

Experiment 6: Faces

Adrian F. Clark

This experiment explores face detection using the Viola-Jones algorithm. It first attempts to find common failure modes of the algorithm and then asks you to think about how you might measure whether a different Haar cascade might be more effective at detecting faces.

There is also a Haar cascade to detect eyes but it often produces identifications them in spurious locations. You are asked to make some modifications to the face detection program to ensure that only valid eyes are found.

Contents

1	Introduction	2
2	Viola-Jones	2
3	Finding eyes	2

1 Introduction

To do this experiment, you will need to download the [accompanying zip-file](#).

2 Viola-Jones

The [zip-file](#) contains a program, `face-detect`, which uses the OpenCV implementation of Viola's and Jones's algorithm to locate faces. You can run it on a test image by typing a command such as:

```
./face-detect ese-small.jpg
```

If you run it with no arguments, it will try to use live video, so if you have a laptop with a camera or a webcam on a desktop machine it should use that. If you don't, it will probably crash.

With the above command line, you will see from the displayed image that most of the faces of the group have been found and are outlined in blue...but not all of them. Can you identify any characteristics of the faces that weren't found to identify possible failure modes in the algorithm? Discuss your findings with a demonstrator.

There are some values in the `face_cascade.detectMultiScale` invocation that adjust the way in which it works. Try adjusting them and find out if they make any difference on the effectiveness of finding faces. Can you figure out an experiment that would provide definitive evidence? Discuss your idea with a demonstrator.

As described in Chapter 8 of the lecture notes, the face location algorithm due to Viola and Jones uses a Haar cascade to locate faces. You are provided with one of the standard ones distributed with OpenCV to use here — but if you go looking on the Web, you will find others. Feel free to download others and adapt the program to use them. How would you work out which Haar cascade is the most effective? Discuss your answer with a demonstrator.

3 Finding eyes

The program also uses a Haar cascade that attempts to find eyes, which it outlines in green. However, it often finds spurious eyes: see the results from the images whose name starts in "eye-". Modify the program so that it:

- ignores eyes in the bottom of the face;
- finds eyes in pairs;
- requires them to be at roughly the same line of the image.

How easy were these to implement? I expect some to be easier to do than others. Can you display rejected eyes in some way?