**FollowUp 0**

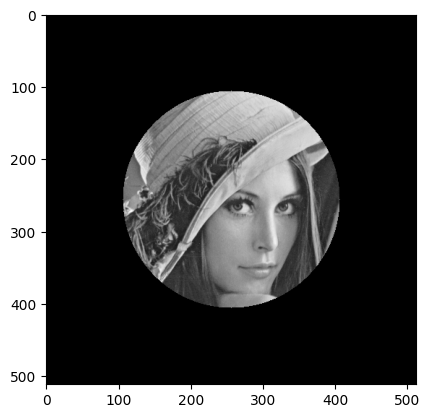
1. Run the 01\_image\_processing\_PIL\_tutorial.ipynb
2. Masks are geometric filters on an image. For instance, if we want to extract a region of an image, we may do it by multiplying the matrix of the original image by a matrix of equal size containing 1′s in the region we want to keep and 0′s otherwise.

In this exercise we extract a circular region of the image [***lena\_gray\_512.tif***](http://www.unioviedo.es/compnum/labs/new/files/lena_gray_512.tif) of radius 150. Follow the next instructions and report every step:

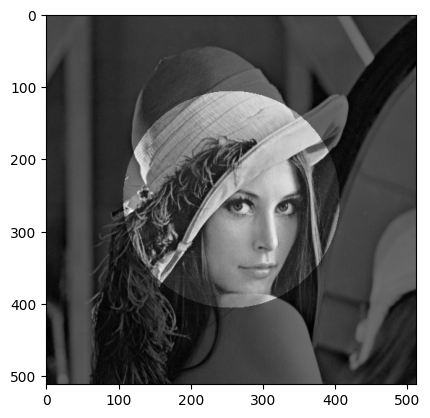
* Read the image and convert it to double.
* Create a matrix of the same dimensions filled with zeros.
* Modify the above matrix to contain 1′s in a circle of radious 150, i.e. if (j−cx)2+(i−cy)2<150exp2, where (cx,cy) is the center of the image.
* Multiply the image by the mask (they are matrices!)
* Show the results.

**Solution**

Code is in the notebook.



When multiplying by zero, you set to black the pixels out of the circle. Modify the program to make visible those pixels with half the intensity.

****

**Hint**

**a.shape[0]** is the number of rows of **a** and **a.shape[1]** the number of columns.

1. Briefly compare PIL and CV2 libraries, similarities, strengths and weaknesses.

Both PIL and OpenCV are widely used for image processing, but they each serve different needs. They share some basic functionality, such as opening, manipulating, and saving images, and both can work with NumPy arrays and integrate with matplotlib for visualization. Where they differ is in scope and complexity. PIL stands out for its simplicity: commands like Image.open(), .convert(), and .save() are very intuitive, which makes it great for beginners, quick experiments, and straightforward tasks like format conversion or simple filtering. On the other hand, OpenCV is much more powerful when it comes to advanced computer vision. It offers highly optimized functions, such as cv2.circle() for drawing shapes, and includes a wide range of tools for tasks like feature detection, object tracking, and more complex transformations. The main drawback of PIL is that it doesn’t go very far beyond basic operations and can be slower with large datasets. OpenCV, while faster and more capable, has a steeper learning curve and can feel like overkill for simple jobs. In practice, PIL is often better suited for educational purposes and lightweight projects, while OpenCV becomes the go-to choice when working with performance-heavy pipelines or deep learning preprocessing on large-scale data.