Second Assignment
Computer Game AI
COMP09041

Issue Date: Tuesday, March 19th, 2024
Due Date: **5pm, Friday, April 19th, 2024**

Face Recognition

A simplified version of the OpenCV "Eigenfaces" face recognition program was introduced in this 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 the face of each member of your team to the database. For each team member, add ten 92×112 8-bit pgm images, named 1.pgm, 2.pgm etc., to a set of new subdirectories within **att_faces** named, say, **s41, s42, s43 etc.**. Then, update the code to operate in a continuous mode, and output a simple notification should *one of your* faces be recognised by the camera. You may optionally display the video from the camera during execution.

You should consider the aspect ratio of the trainging data (i.e. the faces in `att_faces`). When you grab a frame from your camera's video, we want to avoid squashing this full image to the required aspect ratio. So, you should highlight/outline a rectangle, with the same aspect ratio as the training data. You should also allow the user to move the rectangle to a better position using the mouse (but don't let them change its size). Use the rectangle to extract a `cv::Mat`, and use that to find a match.

For fun, apply a filter to the video shown within the rectangle (e.g. monochrome, blur, flip, or something surprising); but remember: don't pass the filtered `cv::Mat` to the matcher.

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

a. 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?

b. Does the program compensate if you are far from the camera? Might a cascading classifier help?

Suggested Approach

First, form a team, with 3 or 4 members; then send me an email as soon as the team is established.

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. Feel free to consult the video input handling code from the lab; e.g. `04_capture_show_video.cpp`.

Resources

You have at your disposal the AT&T Face Database within the `att_faces` subdirectory.

The GIMP image manipulation program is available in the computer labs, and can help you to create the pgm files of the correct size. You will likely also use it to "crop" images to 92×112 (or double or triple that size).

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

Submission

Your submission should include your CMakeLists.txt file; your source code; and a simple text file containing your answers to the two questions above. You may optionally also include the images of your team's faces; *or* a screenshot of Windows Explorer such as that provided in faces.jpg.

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:

An operational program (meets the specification above)	8%
Ensure a 92×112 `cv::Mat` is given to `predict`	8%
A rectangle is drawn for the user's face	4%
A filter or effect is applied within the rectangle	4%
Move the rectangle using the mouse click/drag	4%
Quality of the code submitted	4%
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 other teams, regarding how to prepare a solution, but the *copying or sharing of code is not permitted*. (Distinctive work is encouraged and ***rewarded***.)