

Übung 1: Kennenlernen der OpenCV-Bibliothek

- Die Dokumentation von OpenCV finden Sie unter: <https://docs.opencv.org/master/>
- In der Vorlesung werden wir mit den drei Hauptmodulen core, imgproc und highgui arbeiten.
- Bilder werden als Mat-Objekte abgelegt. Die wichtigsten Informationen zur Arbeit mit Mat-Objekten finden sich im Tutorial: https://docs.opencv.org/master/d6/d6d/tutorial_mat_the_basic_image_container.html
- Datentypenbezeichnung in C++: CV_8U=0 (uchar), CV_8S=1 (schar), CV_16U=2 (ushort), CV_16S=3 (short), CV_32S=4 (int), CV_32F=5 (float), CV_64F=6 (double); mehrkanalige Typen: CV_8UC1 ... CV_64FC4 (für eine Anzahl von 1 bis 4 Kanälen)
- Im Tutorial: https://docs.opencv.org/master/d5/d98/tutorial_mat_operations.html lernt man die Ein- und Ausgabe von Bildern, den Zugriff auf einzelne Pixel und die Anzeige von Bildern.
- OpenCV verwaltet den Speicher in C++ automatisch. Speicher für Ausgabeparameter von Funktionen werden automatisch allokiert, wobei die Größe und der Typ automatisch anhand der Eingabeparameter bestimmt werden. Die Vektoren und Matrizen haben Destruktoren zur Speicherfreigabe, wenn dies notwendig ist.
- Testen Sie nun die Ausführung eines ersten kleinen Beispiels, bei dem ein Bild invertiert wird:

```
#include <opencv2\opencv.hpp>

using namespace cv;

int main( int argc, char** argv )
{
    Mat bild;
    bild = imread("bild.bmp", 0);           // Einlesen des Bildes
    for(int i=0; i < bild.rows; i++)
        for(int j=0; j < bild.cols; j++)
            bild.at<uchar>(i,j) = 255 - bild.at<uchar>(i,j); //Invertieren
    imshow( "Test", bild );                 // Anzeige des Bildes
    waitKey(0);                             // Warten auf einen Tastendruck
    return 0;
}
```

- Schreiben Sie nun selbst ein kleines Programm, welches die folgenden Aufgaben umsetzt:
 - Lesen Sie das Bild „zelle_grau.bmp“ aus dem Download-Bereich ein und zeigen Sie es auf dem Bildschirm an! (imshow)
 - Geben Sie die Bildgröße und die Anzahl der Farbkanäle aus! (size, channel)
 - Bestimmen Sie den Durchschnittsgrauwert (mittleren Grauwert) im Bild. Wie stark weichen die einzelnen Werte von diesem Mittelwert ab? Eine Aussage darüber liefert die Standardabweichung. Geben Sie die beiden Werte aus! (meanStdDev)
 - Bestimmen Sie sich den größten und den kleinsten Grauwert im Bild (minMaxLoc)
 - Ermitteln Sie, wie viele Pixel einen Grauwert besitzen, der größer als der mittlere Grauwert ist. Geben Sie diese Zahl aus und geben Sie ein Bild aus, in dem diese Pixel weiß markiert sind. Der Rest ist schwarz. (compare, countNonZero)
 - Erzeugen Sie aus dem Grauwertbild ein Farbbild! (cvtColor) Erst einmal sollte sich an dem Bild nichts ändern, da in einem konvertierten Grauwertbild alle Kanäle den

gleichen Wert besitzen. Um eine farbliche Markierung hinzubekommen, soll an den Stellen der weißen Pixel vom vorherigen Schritt der Pixel rot gesetzt werden (in BGR die Werte (0,0,255)!