

## Aufgaben zu Gesichtserkennung

Lösen Sie die nachfolgenden Aufgaben und bereiten Sie diese bis zum nächsten Lehrveranstaltungstermin vor. Unterstrichene Aufgaben sind nach Möglichkeit während der Lehrveranstaltung zu lösen.

### LB-GE 00.

- a) Schreiben Sie ein Programm, das als einziges Argument den Dateipfad eines Videos erwartet (verwenden Sie den zur Verfügung gestellten Trailer des Filmes *Matrix: Reloaded* zum Testen) und die Frames dieses Videos in einem Fenster abspielt. Verwenden Sie zum Laden der Videoframes die `cv::VideoCapture`-Klasse und zeigen Sie jeden Videoframe nur so lange an, wie es für eine annähernd echtzeitkorrekte Videowiedergabe notwendig ist. Das Drücken einer Taste im Videofenster soll die Wiedergabe abbrechen.

Benötigte Header: `opencv2/highgui.hpp`

- b) Erweitern Sie Ihr Programm aus a) derart, dass ein  $123 \cdot 123$  Pixel großes, rot umrahmtes Rechteck in die Mitte jedes Frames gezeichnet wird, bevor der Frame dargestellt wird. Zudem soll ein identisch formatiertes zweites Rechteck in das erste gezeichnet werden, das im ersten Frame gleich groß wie das erste Rechteck und im letzten Frame minimal klein ist. In den dazwischen liegenden Frames soll das zweite Rechteck gleichmäßig schrumpfen.

Zusätzlich benötigte Header: `opencv2/imgproc.hpp`

### LB-GE 01.

- a) Erweitern Sie Ihr Programm aus LB-GE 00. b) derart, dass die rot umrahmten Rechtecke nur um Gesichter gezeichnet werden. Verwenden Sie zum Erkennen derselben die `cv::CascadeClassifier`-Klasse und die in der Datei `haarcascade_frontalface_alt.xml` gespeicherten, vortrainierten Kaskadenklassifizierer. Geben Sie den Pfad zu dieser Datei aus dem *OpenCV*-Installationsordner (`/usr/local/share/OpenCV/haarcascades`) absolut an und entfernen Sie alle künstlichen Verzögerungen und Wartezeiten.

Zusätzlich benötigte Header: `opencv2/objdetect.hpp`, `opencv2/imgproc.hpp`

- b) Modifizieren Sie Ihr Programm aus a) derart, dass die Objekterkennung deutlich schneller abläuft, indem Sie zumindest einen Parameter der Gesichtserkennungsfunktion entsprechend abändern. Die Erkennungsrate soll dabei nicht deutlich geringer werden.
- c) Modifizieren Sie Ihr Programm aus b) derart, dass die Erkennungsrate in etwa wieder jener aus a) entspricht, ohne dass die Gesichtserkennung dabei langsamer wird als in b). Die False-Positive-Rate darf dabei signifikant höher werden.

**LB-GE 02.**

- a) Erweitern Sie Ihr Programm aus LB-GE 01. c) derart, dass grüne Rechtecke um Gesichter gezeichnet werden, auf deren Nase sich eine Sonnenbrille befindet. Erstellen Sie dazu mittels des Methodenaufrufes `cv::face::createLBPHFaceRecognizer(1, 8, 8, 8, 120)` eine Instanz der Klasse `cv::Ptr<cv::face::LBPHFaceRecognizer>`. Laden Sie mit deren Methode `load` das zur Verfügung gestellte Modell zum Erkennen von Sonnenbrillen in Gesichtern, dessen Dateipfad als zusätzlicher, zweiter Parameter Ihres Programmes angegeben wird. Das zur Verfügung gestellte Modell ist derart trainiert, dass ein Aufruf der `predict`-Methode den Rückgabewert 0 zurückliefert, wenn in dem der Methode übergebenen Gesicht eine Sonnenbrille erkannt wurde. Beachten Sie, dass die `predict`-Methode Grauwertbilder erwartet. Um Farbbilder in Grauwertbilder umzuwandeln, kann die Funktion `cv::cvtColor` verwendet werden.

Zusätzlich benötigte Header: `opencv2/face.hpp`

- b) Modifizieren Sie Ihr Programm aus a) derart, dass grüne Rechtecke nur um Gesichter gezeichnet werden, die dem von *Neo* ähnlich sind. Verwenden Sie dazu das zweite zur Verfügung gestellte Modell, das *Agent Smith* (Rückgabewert 0 beim Aufruf von `cv::face::FaceRecognizer::predict`) und *Neo* (Rückgabewert 1) voneinander zu unterscheiden versucht. Beschriften Sie zudem alle durch das Modell zuordenbaren Gesichter namentlich und in roter Schrift über der linken oberen Kante des gefundenen Gesichtsrechtecks.