



César Soltero Pérez

18310460

7E1

Sistemas de visión artificial y procesamiento de imágenes

**Práctica 12**

## Teorema de Convulción

**Objetivo:** Demostrar paso a paso del teorema de la convolución.

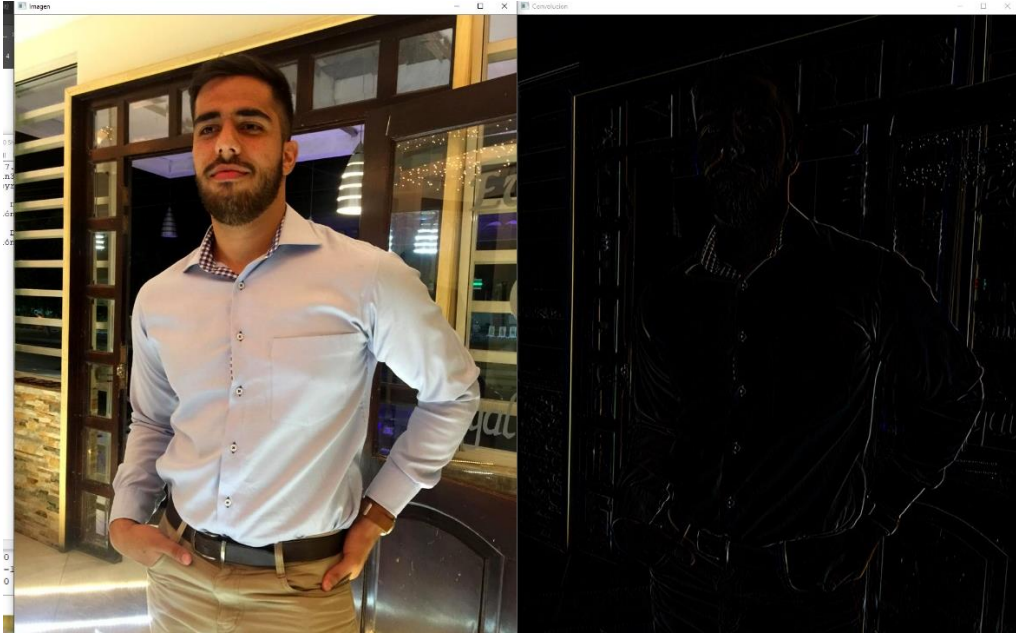
**Desarrollo:** Se aprecia que para una máscara de convulción de 3x3, el valor del píxel  $I'(x,y)$  tras el filtrado depende únicamente del valor del píxel  $I(x,y)$  y de sus ocho vecinos antes del filtrado. Para mantener el resultado de la operación dentro de un rango presentable se puede añadir a la expresión anterior un factor de división y un factor de suma.

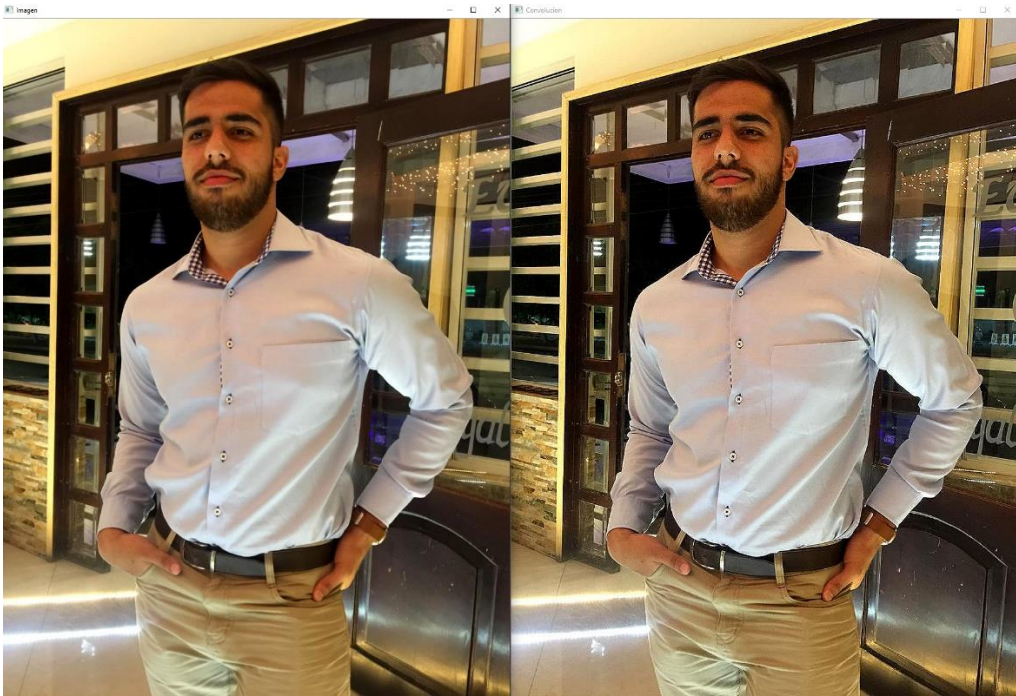
El filtrado paso bajo espacial se basa en el promediado de los píxeles adyacentes al píxel que se evalúa.

```
a = [ [ 1, 1, 1 ],  
      [ 1, 1, 1 ],  
      [ 1, 1, 1 ] ]  
kernel = (1/9)*np.asarray(a)
```



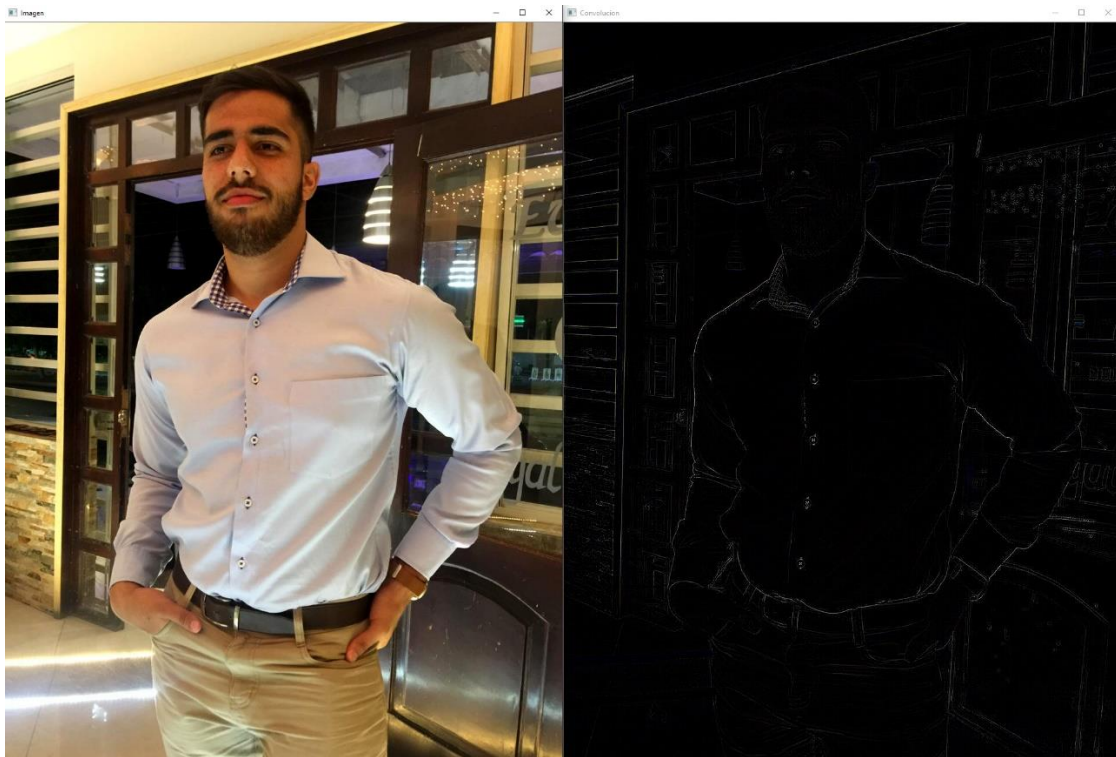
```
a = [ [ 1, 0, -1 ],  
      [ 2, 0, -2 ],  
      [ 1, 0, -1 ] ]  
kernel = (1/4)*np.asarray(a)
```



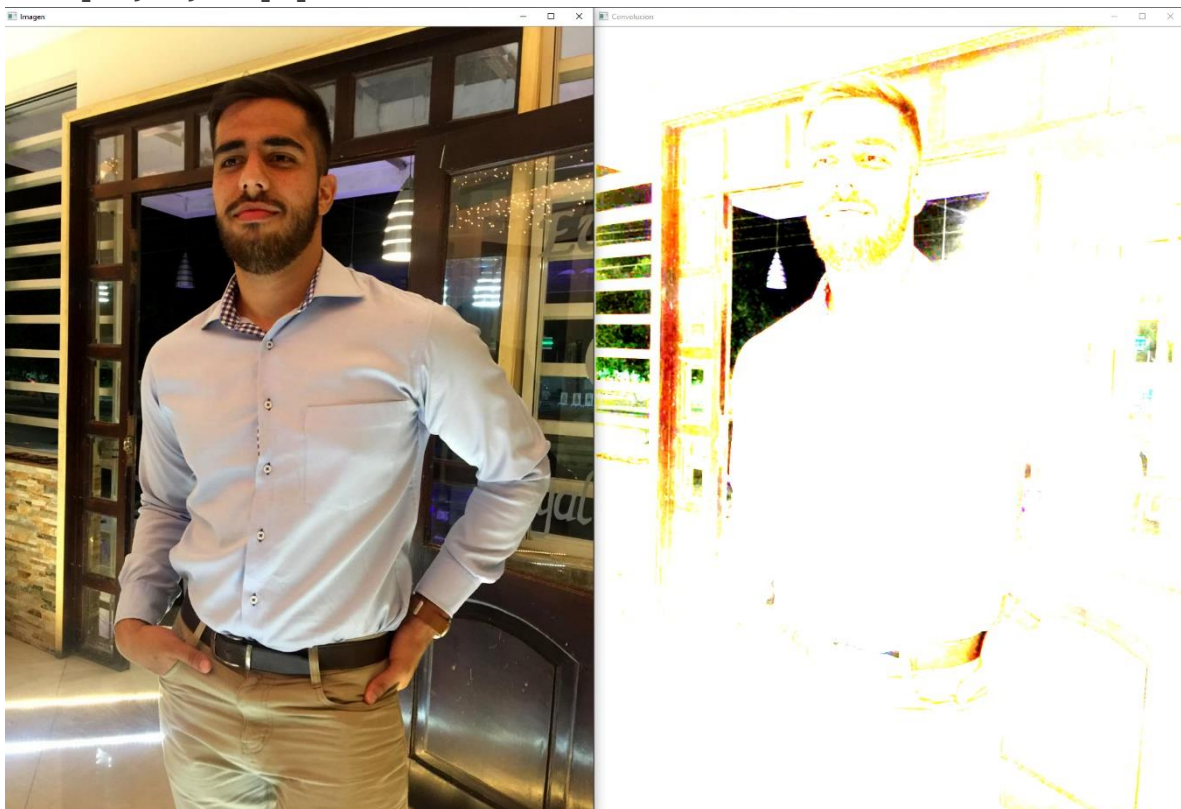
$$a = \begin{bmatrix} [0, -1, 0], \\ [-1, 5, -1], \\ [0, -1, 0] \end{bmatrix}$$


$$a = \begin{bmatrix} [0, 1, 0], \\ [1, -4, 1], \\ [0, 1, 0] \end{bmatrix}$$





$a = \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$



### Código:

```
import cv2
import numpy as np
import imutils

img = cv2.imread('editada.jpg')

a = [ [ 1, 2, 1 ],
      [ 2, 4, 2 ],
      [ 1, 2, 1 ] ]

kernel = (1)*np.asarray(a)

dst = cv2.filter2D(img, -1, kernel)
cv2.imshow('Convolucion', dst)
cv2.imshow("Imagen", img)
cv2.waitKey(0)
cv2.destroyAllWindows()
```

### Repositorio:

<https://github.com/Cesarsp41/Practica-12>