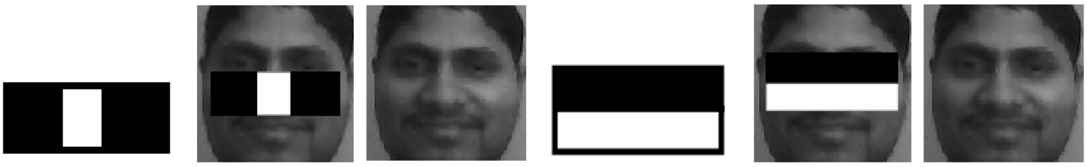
Real-time detection of faces and facial features in a video stream produced by a webcam

S. Blaes, L. Klimmasch, M. Murakami

This project implements a framework for real-time detection of faces and facial features in a video stream produced by a webcam. The video signal is captured and processed using OpenCV, an extensive library for computer vision that is widely used in academic as well as commercial projects. The detection pipeline is based on a cascade classifier with haar-like feature detectors. Models already trained for face and eye detection as well as an performant implementation of the cascade classifier are provided by the OpenCV project. The results of the detection pipeline are further processed and used to alter the output of the video stream as well as control the amplitude and pitch of a tone generated with brian, a spiking neural network simulator. The project is part of a series of projects that were conducted by students of the Frankfurt Institute of Advanced Studies (FIAS) during the FIAS retreat 2017. The projects were planed and implemented over the course of two days.

The task of detecting and/or recognising objects in an image can be quite challenging depending on the complexity and number of different objects under consideration as well as the complexity of the scene. In recent years convolutional neural networks (DCNNs) became state of the art in this two and many other disciplines **(citations)**. These tasks become even harder if they need to be solved in real-time to, for example, detect/recognise objects in a video stream. Such a stream usually consists of 24, 30 or even 60 frames (individual images) per second imposing an additional time constrained on the problem at hand since each frame has to be processed in under 41 to 16 milliseconds. In this case it can be beneficial to use models of lower complexity to reduce the computation time for each frame. Another reason for using a simpler model might be because of educational purposes since hand crafted features are usually easer to understand as the abstract features that are learned in the deeper layers of a DCNN.

Is the number of objects under consideration limited, in the extreme case to only one object type like faces, one approach to solve these tasks within the limited time frame is by using very simple and fast to compute feature detectors that make use of known properties of the input statistics. One family of such filters are called haar filters. Two instances of these filters are shown in the two left most columns of panels 'a' and 'b' in figure 1. These filters can be used, for instance, to detect common pattern in human faces. The particular instance of a haar filter shown in panel 'a' of figure 1 strongly responds to brighter areas that are surrounded on the left and right sides by darker areas as it is common for the area of the nose. The filter shown in panel 'b', on the other hand, responds well to a darker area on the top and a brighter area on the bottom which is a typical pattern for the area of the eyes where the eye browns are usually darker and the eyes itself are brighter.



(a) Haar-like filter that responds to the nose.
(b) Haar-like filter that responds to the eyes.

Figure 1: Taken from ...

An additional gain in processing speed of an individual frame can be achieved by using a cascade of classifiers that rules out, first roughly and later on in a more fine grained manner, areas that are very unlikely to contain the objects one is looking for. In the end, this concentrates most of the computation time per frame on areas with the highest likelihood to contain the relevant objects.

Both ideas are combined in the work of Viola & Jones (2001)[1]. The four most important aspects of their work and the main steps in the detection pipeline are (1) the description of an image in form of an integral image with which features can be computed in constant time. (2) The selection of a set of basis function (features) that respond particularly well to common facial features. (3) Using AdaBoost to learn a subset of pairs of basis function and position out of all possible such pairs that is sufficient to detect a face in the searing window and (4) a classification cascade that concentrates most of the computation time on the most promising sub-windows, that is, sub-windows that potentially contain a relevant object in the input.

References

[1] Viola, P. & Jones, M. (2001). *Rapid object detection using a boosted cascade of simple features.* In Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on (Vol. 1, pp. I-I). IEEE.