

Second Assignment

Computer Game AI

COMP09041

Issue Date: Tuesday, March 10th, 2020
Due Date: **5pm, Monday, March 30th, 2020**



Face Recognition

The simplified version of the OpenCV “Eigenfaces” face recognition program was introduced in last week’s lab session (and included here as `facerec.cpp`). Using the AT&T Face Database, the program removes a randomly selected image of a face from the 400 in the database; and then attempts to match that image with those remaining of the same class; i.e. of the same person.

Add your own face to the database by adding ten 92×112 8-bit pgm images, named `1.pgm`, `2.pgm` etc., to a new subdirectory of **`att_faces`** named, say, **`s41`**. Then, update the code to operate in a continuous mode, and output a simple notification should *your* face be recognised by the camera. You may optionally display the video from the camera during execution.

After completing your coding, please also answer these two follow-up questions:

- The `FaceRecognizer::predict` method returns a label indicating a match has been found. Is it possible to obtain a measure of the system’s *confidence* in that match?
- Does the program compensate if you are far from the camera? Might a cascading classifier help?

Suggested Approach

After having updated the database, you will want to add a loop to the code of `facerec.cpp`, to check for the appearance of your face. Note that your call to

the `FaceRecognizer::train` method will occur *before* this loop. Note too that you may need to come closer to the camera, or use a *cropping* operation, to obtain a match. Feel free to consult the video input handling code from Session 6, as seen in `4.capture_show_video.cpp`.

Resources

You have at your disposal from Session 7's Moodle site:

- The AT&T Face Database: `att_faces.zip`

As before, ensure that you have this unpacked in the same directory as the `facerec.cpp` source file.

The XnView image manipulation program is available in the computer labs, and may help you to create the required `pgm` files; right-clicking an image and selecting “crop” may be useful.

Please get in touch if you would like to borrow a camera.

Other Languages

Other languages such as C# can be used, but the specification remains the same: you must support the AT&T face database, and add your face. The program must run interactively from a video stream, and use OpenCV.

Submission

Your submission should include your `CMakeLists.txt` file (or other project file), and your source code, along with a `.txt` file containing your answers to the two questions above. You may optionally include the ten images of your face; *or* a screenshot of Windows Explorer such as that provided in `faces.jpg`. You should work individually.

Marking Scheme

The assignment is worth 40% of the marks awarded for the entire COMP09041 module. The following provides a breakdown of the marking scheme for this assignment:

Meeting the program specification	20%
Quality of the code submitted	12%
Answer to Question a.	4%
Answer to Question b.	4%

Plagiarism

Ensure your work is yours alone; or clearly credit the source of your borrowing. You can discuss ideas with your fellow students regarding how to prepare a solution, but the *copying or sharing of code is not permitted*. (Distinctive work is encouraged and ***rewarded***.)