Opencv and numpy examples

Christoph Mayer and Lars Podsiadlowski

February 2023

1 Numpy again

Numpy arrays are used for a lot of different tasks. First let us recap what they are used for and do some exercises:

Please watch the following video to recap numpy arrays:
Ultimate Guide to NumPy Arrays - VERY DETAILED TUTORIAL for beginners!

One of the first things we learned in the video was that they are much faster than normal Python lists.

Exercise 1: Write different functions that fill a lists and a numpy arrays with 100000 random numbers drawn from a uniform distribution in the range from 10 to 30.
There are different ways this can be done. You can work together to come up with different solutions. Now determine the mean of all the elements in the list.
Now add code to determine how long python needs for the different function calls.

Hint: You can determine the current time by doing the following:

```
import timeit
start = timeit.default_timer()
...
stop = timeit.default_timer()
print('Time used to ... : ', stop - start)
```

Compare the run times for the different versions to fill the array and determine the mean value.

2 Image processing with opencv and numpy slicing

Numpy or list slicing refers to extracting parts of a numpy array. This can be columns, rows or rectangles in a 2D array.
A nice example for numpy slicing is image manipulation. In order to be able to manipulate images we have to learn how to import and work with images in Python. One of the most widely used image processing libraries is the opencv library.
I recommend the following series of introductory videos for opencv:

- OpenCV Python Tutorial #1 - Introduction & Images Please implement all examples in this tutorial yourself.
- OpenCV Python Tutorial #2 - Image Fundamentals and Manipulation Please implement all examples in this tutorial yourself. Here you will practice how to extract parts of a numpy array and copy it somewhere else.
- Unfortunately, the you will probably only be able to run the code in this tutorial if you have Python installed on your own computer and you have a camera installed. If you do not have this it might still be fun watching this video.

  OpenCV Python Tutorial #3 - Cameras and VideoCapture

  /item This video also uses images from the camera. But it should be easy to adapt the code in this video to manipulate images that you read from a file.

  OpenCV Python Tutorial #4 - Drawing (Lines, Images, Circles & Text)
- Again this tutorial uses the camera, but you can load any image to run the examples: OpenCV Python Tutorial #5 - Colors and Color Detection
- The last tutorial is comparatively difficult. You should watch the video, but I see that it might be a bit too difficult for you. OpenCV Python Tutorial #6 - Corner Detection

Exercise 2:

1. An image can be turned into a gray scale image with different formula:

$$gray = (red + green + blue)/3 \tag{1}$$

or

$$gray = 0.299 * red + 0.587 * green + 0.114 * blue \tag{2}$$

Implement both formula and convert landscape images with both formula. Which formula gives nicer gray scale images?

2. Identify all pixel that have a gray scale value larger than some threshold value and convert them to white pixel.

3. Set all pixels in an image that are white to the colour red.

4. Create an image mask which has white pixels for gray values > 0.8 and black pixels for all other gray values.

5. (A bit more difficult) Determine the mean gray scale value in an image. Apply a transformation that makes pixel brighter than the mean even brighter and pixels darker than the mean even darker. Transform the pixel value proportional to the distance from the mean. Apply this transformation to the original colour image.