Image Processing and Computer Vision
Mid-term mock exam problems.

1. If a scene is illuminated in extreme low light conditions, images are recorded according to the photon-counting (PC) model. In these conditions, it is assumed that image i is statistically modeled by the Poisson distribution. According to this law, the probability that no photon reaches pixel (k,l) is

$$P(0;k,l) = \exp\left(-i_n(k,l)\right).$$

where i_n is the scene normalized according to the following formula

$$i_n(k,l) = N_p \frac{i(k,l)}{\sum_{k,l} i(k,l)}.$$

Note that $\sum_{k,l} i(k,l)$ is the sum of the values of the pixels of the image and N_P is the mean number of photons per pixel. The photon-counting binary image i^{pc} recorded by a PC camera is obtained according to

$$i^{\mathrm{pc}}(k,l) = \begin{cases} 0 & \text{if } \mathrm{rand}(k,l) \le \mathrm{P}(0;k,l) \\ 1 & \text{otherwise} \end{cases};$$

rand(k,l) is a uniformly distributed random number within the range [0,1]. To recover a gray level image, multiples frames must be recorded and added. The following example might help you to understand how this process work: https://en.wikipedia.org/wiki/Shot_noise#/media/File:Photon-noise.jpg

Simulate the image recording procedure in low light conditions. The framerate of the camera used is 30 frames/s and $N_p = 10^3$. Using the luminance L of `skimage.data.chelsea`, calculate:

The recovered images after 10” (300 PC images), 100” (3000 PC images) and 1000” (30000 PC images). For the three cases considered, show the resulting image and the histogram; calculate the structural similarity index (SSIM) between the original Chelsea image and the resulting one. Display the SSIM value in the title of the corresponding subplot (use three decimal places).

Display the results in a single figure with 4x2 subplots arranged as follows:

Original Chelsea image	Original image histogram
Recovered PC image (300 frames)	Recovered PC image histogram (300 frames)
Recovered PC images (3000 frames)	Recovered PC image histogram (3000 frames)
Recovered PC images (30000 frames)	Recovered PC image histogram (30000 frames)

2. Some global threshold binarization methods are based on the analysis of the histogram. Consider the following algorithm:
   a) Calculate the histogram of the image: h(g)
   b) Split the histograms in two parts, from 0 to T and from T to 255, where T is a certain gray level, i.e. $h(g\in[0,255]) = h(g\in[0,T]) + h(g\in[T, 255])$
   c) The binarization threshold T_B is determined according to the following criteria: The sum of the histograms values in $h(g\in[0, T_B])$ is equal to the sum of the histogram values in $h(g\in[T_B, 255])$.

Use the green channel of the astronaut image (`skimage.data.astronaut`)

- Implement de previous algorithm. Determine T_B. Display T_B on the console.
- Binarize the image using the Otsu method. https://scikit-image.org/docs/dev/api/skimage.filters.html#skimage.filters.threshold_otsu
- Binarize the image using an adaptive threshold. Instead of using the median value, use the mean of a 25x25 pixels neighborhood.
- Calculate the Structural Similarity index between the original green channel image and the three binarized images.
- Display the three binarized images in single figure.

imM = ndimage.generic_filter(im, np.mean, footprint=np.ones([25,25]))

3. A binary plaintext B can be split into two random 2D arrays using the XOR logic function. First, a random binary distribution A1 should be produced and then, array A2 that fulfills A1^A2 = B is determined. Distributions A1 and A2 can be given to two different recipients: the information B is only accessible when both arrays are used together.
   The XOR logical function is described by the following table:

A1	A2	A1 ^ A2
0	0	0
0	1	1
1	0	1
1	1	0

   1. Binarize an image using the error-diffusion algorithm. Split the resulting image using the procedure explained above.
   2. Implement this method using gray-level images: first produce 8 bit-planes. Then, use the XOR strategy explained above for each bit plane; and finally, join the two bit-plane sets into two random gray-level images.
   3. Generalize this method for use with 24-bit (color) images.
   4. In the three cases considered (binary, gray-level, and color), recombine the two XOR-related distributions and recover the original image.

Use the image `skimage.data.astronaut()`. Produce three figures with 1x4 subplots containing the original, the two XOR related arrays and the recovered image.