with object tracking function to track human face and to normalize the human face roi to 228x228 image or video files, and perform facial detection.
7. Test your program in 6 for 2 video clips from youtube.

Submit project report together with

- (1) Source code subject to testing and verification;
- (2) submit the photo/video of your implementation (for video not to exceed 10 MB).

8. Rubrics for lab demo:

- (1) satistisfies the requirements stated above;

(2) lab report should cover each of the above program implementation, should cover

Algorithm description

Flow chart(s)

Pseudo code

testing and verification section

source code listing (appendix)

(END)