

Exercises #2

Purpose

This unit covers dealing with color spaces and file operations.

Use the `'help()'` function in python to get information about functions etc.

For example `'help(cv2.imread)'`

Color Spaces

Color spaces are used to define how we represent images. Mainly RGB, HSV, and Grayscale. Images are represented as 2D arrays of pixels. Each pixel has color values in a particular color space. The most commonly used color space is RGB.

The colors based RGB are (255,0,0) for red, (0,255,0) for green and (0,0,255) for blue. For 8 bit images the range is (0 to 255) and for 16 bit image the range is (0 to 65535).

By default, OpenCV opens images in the BGR color space. In order to convert between color spaces, we can use the `'cv2.cvtColor(image, cv2.COLOR_BGR2RGB)'` function. Given a BGR image, this function returns an image in the RGB color space.

Exercise 201

In this exercise, you are expected to convert the color space of the input image to HSV and display each channel.

Use the `'cv2.cvtColor()'` function to convert color spaces and the `'cv2.split()'` function to split the image into its channels. Create an array of zeros and merge it with the channel. Finally, use Matplotlib to display the images.

Exercise 202

Following the example given in **fileOperations.py**, you have to create a script that saves some metadata about the images in the inputs folder. The metadata has to be saved to a JSON file named **metadata.json**.

Additionally, if the **metadata.json** exists, ask the user if they'd like to replace the old file or create a new one. The naming scheme is up to you.

Exercise 203

In this exercise, you have to complete the script so that the video output should be displayed as a 3x3 grid. You can use the functions provided in the **'mergeImage.py'** file. Since you'll be stacking the same frame onto each other, you don't have to resize the frames. The best practice would be to create a row of images and then stack the row vertically to get the final image.

