TNM087 - Image Processing and Analysis

Task 2 – Simple Replacement of Pixel Values

Rows-Columns-Rectangular operations are especially easy in Matlab. Here you replace a collection of rows columns and grid points with special color values

Background:

Accessing rectangular parts of images can be done with one command line

Task:

Read in the image given by the pathname. Open a new figure and assign it the output filehandle figh.

Change the original image as follows:

a) Replace all pixels in the rows with indices given by rows by red pixels with maximum pure red color
b) Replace all pixels in the columns with indices given by cols by green pixels with maximum pure green color
c) Replace all pixels in the rectangle specified in box by black pixels
d) The intersection of the rows and cols define grid points replace the pixels at these positions with white pixels

Syntax:

```
function figh = SimpleReplace(filename, rows, cols, box )
```

Hints:

For loops are not necessary and not accepted as solutions

Note that if you want to replace the colors of pixels in a region you have to change the values in the red, green and blue channels. In general you may need more than the only line of code in the template.

You can assume that the specifications of the regions are correct. You may add additional tests in which you check some conditions. Examples are (1) positive integer indices (2) index range within the size of the image. If you implement a test decide what the consequences are: (A) return immediately with an error message, (B) repair violations automatically and continue, (C) interactive repair before continuing with the computations

Details:

- `%% create figure and read image`

  Generate a new figure and read in the image (filename is the path to the image file

- `%% Copy image and edit`

  Often it is a good idea to work internally with a copy of your variable even if it is not really necessary here.
  Using vectors you can select rectangular regions in images with one command.
  You can assume that the image is of type uint8 which is important when you generate white points.
  Extra: If you want you can comment on the changes needed when the image is of another type (say 12-bit pixel values in an uint16 array)