Image Processing and Computer Vision
Bachelor Degree in Physics
Remedial exam, September 7th, 2021

---

a) Write a Python script named `yourname_yoursurname1_yoursurnam2_prob1.py` containing the code developed for the exercise. **Do not use symbols** such as spaces, accents, ñ, ç, et cetera. Upload the script to campusvirtual2 (do not upload any other file).
b) Do not change the name of the images provided
c) Note that **the code should run without errors**. Otherwise, it will not be considered.
d) In case of any ambiguity (description of the tasks to develop, possible alternatives to implement et cetera), decide what to do and justify your decision as a comment in the source.
e) Time: 9:00 – 12:00 (+5' extra time for uploading). **Campus Virtual will prevent you to upload your code after the time limit.**

Problem #1

The census transform transform the value of each 8-bit pixel into another one that produces an image that is invariant to changes in the illumination. Read the description of the Census Transform (CT) in https://en.wikipedia.org/wiki/Census_transform and write the code that implements this transformation.

Use the three channels of the astronaut image `skimage.data.astronaut()`and calculate the Census Transform for the three channels. Show the results in a window with 2x3 subplots; display the three channels of the astronaut image and the three Census transformed distributions (CTR, CTG and CTB).

Calculate the Hamming distance between CTR and CTG, CTR and CTB and CTG and CTB. Print the results on the console. Note: The Hamming distance between two gray-level images is defined as the number of pixels that contains different values, e.g.:

Im1= | 128 | 250 |
| 100 | 160 |

Im2= | 128 | 250 |
| 100 | 159 |

$D_H = 1$

Problem #2

Inverse halftoning is the technique for obtaining gray level images from a binary one. The technique we are describing here is particularly suitable for error diffusion (dithering) binary images. According to Wikipedia, [https://en.wikipedia.org/wiki/Halftone#Inverse_halftoning]: *The most straightforward way to remove the halftone patterns is the application of a low-pass filter either in spatial or frequency domain. A simple example is a Gaussian filter. It discards the high-frequency information which blurs the image and simultaneously reduces the halftone pattern.*

Use images 'grayscale cat' [https://en.wikipedia.org/wiki/Halftone#/media/File:Grayscale_Cat.jpg] and 'dithered cat' [https://en.wikipedia.org/wiki/Halftone#/media/File:Dithered_Cat.jpg]. Download these files and do not change the filenames. Do not upload these files to CV.

1. Figure 1 (with 2x2 subplots): Display images 'grayscale cat' and 'dithered cat' and their corresponding histograms.
2. Figure 2, with 1x3 subplots: convert the dithered binary image of the cat into a grayscale one.
   (i) Design and display a Gaussian filter in Fourier space (select an appropriate value of sigma). Display the filter used (131). (ii) Using the Gaussian Filter calculated in the previous step, filter the dithered image, and display the result (132). (iii) In the same plot (133), display the histograms of both the filtered image and the original gray-level image.