

Visión Artificial

“Practica 8”



Nombre:	Raul Rodriguez Valencia	Registro:	19110337
Fecha:	15/Mayo/2022	Materia:	VA
Grupo:	7E1	Semestre:	5

"Detección de Bordes – Laplaciano, Sobelx, Sobely, Canny.

Objetivo: Dejar en la imagen solamente los bordes que deseamos y saber cuál es el mejor método."

- Método Laplaciano

```
import numpy as np
```

```
import cv2
```

```
image = cv2.imread("FotoBella.jpg")
```

```
image = cv2.cvtColor(image,cv2.COLOR_BGR2GRAY)#Convertir imagen a imagen en escala de grises
```

```
cv2.imshow("Original",image)
```

```
cv2.waitKey()
```

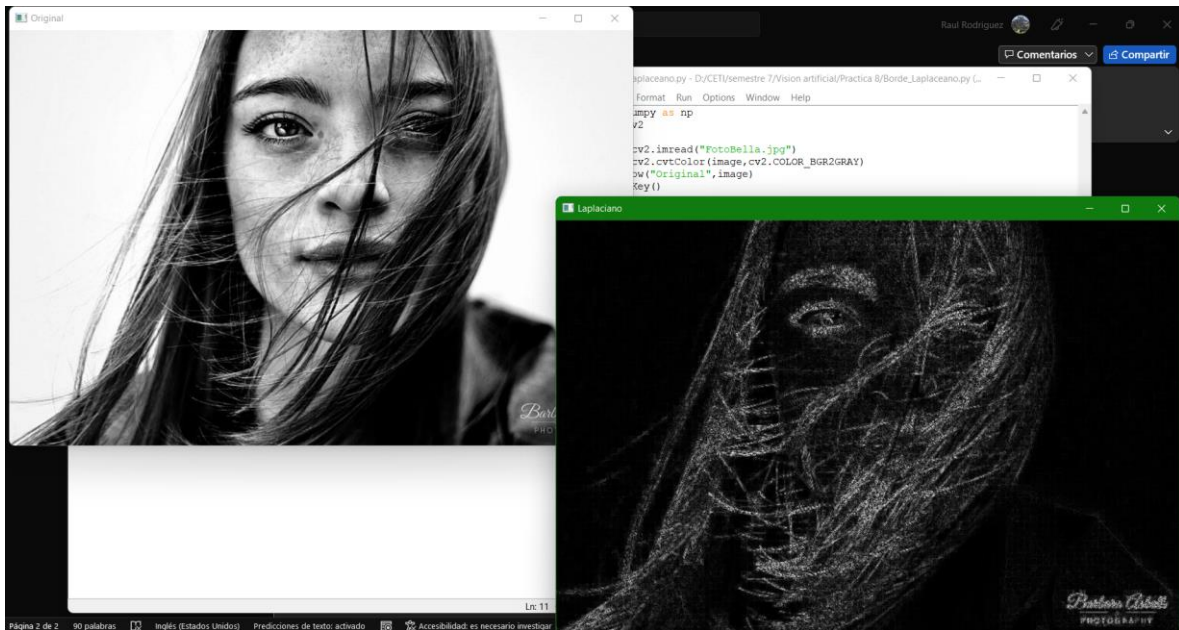
```
# Detección de borde de Laplace
```

```
lap = cv2.Laplacian(image,cv2.CV_64F)# Detección de borde de Laplace
```

```
lap = np.uint8(np.absolute(lap))## Ir al valor absoluto de vuelta
```

```
cv2.imshow("Laplaciano",lap)
```

```
cv2.waitKey()
```



- Metodo Sobel en X y Y

```
port numpy as np
```

```
import cv2
```

```
image = cv2.imread("FotoBella.jpg")
```

```
image = cv2.cvtColor(image,cv2.COLOR_BGR2GRAY)
```

```
cv2.imshow("Original",image)
```

```
cv2.waitKey()
```

```
sobelX = cv2.Sobel(image,cv2.CV_64F,1,0)#x gradiente de dirección
```

```
sobelY = cv2.Sobel(image,cv2.CV_64F,0,1)#y gradiente de dirección
```

```
sobelX = np.uint8(np.absolute(sobelX))#x gradiente de dirección valor absoluto
```

```
sobelY = np.uint8(np.absolute(sobelY))#y valor absoluto del gradiente de dirección
```

```
sobelCombined = cv2.bitwise_or(sobelX,sobelY)#
```

```
cv2.imshow("Sobel X", sobelX)
```

```
cv2.waitKey()
```

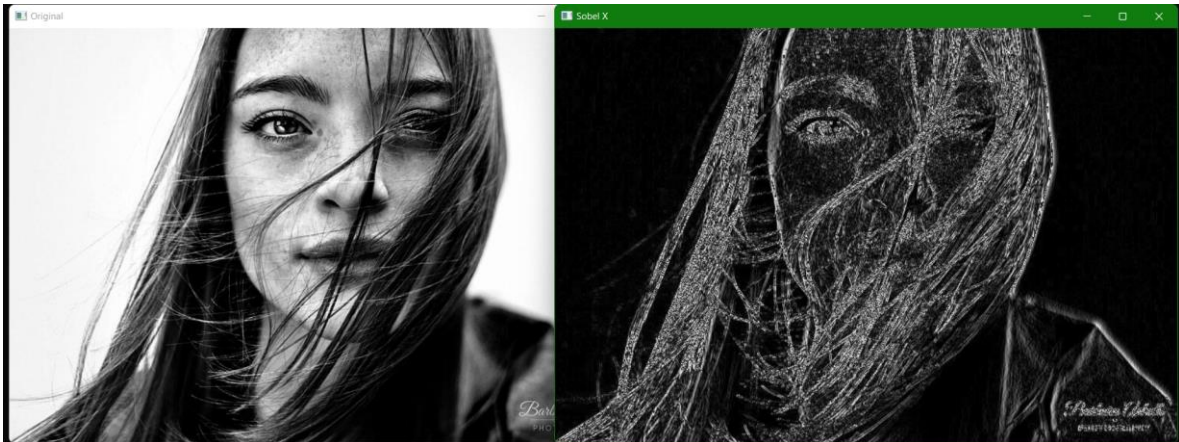
```
cv2.imshow("Sobel Y", sobelY)
```

```
cv2.waitKey()
```

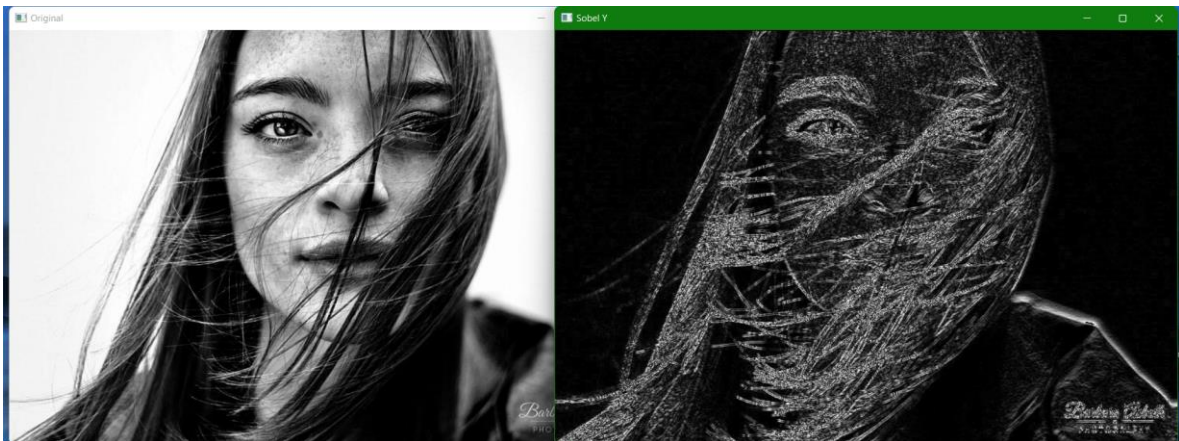
```
cv2.imshow("Sobel Combinada (XY)", sobelCombined)
```

```
cv2.waitKey()
```

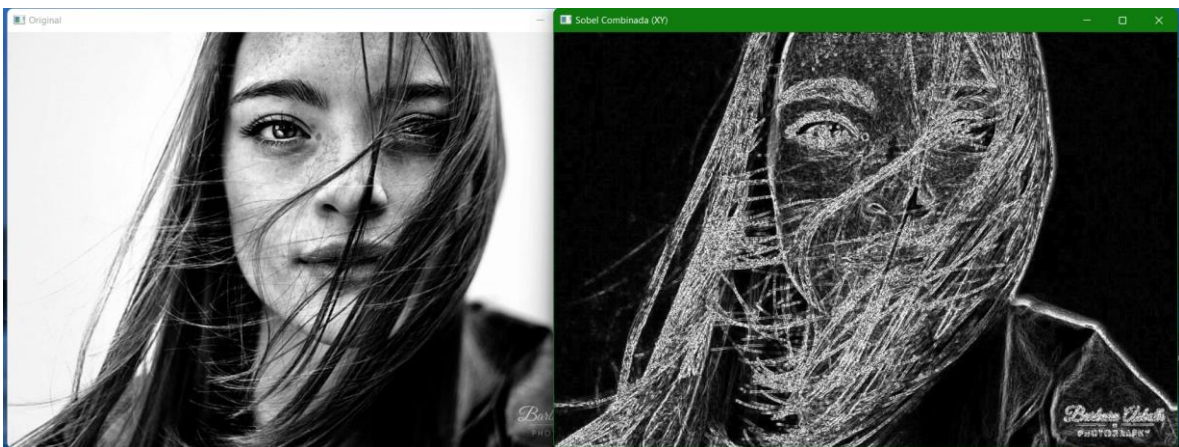
1. Sobel en X:



2. Sobel en Y:



3. Sobel en XY:



- Método Canny:

```
import numpy as np
```

```
import cv2
```

```
image = cv2.imread("FotoBella.jpg")# Leer en imagen
```

```
image = cv2.cvtColor(image,cv2.COLOR_BGR2GRAY)#Convertir imagen a imagen en escala de grises
```

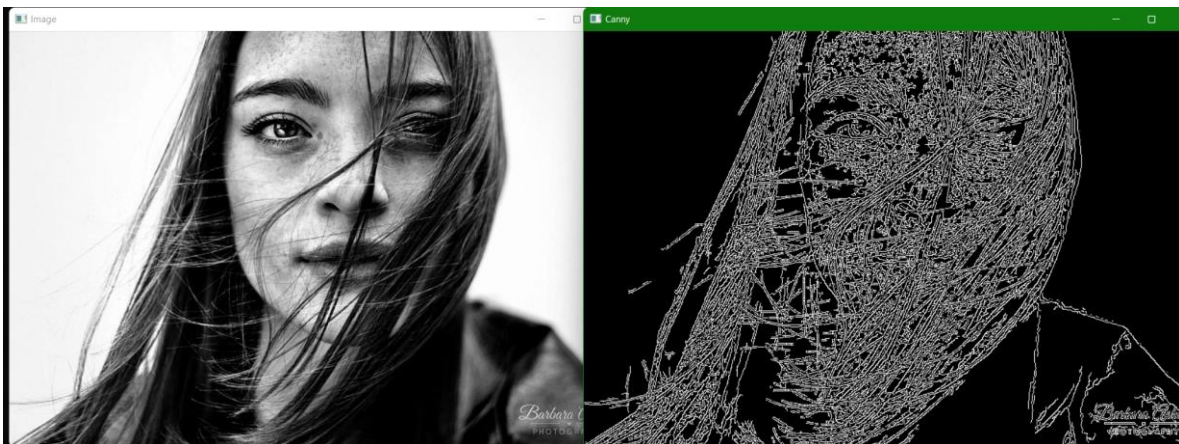
```
cv2.imshow("Image",image)#Mostrar imagen
```

```
cv2.waitKey()
```

```
canny = cv2.Canny(image,30,150)
```

```
cv2.imshow("Canny",canny)
```

```
cv2.waitKey()
```



Conclusion:

El método para mi aunque sea mas largo me gusta más el resultado sería el método Sobel en XY porque siento que es el mas preciso y la imagen que mas me gusto pero también creo que el método Canny para algunos tipos de imágenes es mejor y mas corto.