Image Processing lab session: steganography and image indexing

N.B. for this labsession, you have to deliver a report with your main results and the code you have developed in each part. Please put all the requested files in a directory named with the last names of the pairs and download the .zip directory on chamilo, Travaux directory one week after the session at the latest.

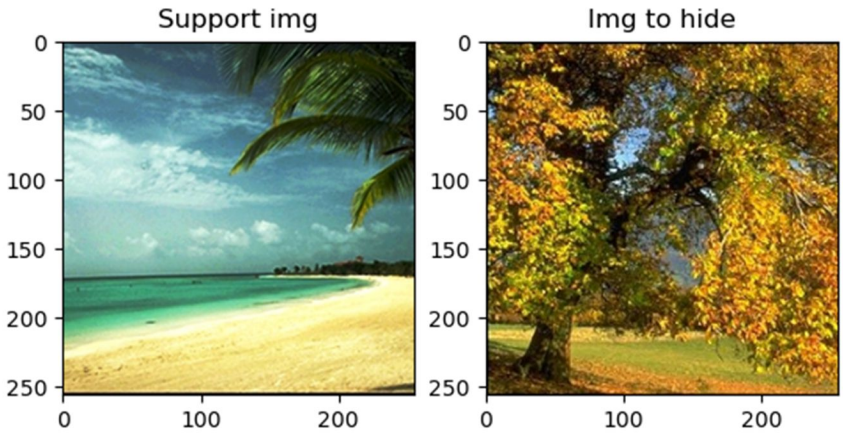
1. Hiding an image into another

The main goal in this part is to hide an image into another. Let's consider a first image called *support image* and an image to be hidden called *img to hide*. Knowing that each pixel of an image is 8-bits coded, the idea is to encode the *img to hide* by replacing the less significant bits of the *support image* by the 4 most significant bits of the *img to hide*.

In this exercise, we suppose that both considered images have the same size.

1.1 Image encoding

1.1.1 Download and display the 8-bits coded ***support_img.png*** and the 8 bits coded ***img_to_hide.png***.



1.1.2 Implement an algorithm that puts to zero the 4 less significant bits of the *support image*. Display the obtained image and compare it with the original *support image*. Comment the result.

1.1.3 Shift to the right the 4 most significant bits of the *img to hide*. Display the obtained image and the histogram of each color channel. Comment.

1.1.4 Build a new image with the 4 most significant bits of the *support image* and the 4 less significant bits of the img to hide. Display the obtained image. Save it in **png format**. In order to check if your code is all right, the ***mixed_img_result.png*** image is provided. Compute the difference between the mixed image you have generated and the provided one.

1.1.5 With your own support image and hidden image, generate a mixed image and save it as ***last_name_mixed_img.png***.

1.2 Image decoding

1.2.1 Implement a code in order to decode the hidden image. Explain how to proceed to retrieve the hidden image. Apply the decoding algorithm first on the mixed image you have created in the previous section in order to check if your code is all right. Display on the same figure the *img to hide* before and after coding/decoding. Compare both images. Compute the max value on each channel of the decoded image. Why is it not possible to have value greater than 240?

1.2.2 Apply image decoding on the following provided images: ***mixed_img1.png*** and ***mixed_img2.png***. Display the decoded hidden image in each case.

1.3 To be delivered

- ✓ Write a report answering the different questions and name it *last_names_hiden_img_report.pdf*.
- ✓ Generate two code files: the first named *last_names_encoder.py* for the mixed_img generation and the second named *last_names.decoder.py* for the hidden image extraction.
- ✓ Put those 3 files and the mixed image generated in section 1.1.4 in the directory entitled with your last names.