Due: March 2022 Total points: 20

Face Detection with OpenCV

For the programming part, you are going to implement face detection with OpenCV. You are free to use online resources for the following parts of the exam.
OpenCV is an open-source library that includes several hundreds of computer vision algorithms. Most algorithms discussed in the class can be found here. The goal of this assignment is to help you get familiar with OpenCV.

1 Install OpenCV

2 point

OpenCV has C++, Python and Java interfaces and supports Windows, Linux, Mac OS, iOS and Android. You can check them out here: `https://opencv.org/releases.html`. For the purpose of this class, please use the python interface. (Optional) For easy management of the packages, we recommend that you use Anaconda or other virtual environment, and install the OpenCV package inside the virtual environment.

Anaconda Installation: `https://www.anaconda.com/products/individual`

Anaconda Documentation: `https://conda.io/projects/conda/en/latest/user-guide/tasks/manage-environments.html`

Here are some useful links for the installing OpenCV:
Windows - Python:
`https://www.geeksforgeeks.org/set-opencv-anaconda-environment/`

Mac - Python:
`https://www.pyimagesearch.com/2016/12/19/install-opencv-3-on-macos-with-homebrew-the-easy-way/`

Linux - Python:
`https://docs.opencv.org/master/d2/de6/tutorial_py_setup_in_ubuntu.html`

In your report, write a short summary of the steps you have followed to install OpenCV and any challenges you have faced. After the installation, follow the tutorial below and write a short program to read and perform canny edge detection on an image. Add a screenshot with your displayed image into your report.

Python version: `https://www.geeksforgeeks.org/python-opencv-canny-function/`

2 Face Detection

8 point

Write a program to implement video-based face detection with OpenCV functions. Record your screen showing the execution of your program and the demonstration of face detection in a video of yourself. Submit a 20-30 seconds screen-recorded video. It would be very helpful for your next section if you can achieve real-time performance.

In your report, describe how the detection works, what kind of challenges you faced, and cases where it fails. Also describe any methods you used to accelerate the program.

The following tutorials might be helpful for you:
Video Processing: `https://www.geeksforgeeks.org/python-displaying-real-time-fps-at-which-webcam-video-file-is-processed-using-opencv/`
Face Detection: `https://www.datacamp.com/community/tutorials/face-detection-python-opencv`.

3 Real-time CV applications

5 points

Understand and implement **one** out of the following applications. The end result is to test the algorithm on a **real-time feed** from either webcam or smartphone camera. You are open to use existing code found online (make sure you cite appropriately and when necessary). Note that your algorithm should be able to run real-time. Thus your final video submission should include the following: the code that's running, and the output video.

3.1 Options for CV applications

Option 1: Virtual Background (like the one in zoom)
Similar to the "Virtual Background" in Zoom, demonstrate a live-feed your background (everything in the scene except for you) is replaced by a fixed image. Example tutorial: `https://www.learnopencv.com/applications-of-foreground-background-separation-with-semantic-segmentation/`
Option 2: Object Detection
Take a fixed object of your choice (such as apple) and display a bounding box around the object as you move it around in the live-feed. Example tutorial: `https://pythonprogramming.net/haar-cascade-object-detection-python-opencv-tutorial/`
Option 3: Gaze Tracking

Track your eyes in the live-feed as you move them around.
Example tutorial: `https://medium.com/@stepanfilonov/tracking-your-eyes-with-python-3952e66194a6`

Option 4: Body Pose tracking
Display a stick-figure that is aligned with the your body pose. Example tutorial: `https://www.learnopencv.com/deep-learning-based-human-pose-estimation-using-opencv-cpp-python/`

Option 5: Snapchat face filter (e.g. virtual beard)
Super-impose virtual objects such as beard, hat etc. on your face that rotate, translate and scale as you move your face around. Example tutorial: `https://github.com/charlielito/snapchat-filters-opencv`

The report should address the following points.
1. Summary: In one page, summarise the key aspects of the implementation you used to solve the CV application of your choice. You will be penalized if you miss to comment on aspects that were crucial for the algorithm to function.(1 points)
2. Parameters: What are all the important parameters that are upto the user of your algorithm to decide? For example, threshold value in binarization. Report images or video recordings from your live-feed that demonstrate the effect of tweaking these parameters. Justify these effects.(1 points)
3. Failure cases: Report images or video recordings from your live-feed where the algo- rithm fails to perform the required application. Justify why that's the case.(1 points)
4. Final output: Submit a 20-30 seconds recorded video from web-cam or smartphone camera demonstrating the your chosen application.(2 points)

Submission Instructions

Every student must submit following 3 files:

- An organized report submitted as a PDF document. The report should describe the implementation, issues (problems encountered, surprises), and an analysis of the test results (interpretation of effects of varying parameters, different image results). Intermediate and final results must be provided.
- Videos to demonstrate your face detection result and real-time CV application result.
- A ZIP file containing the necessary codes.

Late Submission Policy

No late days are allowed for this assignment.

Collaboration Policy

I encourage collaboration both inside and outside class. You may talk to other students for general ideas and concepts but the programming must be done independently.
For the written part, there will be no collaboration permitted.

Plagiarism

Plagiarism of any form will not be tolerated. You are expected to credit all sources explicitly. If you have any doubts regarding what is and is not plagiarism, talk to me.