



UNIVERSIDAD NACIONAL DE TRUJILLO

Facultad de ingeniería
Programa de ingeniería mecatrónica

Práctica 06 - Procesamiento en el dominio del espacio

PROCESAMIENTO DIGITAL DE SEÑALES E IMÁGENES

ESTUDIANTE(S) :

CORTÉZ GOMEZ BRAJAN LEONEL
ESPINOZA LEÓN KARL ALEJANDRO
GUTIÉRREZ GUTIÉRREZ ITALO AARÓN

DOCENTE :

MS. ING. EMERSON MÁXIMO ASTO RODRIGUEZ

CICLO :

2023 - II

Trujillo, Perú
2023

RESUMEN

En el presente informe se resuelven ejercicios de procesamiento de imágenes, siendo estos la implementación de un algoritmo en entorno local OpenCV en 2 códigos: uno de ellos está relacionado con el empleo de un slider para controlar el nivel de suavizado de una imagen, y el otro, relacionado con la modificación de la ganancia de un filtro de alto aumento para el enfatizado de una imagen.

Objetivos:

- Familiarizarse con las operaciones de suavizado y enfatizado en el procesamiento de imágenes.
- Comprender y aplicar algoritmos para la implementación de filtros espaciales en imágenes digitales.

Requisitos:

- Computadora
- Python
- OpenCV

Instrucciones:

1. Implemente el siguiente algoritmo en un entorno local de OpenCV.

```
1. slider_max = 100
2. title_window = 'Brillo'
3.
4. def on_trackbar(val):
5.     dst = src1 + val/255
6.     cv2.imshow(title_window, dst)
7.
8. src1 = cv2.imread("../Imagenes/barbara.jpg")/255
9.
10. try:
11.     cv2.namedWindow(title_window)
12.     trackbar_name = f"Brilo {slider_max}"
13.     cv2.createTrackbar(trackbar_name, title_window, 0, slider_max, on_trackbar)
14.     on_trackbar(0)
15.
16.     while True:
17.         key = cv2.waitKey(1)
18.         if key != -1:
19.             break
20.     cv2.destroyAllWindows()
21. except:
22.     cv2.destroyAllWindows()
```

Este algoritmo se ha implementado en un entorno local de OpenCV para los siguientes 2 ítems cuyos códigos se muestran a continuación.

2. Implemente un algoritmo que le permita tener un slider para controlar el nivel de suavizado de una imagen. El slider controlara el tamaño del filtro espacial que se aplicará. Y debe poderse seleccionar si se desea un filtro promedio.

Algoritmo (Código):

```
import cv2
from scipy import signal
import numpy as np

slider_max = 16
title_window = 'Brillo'

def on_trackbar(val):

    A = val

    kernel = np.ones((2*A + 1, 2*A + 1))/((2 * A + 1)*(2*A + 1))

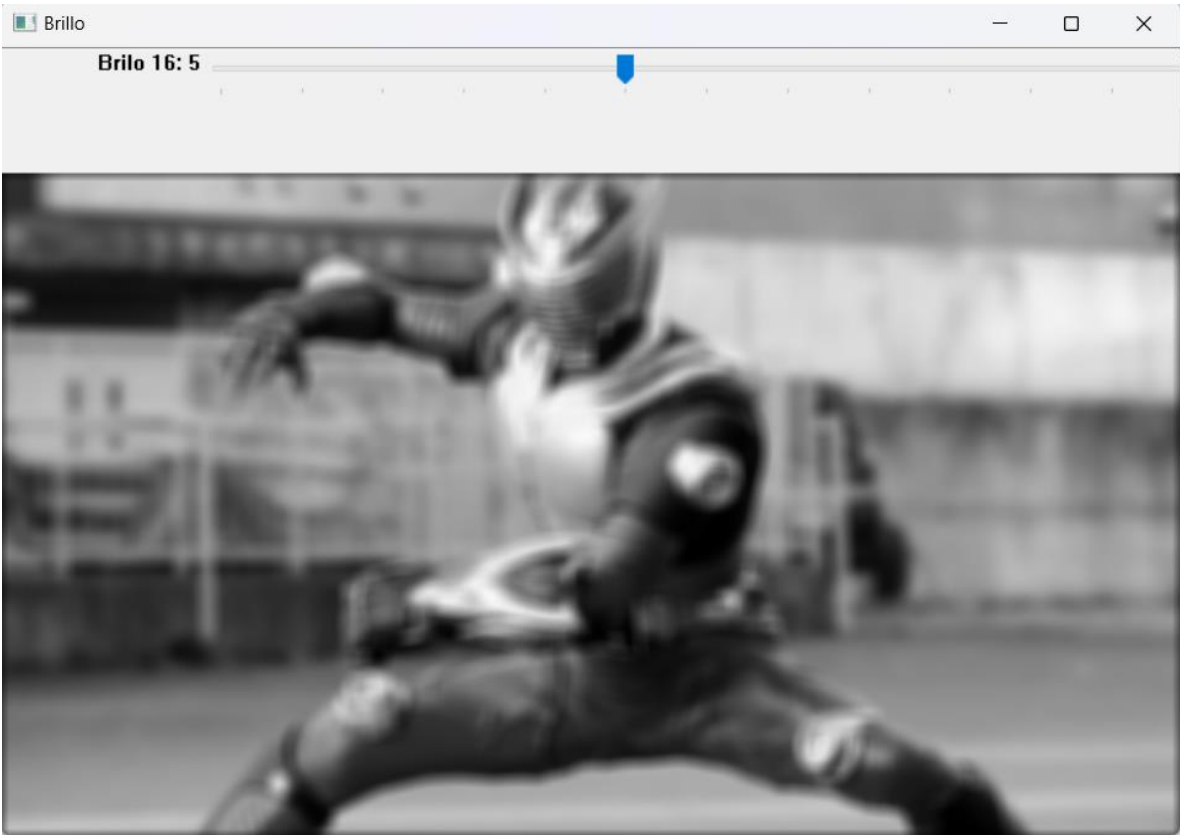
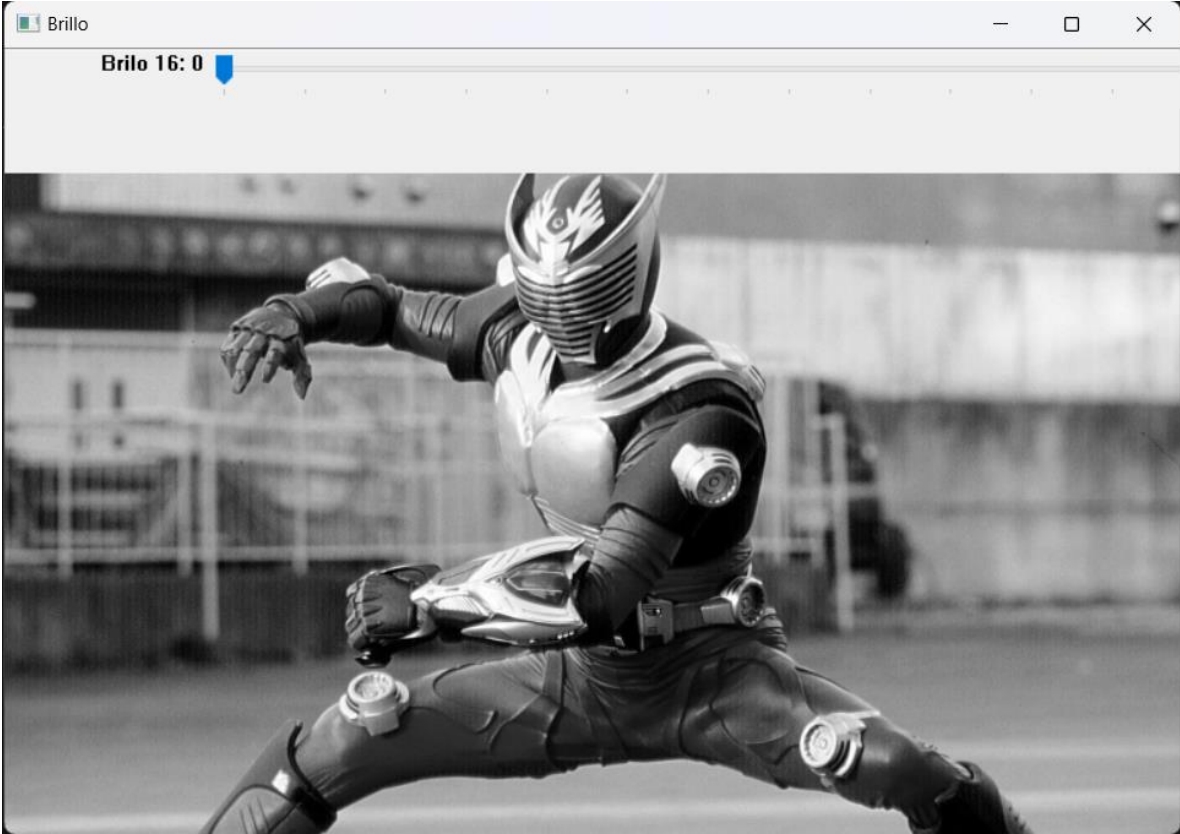
    out = signal.convolve2d(src1, kernel, mode= "same")
    cv2.imshow(title_window, out)

src1 = cv2.imread("barbara.jpg", 0)/255
width, height = src1.shape
src1 = cv2.resize(src1, (int(0.4*height),int(0.4*width)))

try:
    cv2.namedWindow(title_window)
    trackbar_name = f"Brilo {slider_max}"
    cv2.createTrackbar(trackbar_name, title_window, 1, slider_max, on_trackbar)
    on_trackbar(0)

    while True:
        key = cv2.waitKey(1)
        if key != -1:
            break
    cv2.destroyAllWindows()
except:
    cv2.destroyAllWindows()
```

Resultado de la ejecución:



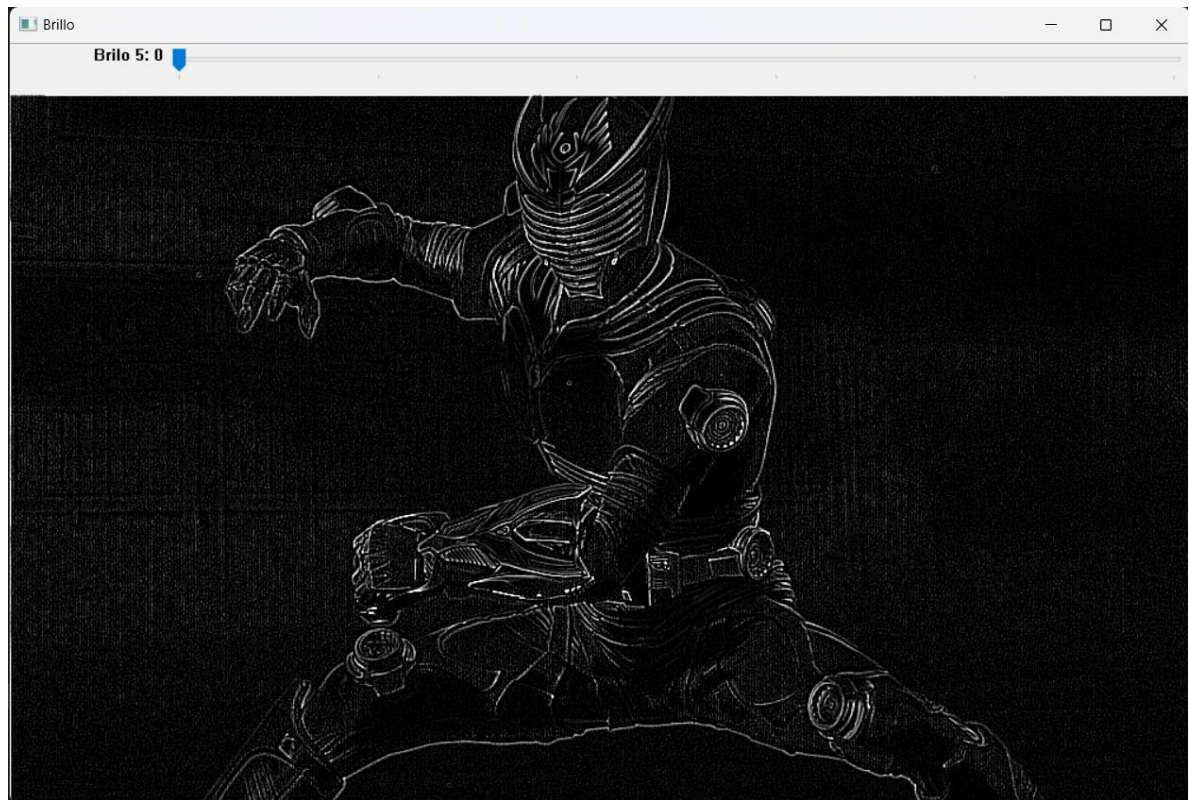
3. Implemente un algoritmo que le permita modificar la ganancia de un filtro de alto aumento. Si el slider es 0 debe verse el laplaciano, si es 1, la imagen debe quedar enfatizada. Si es mayor que 1 adicionalmente al enfatizado, la imagen debe tener mas brillo.

Algoritmo (Código):

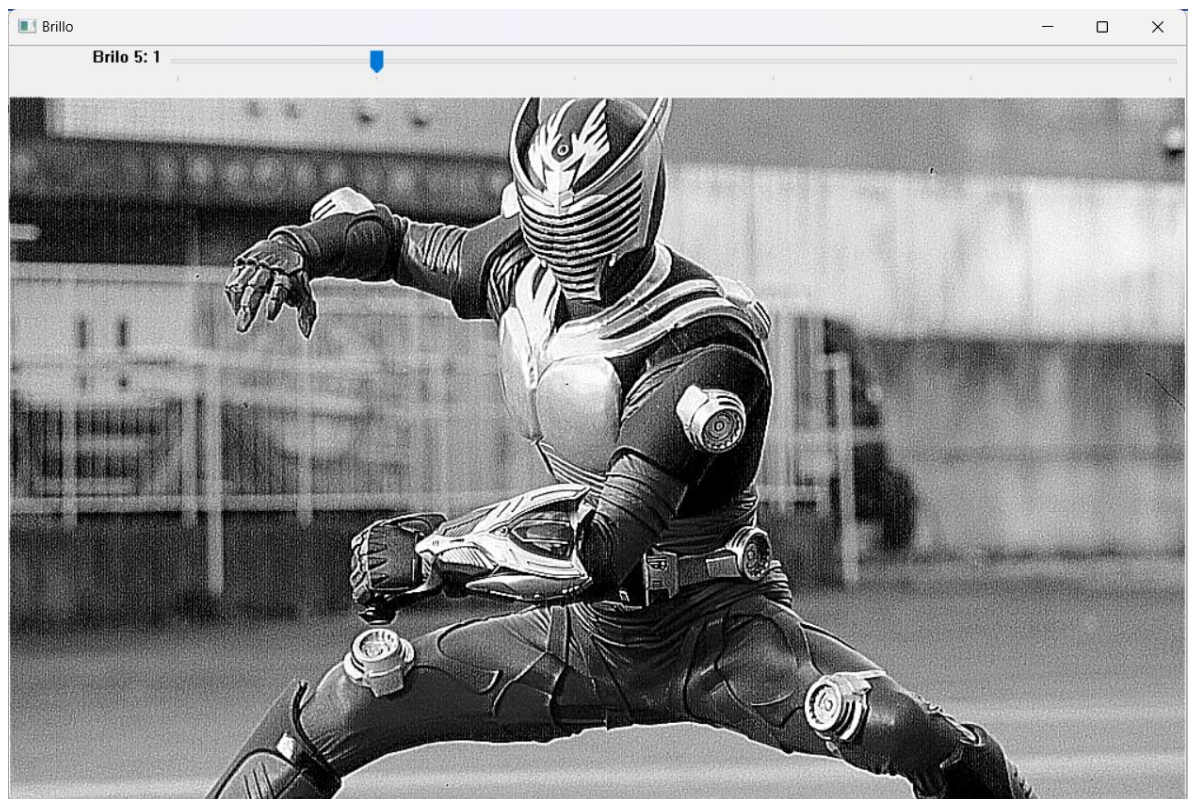
```
practica.py > ...
1  import cv2
2  from scipy import signal
3  import numpy as np
4
5  slider_max = 5
6  title_window = 'Brillo'
7
8  def on_trackbar(val):
9
10     A = val
11
12     kernel = np.array([[ -1, -1, -1], [ -1,A+8,-1], [ -1,-1,-1]])
13
14     out = signal.convolve2d(src1, kernel, mode= "same")
15     cv2.imshow(title_window, out)
16
17
18
19 src1 = cv2.imread("barbara.jpg", 0)/255
20 src1 = cv2.resize(src1, (1000,600))
21
22 try:
23     cv2.namedWindow(title_window)
24     trackbar_name = f"Brilo {slider_max}"
25     cv2.createTrackbar(trackbar_name, title_window, 0, slider_max, on_trackbar)
26     on_trackbar(0)
27
28     while True:
29         key = cv2.waitKey(1)
30         if key != -1:
31             break
32     cv2.destroyAllWindows()
33 except:
34     cv2.destroyAllWindows()
```

Resultado de la ejecución:

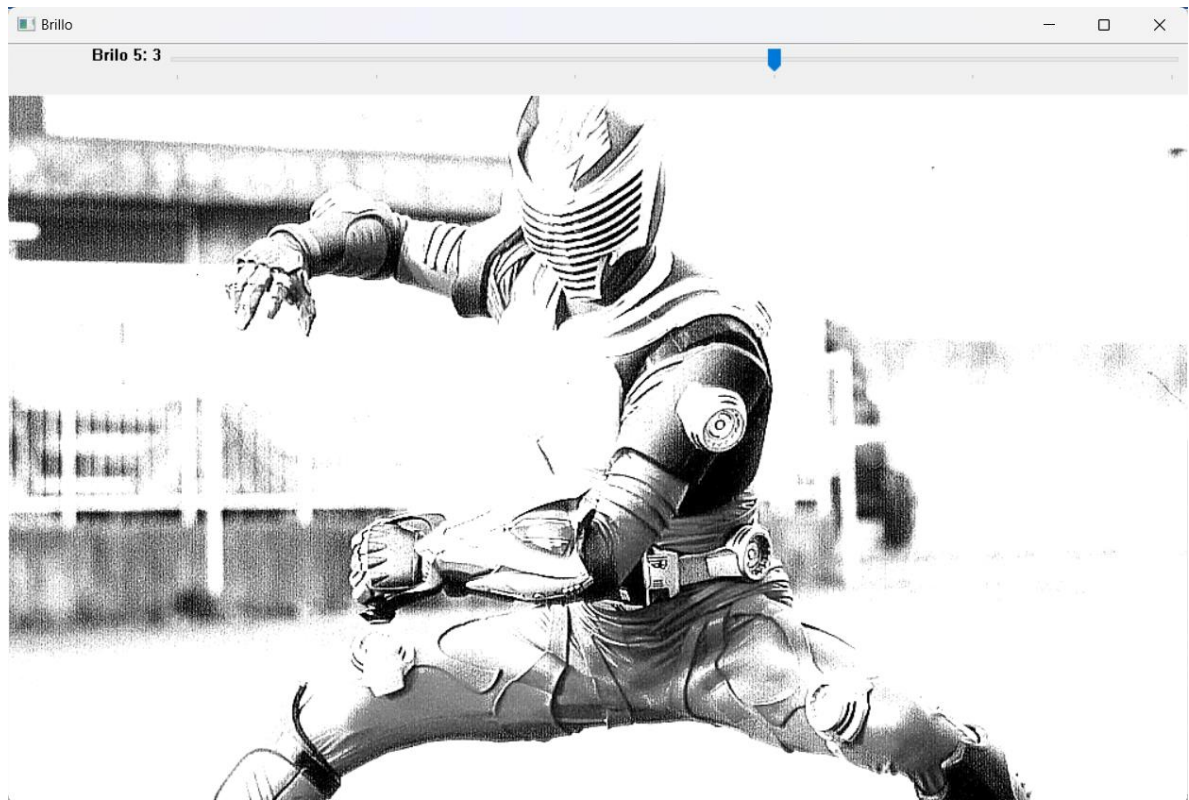
Brillo (Slider) a un nivel bajo



Brillo (Slider) a un nivel medio bajo



Brillo (Slider) a un nivel medio alto



4. BIBLIOGRAFÍA

Friendly, C. (2 de Marzo de 2021). "*Histogram matching in digital image processing*".
Obtenido de Youtube:
https://www.youtube.com/watch?v=r565euxWZBs&ab_channel=CollegeFriendly