Universidad De Guadalajara

CENTRO UNIVERSITARIO DE CIENCIAS EXACTAS E INGENIERIAS

ALGORITMO 1

VISION ROBOTICA

INGENIERIA EN ROBOTICA

Alumno: Carlos Omar Rodriguez Vazquez

Introducción

Este estudio se centra en la exploración práctica de los conceptos fundamentales del procesamiento de imágenes digitales. A través de una serie de ejercicios, se busca comprender la creación y manipulación de imágenes en escalas de grises y RGB. Este informe detalla los procedimientos y resultados de los ejercicios realizados, ofreciendo una visión general de las habilidades adquiridas en este campo.

Marco Teorico

OpenCV

OpenCV (Open Source Computer Vision Library) es una biblioteca de código abierto ampliamente utilizada para procesamiento de imágenes y visión por computadora. Ofrece una amplia gama de funciones para trabajar con imágenes y videos, incluyendo operaciones básicas como carga, manipulación y visualización de imágenes, así como técnicas más avanzadas como detección de objetos, seguimiento de objetos en movimiento, reconocimiento facial, entre otras. La biblioteca está escrita en C++ y cuenta con interfaces para Python, Java y otros lenguajes de programación.

```
import cv2 as cv
```

NumPy

NumPy es una biblioteca fundamental para la computación científica en Python. Proporciona un potente objeto de matriz multidimensional, junto con una amplia variedad de funciones matemáticas para operar en estas matrices. NumPy es ampliamente utilizado en el procesamiento de imágenes para manipular y procesar datos en forma de matrices, lo que lo hace ideal para la representación y manipulación de imágenes digitales en forma de matrices NumPy.

```
import numpy as np
```

Matplotlib

Matplotlib es una biblioteca de trazado en 2D en Python que produce figuras de calidad de publicación en una variedad de formatos y entornos. Es útil para visualizar datos y resultados, incluyendo imágenes generadas por computadora. Matplotlib se utiliza comúnmente junto con NumPy para mostrar imágenes y gráficos generados a partir de datos numéricos. Ofrece una amplia gama de funciones para crear gráficos estáticos, gráficos interactivos y animaciones, lo que lo convierte en una herramienta versátil para la visualización de resultados en el procesamiento de imágenes y otras áreas de la ciencia y la ingeniería.

```
import matplotlib.pyplot as plt
```

Elaboración de una imagen

Para crear una imagen digital, se utiliza una representación matricial donde cada elemento de la matriz corresponde a un píxel. En Python, la biblioteca NumPy es comúnmente utilizada para manipular matrices, lo que la convierte en una herramienta eficaz para la generación de imágenes. Por ejemplo, al crear una

imagen en escala de grises, se puede inicializar una matriz NumPy de dimensiones específicas con el método np.zeros. Esta matriz se puede considerar como un lienzo en blanco (en realidad negro) sobre el cual se pueden agregar detalles posteriormente.

```
Im1=np.zeros((filas,columnas), dtpye=uint8)
```

Dibujo de un píxel en escala de grises

Para dibujar un píxel en una imagen en escala de grises, basta con modificar el valor de un elemento específico de la matriz NumPy. Cada valor en la matriz representa la intensidad de gris de un píxel, donde 0 corresponde al negro y 255 al blanco. Al cambiar el valor de un elemento de la matriz, se altera la intensidad de gris del píxel correspondiente, permitiendo así crear puntos o detalles en la imagen.

```
Im[fila,columna] = intensidad de gris
```

Dibujo de un rectángulo en escala de grises

Para dibujar un rectángulo en una imagen en escala de grises, se modifica el valor de una región específica de la matriz NumPy, utilizando rangos para especificar las filas y columnas afectadas. Esto se logra asignando un valor de intensidad de gris a la submatriz definida por los rangos seleccionados. Al cambiar los valores de estos elementos, se crean los píxeles que conforman el rectángulo en la imagen.

```
f_i = fila inicio
f_f = fila final + 1
c_i = columna inicio
c_f = columna final + 1
Im[f_i:f_f,c_i:c_f] = intensidad de gris
```

Elaboración de una imagen RGB

En el caso de las imágenes en color, se emplea una representación tridimensional donde cada elemento de la matriz NumPy contiene los valores de los canales rojo, verde y azul (RGB) de un píxel específico. De esta manera, una imagen RGB se crea inicializando una matriz tridimensional de dimensiones específicas, donde cada dimensión corresponde a un canal de color.

```
Im=np.zeros((filas, columnas, 3), dtype=uint8)
```

Dibujo de un píxel en RGB

Para dibujar un píxel en una imagen RGB, se modifica el valor de los elementos correspondientes a los canales rojo, verde y azul de un píxel específico en la matriz NumPy tridimensional. Al cambiar estos valores, se ajusta el color del píxel, lo que permite crear puntos de diferentes colores en la imagen.

```
Im[fila,columna,canal] = intensidad de color
```

Dibujo de un rectángulo RGB

Al igual que en el caso de las imágenes en escala de grises, para dibujar un rectángulo en una imagen RGB se modifican los valores de una región específica de la matriz NumPy tridimensional, utilizando rangos para definir las filas, columnas y canales de color afectados. Mediante la asignación de valores adecuados a estos elementos, se logra crear un rectángulo de color en la imagen.

```
f_i = fila inicio
f_f = fila final + 1
c_i = columna inicio
c_f = columna final + 1
Im[f_i:f_f,c_i:c_f, canal] = intensidad de color
```

Ejercicios

1. Construye una imágen de 8x15, en escala de grises y dibuja 3 pixeles de diferentes color.

```
Im = np.zeros((8,15), dtype=np.uint8)
Im[3,5] = 200
Im[7,8] = 100
Im[2,12] = 60
```

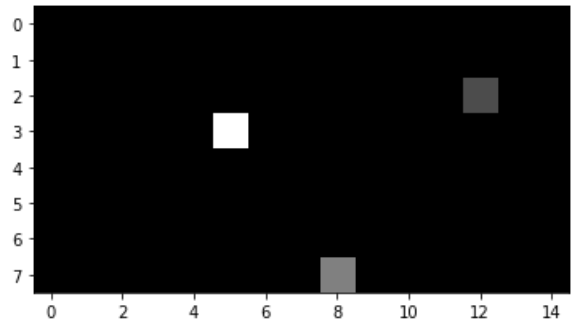

Figura 1: Imágen del ejercicio 1.

El codigo crea una matriz de 8x15 y modifica los pixeles (3,5), (7,8) y (2,12) a diferentes intesidades de gris.

2. Construye una imágen de 12x6, en escalas de grises y dibuja 2 rectángulos.

```
Im2 = np.zeros((12,6), dtype=np.uint8)
Im2[3:6,1:3] = 100
Im2[8:11,2:5] = 50
```

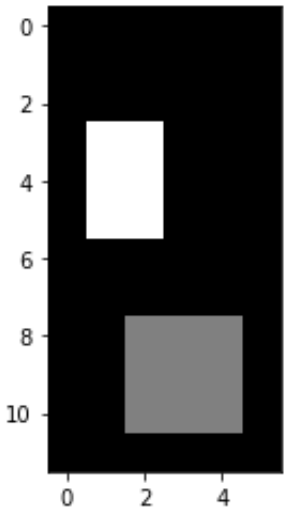

Figura 2: Imágen del ejercicio 2.

El codigo crea una matriz de 12x6 y modifica los pixeles del rectangulo que inicia en (3,1) y termina en (5,2) y el rectangulo que incia en (8,2) y termina en (10,4) a diferentes intensidades de gris.

3. Construye una imágen de 9x8, RGB y dibuja 3 pixeles de diferente color.

```
Im3 = np.zeros((9,8,3), dtype=np.uint8)
Im3[3,1,0] = 150
Im3[7,6,1] = 200
Im3[2,4,2] = 180
```

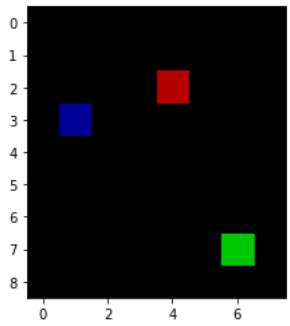

Figura 3: Imágen del ejercicio 3.

El codigo crea una matriz de 9x8x3 y modifica el pixel (3,1) la intensidad de azul, el pixel (7,6) la intensidad de verde y el pixel (2,4) la intesidad de rojo.

4. Construye una imagen RGB del tamaño que quieras y dibuja 3 rectángulos de diferente color.

```
Im4 = np.zeros((10,7,3), dtype=np.uint8)
Im4[0:10,0:7,2] = 255
Im4[1:9,1:6,0] = 255
Im4[1:9,1:6,1] = 255
Im4[1:9,1:6,2] = 255
Im4[9,2,1] = 255
Im4[6:9,2,0] = 0
Im4[3:8,3,0] = 0
Im4[1:5,4,0] = 0
Im4[0,4,1] = 255
```

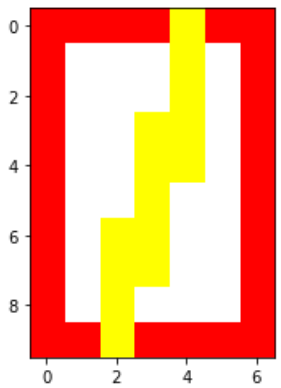

Figura 4: Imágen del ejercicio 4

Para este ejercicio realice el escudo de mi superheroe favorito "Flash".Primero dibuje un rectangulo de toda la imagen con intensidad 255 en el canal 2 el cual es un rojo, despues dibuje un rectangulo blanco, para esto le di valores de 255 en todos los canalaes. Para pintar el color amarillo en necesario darle valores altos unicamente al canal 1 y 2 (verde y rojo), asi que para los pixeles en blanco le di valores de 0 al canal 0 (azul) y a los pixeles rojos le agregue valores de 255 al canal 1 (verde).

Conclusión

En este estudio, se exploraron los conceptos fundamentales del procesamiento de imágenes digitales a través de una serie de ejercicios prácticos. Se utilizó una combinación de las bibliotecas OpenCV, NumPy y Matplotlib en Python para realizar operaciones de manipulación, análisis y visualización de imágenes. A través de la elaboración de imágenes en escalas de grises y RGB, así como la creación de puntos y rectángulos de diferentes colores, se adquirieron habilidades básicas en el manejo de imágenes digitales. Estos ejercicios proporcionaron una comprensión práctica de los conceptos teóricos presentados, y sirvieron como una introducción sólida al procesamiento de imágenes en entornos de programación. Se espera que este conocimiento sea de utilidad en futuros proyectos y aplicaciones relacionados con el procesamiento de imágenes y la visión por computadora.