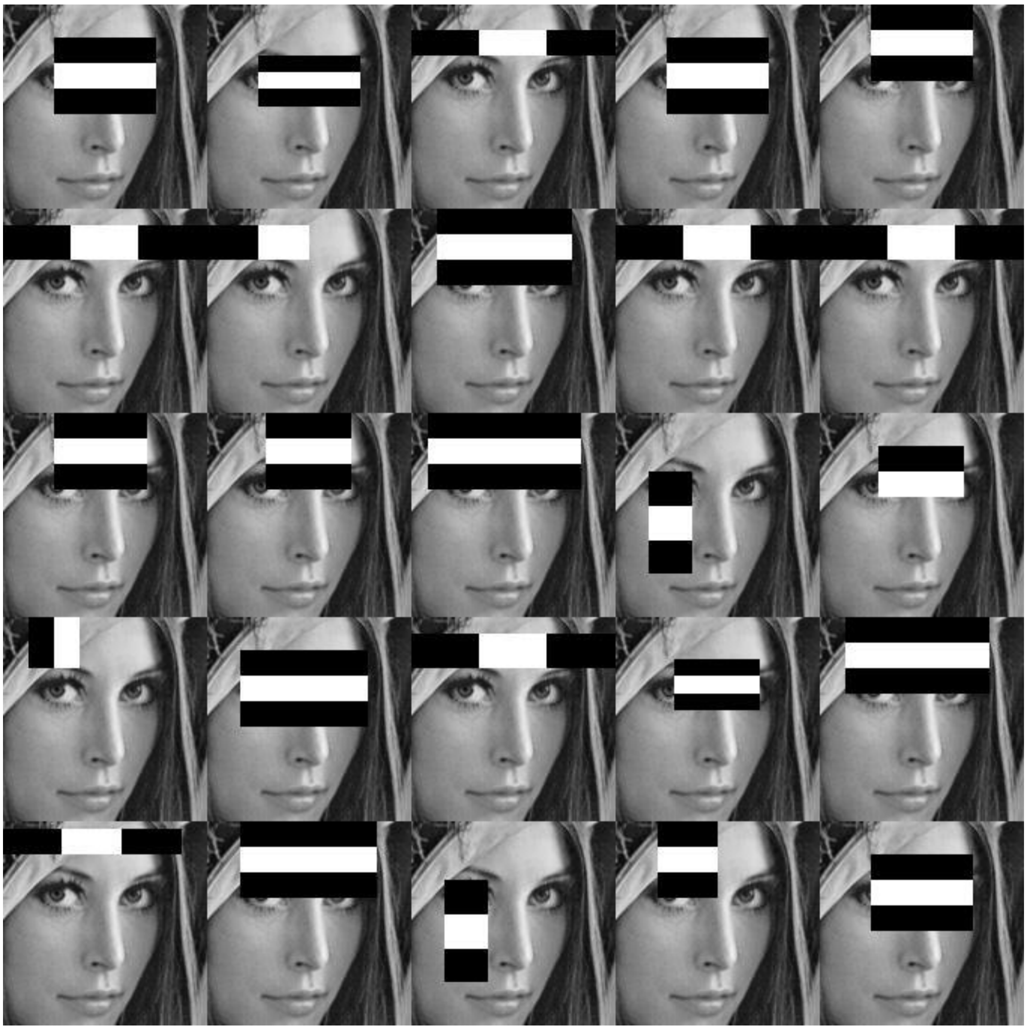
Coursework 2 – Face Detection

Brief

Due date: Friday 19th March, 16:00.
Software tools:
Java– SimpleFaceDetector-1.1-tool.jar and README.md in Teams coursework folder
Python- https://docs.opencv.org/3.4/d2/d99/tutorial_js_face_detection.html
Handin: https://handin.ecs.soton.ac.uk/handin/2021/COMP6223/1/
Required files: report.pdf
Credit: 10% of overall module mark

Overview

Face detection algorithms such as the classic Viola-Jones Haar Cascade algorithm (pdf available in the Teams CW folder) are trained to select weakly-discriminative features that are commonly present (or not present) in images for faces. For example, a common feature in images of faces is that the average intensity of a rectangular region that crosses the eyes is lighter than the region immediately above and below it. The image below shows some common discriminative features:



Examples of common discriminative features from a Haar cascade face detector. Taken from https://www.flickr.com/photos/unavoidablegrain/6884354772.

The aim of this coursework is for you to explore how you might camouflage a face in order for it not to be detected by software, whilst still being recognisable as a face by humans. We have provided a software implementation of a Haar cascade detector that you can use for experiments.

Details

The use of Haar-like features coupled with a cascade of classifiers has proved to be an effective approach to object detection. The method was originally proposed by Paul Viola (Paul now works for Amazon and is Vice President of Science for the team that is working on drone deliveries!) and Michael Jones in their 2001 paper, "Rapid Object Detection using a Boosted Cascade of Simple Features". The technique incorporates a machine learning based approach in which a cascade function is trained from a large number of positive and negative images. The classifier learnt during the training phase is then used to detect objects in other images.

We have written a Java tool that is capable of using the Viola-Jones method to detect faces in still images and live videos from a web-cam. The tool and documentation are on the Teams CW folder. For Python users, a tutorial and an online testing tool for the same implementation are available on the official OpenCV site.

For this coursework, we want you to explore how the face detector can be tricked into not detecting your face through the application of camouflage. You should take photographs of yourself and run them through the face detector to see if your face is detected. You might try investigating some of the following:

- Can you apply face-paint or make-up to subvert the detector?
- Can you wear a disguise (e.g. put on a hat, glasses, ...)?
- Can you create some form of innovative jewellery that stops the face detector finding your face?

For every type of camouflage you investigate, you should test it under a range of different poses (bearing in mind that the detector will only work for upright, near-frontal faces). Ideally, any camouflage that you design should be as non-invasive as possible and also have minimal occlusion - your face should still be detectable or visible to a human, so putting a cardboard box over your head, whilst technically subverting the detector, is not a good solution!.

Report

You need to produce a short report that describes and illustrates (with photographs) your camouflage techniques. You should include short summary of the Haar cascade face detector, describe the process you went into designing the camouflage and relate its functionality to the way the Haar cascade detector works. The report should also demonstrate (through pictures) that your techniques work under a range of near-frontal face poses. The limit is 2 pages with the CVPR_Final Latex format in the teams CW folder (no change in margin and font size). 1 additional page of appendix is allowed only for supplemental images and reference.

What to hand in

You need to submit to ECS Handin the following items:

- The report (as a PDF document)

Marking and feedback

Marks will be awarded for:

- Identifying a range of techniques to subvert the face detector implementation.
- Describing the rationale for these techniques and related them to the way the face detector works.
- Evidence of professionalism in the implementation and reporting.
- Quality and contents of the report.

Standard ECS late submission penalties apply.

Individual feedback will be given covering the above points.

Useful links

- The Original Viola/Jones Conference paper
- Extended journal paper
- Wikipedia description of the algorithm
- OpenCV tutorial
- OpenCV document and implementation
- The actual detector data "haarcascade_frontalface_alt2.xml" file
- The Java source-code for the detector software (Teams folder)
- OpenIMAJ, which provides an alternative Java implementation
- The detector is a "Tree-based 20x20 gentle adaboost frontal face detector", created by Prof. Rainer Lienhart.

Questions

If you have any problems/questions, use the Q&A channel on Teams, or email Xiaohao and Hansung