Practical works – n^{o}3

Convolution

• **Exercise 1** – *1D Convolution*

1.1 Defining a signal x and an impulse response h as parameters, write a program `convFn` to compute the convolution product $y = h * x$. Do not use the python or matlab function and just consider zero padding to solve the border problem.

1.2 Test your function, while using the following signal ($x[n] = \{\underline{1}, 2, 3, 4\}$) and following filters, ($\delta[k-5]$, $H[k-5]$, e^{k-5}, and $\{-1, 1\}$), respectively.

1.3 Consider the previous input signal, wrire it's extended version while considering *symmetry*, *periodic*, *constant values* padding and $h = H[k-5]$.

• **Exercise 2** – *2D Convolution*

2.1 Define a 2D convolution function (Only consider zero-padding the image).

2.2 Using your function, smooth the " lena" image with the following Gaussian kernel.

$$K = \frac{1.}{256}\begin{bmatrix} 1 & 4 & 6 & 4 & 1 \\ 4 & 16 & 24 & 16 & 4 \\ 6 & 24 & 36 & 24 & 6 \\ 4 & 16 & 24 & 16 & 4 \\ 1 & 4 & 6 & 4 & 1 \end{bmatrix}$$

2.3 Apply sobel filtering using convolution.

• **Exercise 3** – *Character recognition using cross-correlation*

3.1 Load "a.png" and "text.png" images, binarized them using otsu thresholding, `threshold-otsu` in python and `graythresh` in matlab.

3.2 Use `signal.correlate2d` and `xcorr` in python and matlab, respectively to correlate your binarized images.

3.3 Find the first 10 characters recognized as **a**.