

## Exercise: First steps with OpenCV

### 1. Preparing to work with OpenCV

Open the “Anaconda Prompt”. Activate the environment `env_mss` with the command

```
activate env_mss
```

Then enter:

```
conda install -c conda-forge opencv
```

Example output after this command:

```
(env_mss) C:\Users\Juergen Brauer>conda install -c conda-forge  
opencv  
Fetching package metadata .....  
Solving package specifications: .
```

```
Package plan for installation in environment C:\Users\Juergen  
Brauer\AppData\Local\conda\conda\envs\env_mss:
```

The following NEW packages will be INSTALLED:

ca-certificates:	2018.1.18-0	conda-forge
icu:	58.2-vc14_0	conda-forge [vc14]
jpeg:	9b-vc14_2	conda-forge [vc14]
libpng:	1.6.34-vc14_0	conda-forge [vc14]
libtiff:	4.0.9-vc14_0	conda-forge [vc14]
libwebp:	0.5.2-vc14_7	conda-forge [vc14]
opencv:	3.4.1-py35_200	conda-forge
openssl:	1.0.2n-vc14_0	conda-forge [vc14]
qt:	5.6.2-vc14_1	conda-forge [vc14]
zlib:	1.2.11-vc14_0	conda-forge [vc14]

```
Proceed ([y]/n)? y
```

```
ca-certificate 100% |#####| Time: 0:00:00  
244.60 kB/s  
opencv-3.4.1-p 100% |#####| Time: 0:00:11  
4.90 MB/s
```

Then try the following code to check whether OpenCV has been installed correctly:

```
import sys  
print ("Your Python version is: " + sys.version)
```

```
import cv2
print("Your OpenCV version is: "+ cv2.__version__)
```

My output:

```
Your Python version is: 3.5.4 |Continuum Analytics, Inc.| (default,
Aug 14 2017, 13:41:13) [MSC v.1900 64 bit (AMD64)]
```

```
Your OpenCV version is: 3.4.1
```

Note: In the case, you get an error regarding NumPy, try to update your NumPy library:

```
pip install -U numpy
```

## 2. Reading frames, Processing frames, Writing frames

Write a small Python script that reads images from a video file, e.g., first download

[https://www.youtube.com/watch?v=Ee\\_VFuVC67U](https://www.youtube.com/watch?v=Ee_VFuVC67U)

and use it as an image source.

Read in frame by frame, compute an edge image using the Canny edge detector that comes with the OpenCV, display the original camera image and the edge image and write each computed edge image to a file.

Then use a tool (e.g. ffmpeg) to combine the saved edge images to a video.

Your result could look like this: <https://youtu.be/GLYU8yCWkpU>