



UANL

**UNIVERSIDAD AUTONOMA DE NUEVO
LEON**

**FACULTAD DE INGENIERIA MECANICA Y
ELECTRICA**



FIME

Nombre: Roberto Erick Aguilar Morales

Matricula: 1871004

Carrera: Ingeniero en Tecnologías del Software

1.- Estructura de imágenes

Materia: VISION COMPUTACIONAL LABORATORIO

Docente: RAYMUNDO SAID ZAMORA PEQUEÑO

Hora: N1-N2

Días: Miércoles

Fecha: 06/10/24

Objetivo

El objetivo de esta práctica es explorar la manipulación de imágenes usando OpenCV con Python. A partir de una imagen a color, se obtendrán sus matrices correspondientes a los canales de color RGB. Luego, se convertirá la imagen a escala de grises, extrayendo su matriz correspondiente. Estos datos se guardarán en archivos CSV para su visualización posteriormente y comprobación con las mismas imágenes

Marco teórico

Introducción a la Visión Computacional

La visión computacional es un campo de la inteligencia artificial y la informática que permite a las máquinas interpretar y comprender el mundo visual. A través de técnicas avanzadas logran dar respuestas cercanas, las computadoras pueden procesar imágenes y videos para reconocer patrones, identificar objetos y realizar análisis complejos. Este campo tiene aplicaciones en diversas áreas, como la medicina, la robótica, entretenimiento, la seguridad, la automoción y la industria.

OpenCV: Biblioteca para el Procesamiento de Imágenes

OpenCV (Open Source Computer Vision Library) es una biblioteca de código abierto que proporciona herramientas para el procesamiento de imágenes y la visión computacional. Es ampliamente utilizada en la investigación y la industria debido a su eficiencia y la variedad de algoritmos que soporta. Entre sus capacidades se encuentran la detección de rostros, la identificación de objetos, la manipulación de imágenes, la calibración de cámaras y la reconstrucción 3D.

Tratamiento de Imágenes

- **Imagen Digital:** Es una representación bidimensional de una escena del mundo real en formato digital. Las imágenes digitales están formadas por píxeles, que son las unidades mínimas de color.
- **Modelo de Color RGB:** Es un modelo aditivo en el que los colores se generan combinando tres componentes primarios: rojo (Red), verde (Green) y azul (Blue). Cada color en una imagen digital se representa como una combinación de estos tres componentes.
- **Escala de Grises:** Es una representación de una imagen donde cada píxel tiene un valor de intensidad que varía entre el negro y el blanco, pasando por diferentes tonos de gris. La escala de grises es útil en aplicaciones donde la información de color no es relevante y se requiere un análisis de contraste o forma.

Procesamiento de Imágenes con OpenCV

El procesamiento de imágenes implica operaciones como la manipulación de píxeles, la conversión de formatos de color y la extracción de información útil. En esta práctica, se utilizarán las siguientes operaciones:

- **Carga y visualización de imágenes:** OpenCV permite leer y mostrar imágenes con funciones sencillas como `cv2.imread()` y `cv2.imshow()`.
- **Separación de canales de color:** Las imágenes en formato RGB pueden dividirse en sus componentes individuales utilizando `cv2.split()`. Esto permite el análisis separado de cada canal de color.
- **Conversión a escala de grises:** Con la función `cv2.cvtColor()`, se puede convertir una imagen de color en una imagen en escala de grises, simplificando su análisis y procesamiento posterior.
- **Almacenamiento de datos en formato CSV:** Para facilitar el análisis y la visualización, las matrices de los canales de color y la imagen en escala de grises se guardan en archivos CSV utilizando `numpy.savetxt()`. Estos archivos pueden ser analizados con herramientas como Excel o MATLAB.

Desarrollo

Importación de Librerías

El primer paso consiste en importar las librerías necesarias para la manipulación y procesamiento de imágenes. En este caso, se utilizan `cv2` para el manejo de imágenes y `numpy` para la manipulación de matrices.

```
import cv2
import numpy as np
```

Carga de la Imagen

Se utiliza la función `cv2.imread()` para cargar la imagen en color desde el archivo. Es importante verificar que la imagen se haya cargado correctamente. Si no se puede cargar, se imprime un mensaje de error.

```
imagen = cv2.imread("Incendio.png")
if imagen is None:
    print("Error: No se pudo cargar la imagen. Asegúrate de que el archivo 'Incendio.png' está en el directorio correcto.")
```

Separación de Canales RGB

La imagen cargada se descompone en sus canales de color: azul (B), verde (G) y rojo (R) utilizando la función `cv2.split()`. Esto permite analizar y manipular cada canal de color por separado.

```
b, g, r = cv2.split(imagen)
```

Guardado de las Matrices de Color en Archivos CSV

Cada canal de color se guarda en un archivo CSV usando `numpy.savetxt()`. Estos archivos CSV contienen la matriz de cada canal, facilitando el análisis de los valores de los píxeles en otras herramientas o programas.

```
np.savetxt('canal_azul.csv', b, delimiter=',', fmt='%d')
np.savetxt('canal_verde.csv', g, delimiter=',', fmt='%d')
np.savetxt('canal_rojo.csv', r, delimiter=',', fmt='%d')
print("Los canales de color se han guardado en 'canal_azul.csv',
'canal_verde.csv', y 'canal_rojo.csv'.")
```

Visualización de la Imagen Original

La imagen original en color se muestra en una ventana utilizando `cv2.imshow()`. Esta función permite visualizar la imagen cargada y compararla con las versiones manipuladas.

```
cv2.imshow("Imagen a Color", imagen)
```

Conversión a Escala de Grises

La imagen a color se convierte a escala de grises mediante la función `cv2.cvtColor()` de OpenCV. En esta conversión, se elimina la información de color, manteniendo solo la intensidad de los píxeles.

```
gray_image = cv2.cvtColor(imagen, cv2.COLOR_BGR2GRAY)
```

Guardado de la Matriz en Escala de Grises en un Archivo CSV

La matriz resultante de la conversión a escala de grises se guarda en un archivo CSV utilizando `numpy.savetxt()`. Esto permite analizar cómo se distribuyen las intensidades de gris en la imagen.

```
np.savetxt('imagen_gris.csv', gray_image, delimiter=',', fmt='%d')
print("La matriz de la imagen en escala de grises se ha guardado en
'imagen_gris.csv'.")
```

Visualización de la Imagen en Escala de Grises

Finalmente, la imagen en escala de grises se muestra en una ventana independiente. Esto facilita la comparación visual con la imagen original en color.

Una vez que se ha mostrado la imagen, se utilizan las funciones `cv2.waitKey(0)` para esperar la pulsación de una tecla y `cv2.destroyAllWindows()` para cerrar todas las ventanas abiertas.

```
cv2.imshow("Imagen a Blanco y negro", gray_image)
cv2.waitKey(0)
cv2.destroyAllWindows()
```

Cálculos y resultados

```
import cv2
import numpy as np

# Lee la imagen
imagen = cv2.imread("Incendio.png")
if imagen is None:
    print("Error: No se pudo cargar la imagen. Asegúrate de que el archivo 'Incendio.png' está en el directorio correcto.")
else:
    # Divide la imagen en sus canales B, G y R
    b, g, r = cv2.split(imagen)

    # Guardar cada canal en archivos CSV separados
    np.savetxt('canal_azul.csv', b, delimiter=',', fmt='%d')
    np.savetxt('canal_verde.csv', g, delimiter=',', fmt='%d')
    np.savetxt('canal_rojo.csv', r, delimiter=',', fmt='%d')
    print("Los canales de color se han guardado en 'canal_azul.csv', 'canal_verde.csv', y 'canal_rojo.csv'.")

    # Mostrar la imagen original
    cv2.imshow("Imagen a Color", imagen)

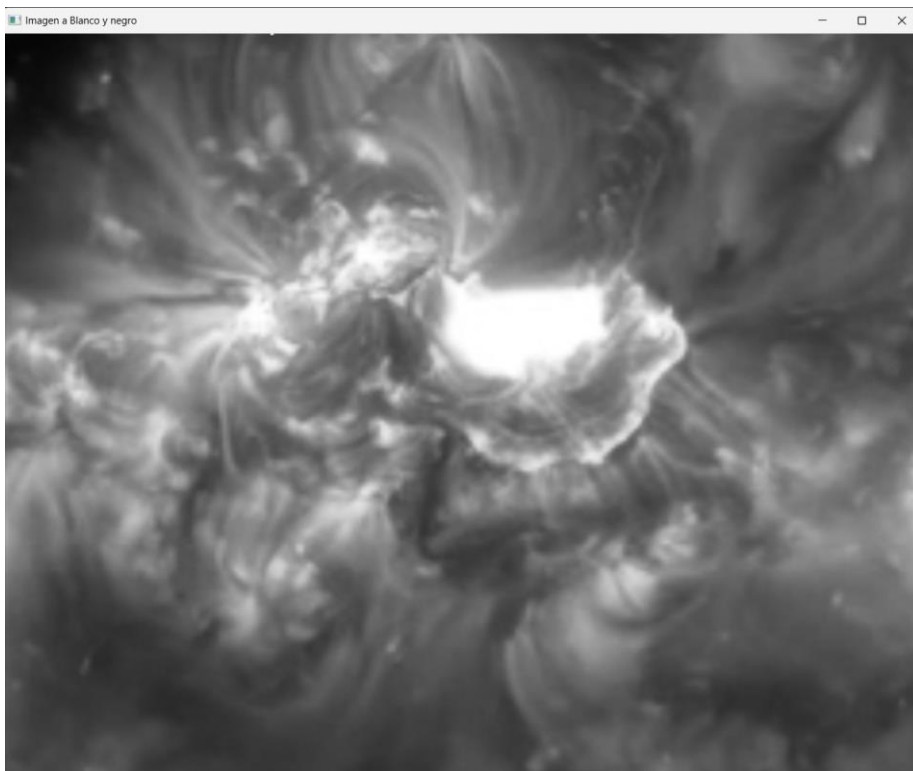
    # Convertir la imagen a escala de grises
    gray_image = cv2.cvtColor(imagen, cv2.COLOR_BGR2GRAY)

    # Guardar la matriz de la imagen en escala de grises en un archivo CSV
    np.savetxt('imagen_gris.csv', gray_image, delimiter=',', fmt='%d')
    print("La matriz de la imagen en escala de grises se ha guardado en 'imagen_gris.csv'.")

    # Mostrar la imagen en blanco y negro
    cv2.imshow("Imagen a Blanco y negro", gray_image)
```



```
cv2.waitKey(0)  
cv2.destroyAllWindows()
```



canal_rojo.csv • Guardado en Este PC

canal_verde.csv • Guardado en Este PC

	A	B	C	D	E	F	G	H	I	J	K	L	M
1	0	1	1	2	3	3	2	2	2	2	2	1	1
2	0	1	1	2	3	3	2	2	2	2	2	1	1
3	0	1	1	2	3	3	2	2	2	2	2	1	1
4	0	0	1	1	2	2	2	1	1	1	1	1	1
5	0	0	1	1	2	2	1	1	1	1	1	1	1
6	0	0	1	1	1	1	1	1	1	1	1	1	1
7	0	0	1	1	1	1	1	1	1	1	1	1	1
8	0	0	1	1	1	1	1	1	1	1	1	1	1
9	0	0	1	1	1	1	1	1	1	1	1	1	1
10	0	0	1	1	1	1	1	1	1	1	1	1	1
11	0	0	1	1	1	1	1	1	1	1	1	1	1
12	0	0	1	1	1	1	1	1	1	1	1	1	1
13	0	0	1	1	1	1	1	1	1	1	1	1	1
14	0	0	1	1	1	1	1	1	1	1	1	1	1
15	0	0	1	1	1	1	1	1	1	1	1	1	1
16	0	0	1	1	1	1	1	1	1	1	1	1	1
17	0	0	1	1	1	1	1	1	1	1	1	1	1
18	0	0	1	1	1	1	1	1	1	1	1	1	1
19	0	0	1	1	1	1	1	1	1	1	1	1	1
20	0	0	1	1	1	1	1	1	1	1	1	1	1
21	0	0	1	1	1	1	1	1	1	1	1	1	1
22	0	0	1	1	1	1	1	1	1	1	1	1	1
23	0	0	1	1	1	1	1	1	1	1	1	1	1
24	0	0	1	1	1	1	1	1	1	1	1	1	1
25	0	0	1	1	1	1	1	1	1	1	1	1	1
26	0	0	1	1	1	1	1	1	1	1	1	1	1
27	0	0	1	1	1	1	1	1	1	1	1	1	1
28	0	0	1	1	1	1	1	1	1	1	1	1	1
29	0	0	1	1	1	1	1	1	1	1	1	1	1
30	0	0	1	1	1	1	1	1	1	1	1	1	1
31	0	0	1	1	1	1	1	1	1	1	1	1	1
32	0	0	1	1	1	1	1	1	1	1	1	1	1

imagen_gris.csv • Guardado en Este PC

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
1	3	4	4	6	8	8	8	8	8	8	8	7	7	7	7	7	7	7
2	2	4	5	6	8	8	8	8	8	8	8	7	7	7	7	7	7	7
3	2	4	5	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7
4	2	3	4	5	7	7	7	7	7	7	7	7	7	7	7	7	7	7
5	2	3	4	5	6	7	7	7	7	7	7	7	7	7	7	7	7	7
6	2	3	4	5	6	6	7	7	7	7	7	7	7	7	7	7	6	6
7	2	3	4	5	6	6	7	7	7	7	7	7	7	7	7	7	6	6
8	2	3	4	5	5	6	7	7	7	7	7	7	7	7	7	6	6	6
9	2	3	4	5	6	6	7	7	7	7	7	7	6	6	6	6	6	6
10	2	3	4	5	6	6	7	7	7	7	7	6	6	6	6	5	5	5
11	2	3	4	5	6	7	7	7	7	7	7	6	6	6	5	5	5	5
12	2	2	4	4	5	6	6	7	7	7	7	6	6	5	5	5	5	5
13	2	3	4	4	5	6	6	6	7	7	6	6	5	5	5	5	5	5
14	2	2	4	4	5	6	6	6	6	6	6	6	5	5	5	5	5	6
15	2	2	3	4	5	5	6	6	6	6	6	6	5	5	5	5	5	5
16	1	2	3	4	4	5	6	5	6	6	5	5	5	5	5	5	5	5
17	1	2	3	3	4	4	5	5	5	6	5	5	5	5	5	5	5	5
18	1	2	2	3	4	4	4	5	5	5	5	5	5	5	5	5	5	5
19	1	2	2	3	4	4	4	5	5	5	5	5	5	5	5	5	6	6
20	1	2	2	3	3	3	4	4	5	5	5	5	5	5	5	6	6	6
21	1	2	2	3	3	3	4	4	5	5	5	5	5	5	6	6	6	7
22	1	2	2	3	3	3	4	4	5	5	5	5	6	7	7	7	7	8
23	2	2	3	3	3	4	4	4	5	5	5	5	6	7	8	8	8	8
24	2	2	3	3	3	4	4	4	5	5	5	5	6	7	8	8	8	8
25	2	3	3	3	4	4	4	5	5	5	5	6	7	7	8	8	8	8
26	2	2	3	3	3	4	4	4	5	5	5	6	6	7	8	8	8	9
27	2	2	3	3	3	4	4	4	5	5	5	6	7	7	8	8	8	9
28	2	2	3	3	3	4	4	4	5	5	6	7	7	8	8	8	9	10
29	2	2	3	3	3	4	4	4	5	5	6	7	7	8	8	8	9	10
30	2	2	3	3	4	4	5	5	6	7	7	8	8	8	8	9	9	10
31	2	2	3	4	5	5	6	6	6	7	7	7	7	9	9	9	9	1
32	2	2	3	4	6	6	6	6	6	6	6	7	7	7	8	9	9	1

Conclusiones

Antes de esta práctica, el conocimiento sobre la manipulación de imágenes y el uso de OpenCV era limitado. Ahora, se ha adquirido una comprensión práctica sobre cómo cargar, visualizar y manipular imágenes en Python. Se ha aprendido a separar los canales de color RGB y a convertir imágenes a escala de grises, comprendiendo la importancia de cada formato en diferentes aplicaciones. Además, se ha ganado experiencia en el almacenamiento de matrices en archivos CSV para su análisis. Esta práctica ha transformado conceptos teóricos en habilidades concretas, fortaleciendo las bases para proyectos avanzados en visión computacional.

Bibliografía

OpenCV modules. OpenCV. (n.d.). <https://docs.opencv.org/4.x/index.html>

¿Qué es la visión artificial?. IBM. (2024, January 11). <https://www.ibm.com/mx-es/topics/computer-vision>