DLCV – Exercises – Color spaces, Binarization

Exercise 1 (pixel access, color spaces)

Develop a Python script that reads an image and calculates its "negative" in the part of the image corresponding to a centered square (or rectangle). The dimensions of the square (or rectangle) correspond to half the resolutions of the original image. The process is illustrated in Figure 1.

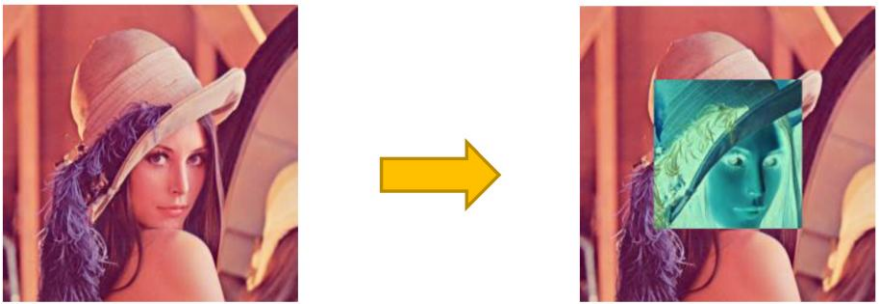

Figure 1 – Applying the "negative" to part of the Lenna image.

The "negative" is obtained by subtracting the color components to the value 255. For instance, the negative of a pixel with BGR components (100, 50, 200) would be (155, 205, 55). The image "lenna.png" is available on Moodle, in the zip attached to the exercises.

Exercise 2 (pixel access, color spaces, conversion)

The aim of this exercise is to build an image that shows color evolution as the Hue component varies, while keeping constant Saturation and Value components (at 255).

The result should be as shown in Figure 2, where the leftmost column corresponds to the Hue value equal to 0 and the rightmost column corresponds to the Hue value equal to 180 (note that 180 is the maximum Hue value in OpenCV and it corresponds to 360 degrees).

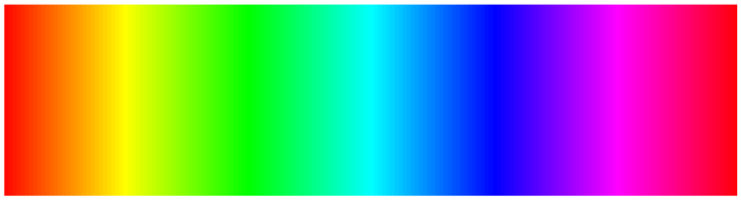

Figure 2 – Varying Hue while keeping Saturation and Value at 255.

Exercise 3 (color spaces, binarization, masks)

In this exercise, the objective is to use masks to build a new image that shows objects of a given color. The input image contains objects of multiple colors. Only the objects of the chosen color should show up in the output image, as in the example shown in Figure 3.

Figure 3 – Selecting the blue Lego pieces.

Carry out the same process for Lego pieces with another color of your choice (green, red, yellow etc.).

The image “legos.jpg” is attached to these exercises on Moodle. If you use binarization in the HSV space to obtain the mask, it may be useful to check the color diagram depicted in Figure 4 to easily discover the values of H and S that correspond to a given color.

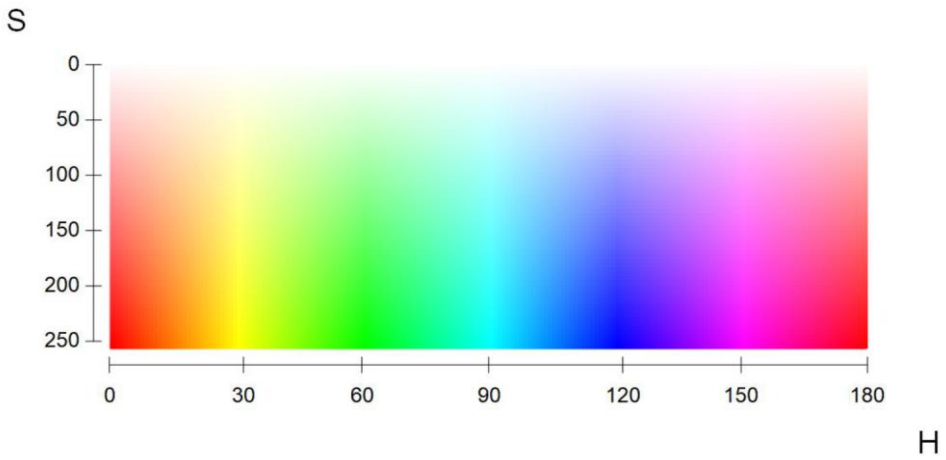

Figure 4 – Color visualization in the HSV space (keeping V=255).

Note: H values are already adapted to the range of values used in OpenCV (from 0 to 180), and not to the true color space range (from 0° to 360°).