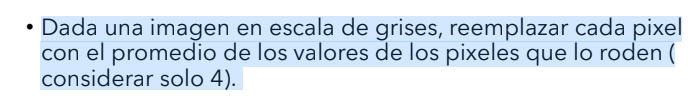
**Librerías y Paquetes: Procesamiento de imágenes Parte I**

**Guía 5**



**Introducción**

El procesamiento de imágenes es un método para realizar operaciones en imágenes con el fin de mejorar su calidad o extraer información útil de ellas. En este informe, presentaremos ejercicios detallados para manipular imágenes en escala de grises y en color utilizando Python y las librerías Numpy, PyPlot y skimage.

**Promedio de los Valores de los Píxeles Circundantes**

En una imagen en escala de grises, reemplace cada píxel con el promedio de los valores de los píxeles que lo rodean.

* Leer la Imagen en Escala de Grises

Utilizamos la función imread de la librería skimage para leer la imagen.

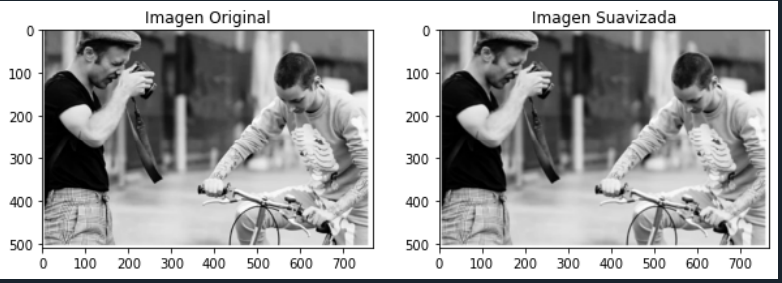
* Crear una Nueva Imagen

Se crea una copia de la imagen original para almacenar los nuevos valores calculados.

* Iterar sobre los Píxeles

Iteramos sobre cada píxel de la imagen (excluyendo los bordes) para calcular el promedio de sus píxeles circundantes y asignarlo al píxel actual en la nueva imagen

**Ejemplo**



**Código de la implementación**

# -\*- coding: utf-8 -\*-

"""

Created on Fri Jun 7 01:49:27 2024

@author: lovito99

"""

**from** **skimage** **import** io

**import** **numpy** **as** **np**

**import** **matplotlib.pyplot** **as** **plt**

**def** **imagenSuave**(image):

# Convertir la imagen a flotantes para evitar problemas de tipo de dato

NuevImagen = np.copy(image).astype(float)

**for** i **in** range(**1**, image.shape[**0**] - **1**):

**for** j **in** range(**1**, image.shape[**1**] - **1**):

# Promediar los valores de los píxeles vecinos ( utilizando solo 4 posiciones)

NuevImagen[i, j] = (image[i-**1**, j] + image[i+**1**, j] + image[i, j-**1**] + image[i, j+**1**]) / **4.0**

**return** NuevImagen

# Leer una imagen en escala de grises

Imagenbn = io.imread(r'C:\Users\lovito99\Desktop\Guia 5\practica de graficos\ejersicios\imagenbn.jpg', as\_gray=True)

# Suavizar la imagen

ImagenProcesada = imagenSuave(Imagenbn)

# Visualizar la imagen original y la imagen suavizada

fig, axes = plt.subplots(**1**, **2**, figsize=(**10**, **5**))

axes[**0**].imshow(Imagenbn, cmap='gray')

axes[**0**].set\_title('Imagen Original')

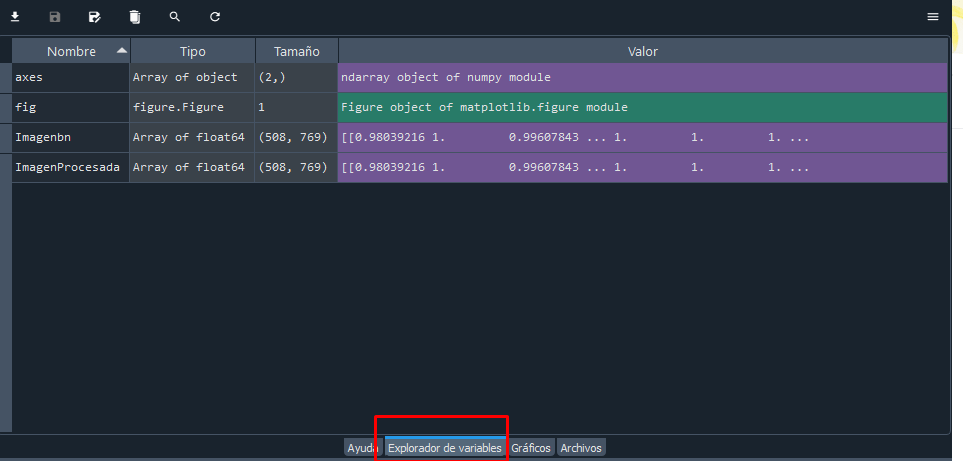
axes[**1**].imshow(ImagenProcesada, cmap='gray')

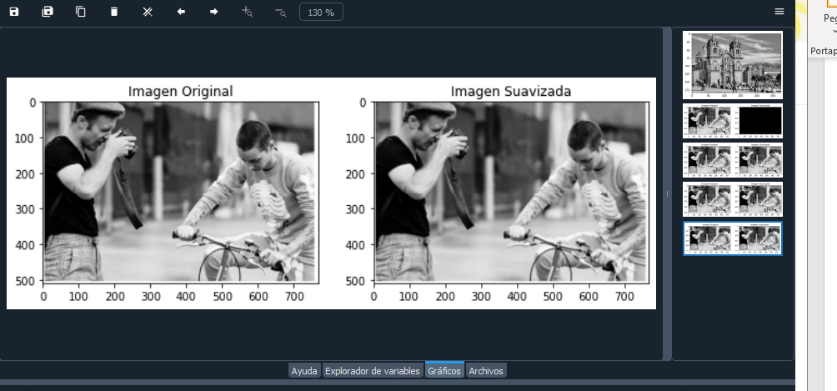
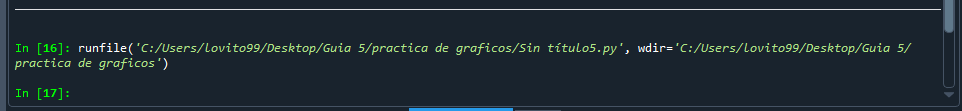
axes[**1**].set\_title('Imagen Suavizada')

plt.show()

Pruebas de la implementación

1. Explorador de variables



1. gráficos
2. Terminal

Caso B



* **Leer la Imagen RGB**

Utilizamos imread de skimage para cargar la imagen a color.

* **Reflejar la Imagen**

Invertimos las columnas de la imagen utilizando indexación de Numpy

**Explicación:**

* Lectura de la Imagen:

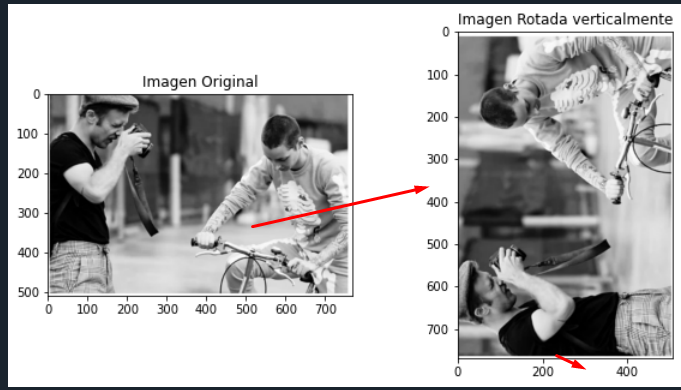
Utilizamos io.imread para leer una imagen a color.

* Reflejar la Imagen

invierte las columnas de la imagen usando indexación de Numpy (image[:, ::-1]).

* Visualización

Utilizamos plt.imshow para mostrar la imagen original y la reflejada verticalmente.



Implementación del código

"""

Created on Fri Jun 7 02:46:19 2024

@author: lovito99

"""

**from** **skimage** **import** io

**import** **numpy** **as** **np**

**import** **matplotlib.pyplot** **as** **plt**

**def** **ImengenReflejada**(image):

# Rotar la imagen 90 grados en sentido antihorario

**return** np.rot90(image)

**def** **RotarImagen**(image):

# Rotar la imagen 90 grados en sentido horario

**return** np.rot90(image, k=**3**)

# Leer una imagen a color

Imagen = io.imread(r'C:\Users\lovito99\Desktop\Guia 5\practica de graficos\ejersicios\imagenbn.jpg')

# Rotar la imagen 90 grados

Rotar = ImengenReflejada(Imagen)

# Para rotar 90 grados en sentido horario, usa la siguiente línea en su lugar:

# Visualizar la imagen original y la imagen rotada

fig, axes = plt.subplots(**1**, **2**, figsize=(**10**, **5**))

axes[**0**].imshow(Imagen)

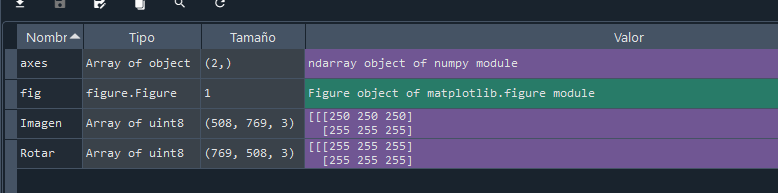
axes[**0**].set\_title('Imagen Original')

axes[**1**].imshow(Rotar)

axes[**1**].set\_title('Imagen Rotada verticalmente')

plt.show()

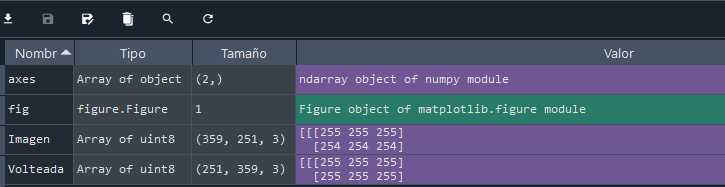
Explorador de variables

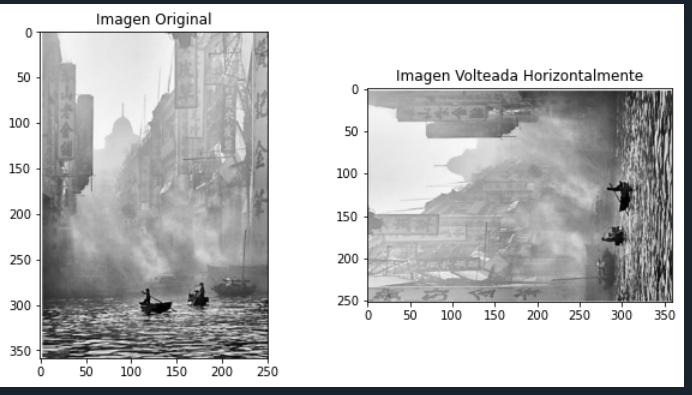




La ejecución es realizar la parte contraria del ejercicio anterior

**Volver a la posicion inicial**





Implementación del código

"""

Created on Fri Jun 7 02:46:19 2024

@author: lovito99

"""

**from** **skimage** **import** io

**import** **numpy** **as** **np**

**import** **matplotlib.pyplot** **as** **plt**

**def** **ImagenVolteada**(image):

# Girar la imagen 90 grados en sentido antihorario

Rotarimagen = np.rot90(image)

# Reflejar la imagen verticalmente

**return** np.flipud(Rotarimagen)

# Leer una imagen a color

Imagen = io.imread(r'C:\Users\lovito99\Desktop\Guia 5\practica de graficos\ejersicios\vertical.jpg')

# Voltear la imagen horizontalmente

Volteada = ImagenVolteada(Imagen)

# Visualizar la imagen original y la imagen volteada horizontalmente

fig, axes = plt.subplots(**1**, **2**, figsize=(**10**, **5**))

axes[**0**].imshow(Imagen)

axes[**0**].set\_title('Imagen Original')

axes[**1**].imshow(Volteada)

axes[**1**].set\_title('Imagen Volteada Horizontalmente')

plt.show()