

Exercise 0 for MA-INF 2201 Computer Vision WS25/26

14.10.2025

Submission on 19.10.2025

Introduction to OpenCV

Strict rules you have to follow for all your submissions for this course:

- You are required to write code compatible with Linux, python 3.12, opencv 4.11 and numpy 2.3.3.
- For each exercise, we will include all the packages you can use in the template code. You are not allowed to use any other packages.
- You must submit a code that runs and produces reasonable results.
- Do not cheat and copy the solution from anywhere. We need to verify that the code is yours and that you fully understand it.

Any violation of these rules will result in receiving zero points for the corresponding sheet.

This sheet is designed to help you get familiar with *OpenCV*. Hint: Install *python*, *opencv-python*, and *numpy* in your virtual environment (e.g., Anaconda) to avoid library conflicts on your machine.

1. Read and display the image `bonn.jpeg` using `imread` and `imshow`. Print the image dimensions and data type. *(0.5 Points)*
2. Convert the image to HSV color space using `cvtColor` and display all three channels (H, S, V) separately. *(0.5 Points)*
3. Create a brightness adjustment function that adds 50 to all pixel values and clips them to `[0, 255]`. Implement this using nested for-loops. Display the original and brightened images side by side. *(1 Point)*
4. Perform the same brightness adjustment as Exercise 3 using vectorized NumPy operations in one line. Compare the execution time with the loop-based approach. *(1 Points)*
5. Extract a 32×32 patch from the top-left corner of the image. Paste this patch at three different random locations in the image. Display the result. Hint: you can use `random` python module to generate random numbers. *(0.5 Points)*
6. Create a masked version of the image: Convert to grayscale, apply a binary threshold at value 128, and use this as a mask to show only the bright regions of the original color image. *(0.5 Points)*
7. Draw a border around the image (20 pixels wide) using `copyMakeBorder`. Then overlay 5 random circles and 5 random text labels on the image at random positions. *(1 Points)*

Please list the names of your group members in the README. Upload all code and documentation to the corresponding Sciebo folder for the sheet (e.g. *sheet0* for this exercise). Note that the points from this sheet are bonus points. However, it is strongly recommended that you solve the exercises to get experience with OpenCV.