



Rodrigo Hernández Zavala 08

de marzo de 2025

Ing. Mecatrónica

22310215

Funcionamiento, Recomendaciones y Posibles mejoras de la practica 1

Visión Artificial

Objetivo de la Práctica: Carga y Visualización de Imágenes con OpenCV

El objetivo principal de esta práctica es **aprender a cargar, visualizar y manipular imágenes en OpenCV**, comprendiendo el manejo de formatos de color y la importancia de la conversión entre espacios de color.

Objetivos Específicos:

Comprender la carga de imágenes en distintos formatos (escala de grises y color).

Distinguir la diferencia entre los espacios de color BGR y RGB y cuándo es necesario convertir entre ellos.

Aprender a visualizar imágenes correctamente en OpenCV y Matplotlib.

Explorar el impacto del formato de color en la percepción de imágenes en visión artificial.

Resultado Esperado:

Al final de la práctica, el estudiante podrá cargar imágenes en **escala de grises y color**, mostrarlas correctamente en OpenCV y Matplotlib, y entender la necesidad de la conversión de espacios de color en diferentes entornos.

Carga y Visualización de Imágenes en OpenCV Explicación

OpenCV permite cargar imágenes en diferentes formatos:

- **Escala de grises:** `cv2.IMREAD_GRAYSCALE`
- **Color (BGR):** `cv2.IMREAD_COLOR`
- **Sin cambios:** `cv2.IMREAD_UNCHANGED`

Una vez cargada, la imagen puede mostrarse con `cv2.imshow()`, pero OpenCV usa el formato **BGR** en lugar de **RGB**.

Recomendaciones y Observaciones

-Siempre verifica si la imagen se cargó correctamente: `if`

`img is None:`

```
print("Error: No se pudo cargar la imagen.")
```

```
exit()
```

-Para mostrar imágenes con **Matplotlib**, convierte de BGR a RGB con:

```
img_rgb = cv2.cvtColor(img, cv2.COLOR_BGR2RGB) plt.imshow(img_rgb)  
plt.show()
```

Diferencia entre BGR y RGB Explicación

- OpenCV usa **BGR** (Blue-Green-Red) como formato por defecto.
- Otras herramientas como **Matplotlib, PIL y muchas APIs gráficas** usan **RGB** (Red-Green-Blue).
- Un píxel rojo en RGB es (255, 0, 0), pero en BGR es (0, 0, 255).

Recomendaciones y Observaciones

-Si trabajas solo con **OpenCV**, no necesitas convertir el formato.

-Si vas a usar **Matplotlib u otras bibliotecas gráficas**, usa la conversión:

```
img_rgb = cv2.cvtColor(img_bgr, cv2.COLOR_BGR2RGB)
```

Percepción del Color en Visión Artificial Explicación

La visión humana interpreta los colores basándose en tres componentes:

- **Matiz (Hue):** Tipo de color (rojo, verde, azul, etc.).
- **Saturación (Saturation):** Intensidad o pureza del color.
- **Brillo (Brightness o Value):** Cantidad de luz reflejada.

Recomendaciones y Observaciones

Para **segmentación de colores**, se recomienda convertir la imagen a **HSV** en lugar de usar BGR o RGB, ya que el matiz es más fácil de diferenciar. `img_hsv = cv2.cvtColor(img_bgr, cv2.COLOR_BGR2HSV)`

Percepción Humana y Procesamiento de Imágenes Explicación

- **Percepción Acromática:** Detección de tonos de gris en baja iluminación (visión nocturna).

- **Sensibilidad Luminosa:** Adaptación del ojo a diferentes niveles de luz.
- **Contraste:** Diferencia de brillo entre un objeto y su fondo.

Recomendaciones y Observaciones

-Para mejorar el contraste en imágenes digitales, usa **ecualización de histograma** con OpenCV:

```
img_eq = cv2.equalizeHist(img_gray)
```

-Si trabajas con imágenes en escala de grises, recuerda que los valores van de **0 (negro) a 255 (blanco)**.

Concepto	Explicación	Recomendaciones
Carga de imágenes	OpenCV usa <code>cv2.imread()</code> con diferentes modos (color, escala de grises, etc.).	Siempre verifica que la imagen se haya cargado correctamente.
BGR vs. RGB	OpenCV usa BGR , mientras que Matplotlib usa RGB .	Convierte con <code>cv2.cvtColor(img, cv2.COLOR_BGR2RGB)</code> si usas Matplotlib.
Matiz y Saturación	Definen el tipo y la intensidad del color.	Usa el espacio de color HSV para segmentación de colores.
Percepción luminosa	El ojo se adapta a la luz y el contraste.	Usa ecualización de histograma para mejorar imágenes con bajo contraste.

Conclusión

-**OpenCV es una herramienta poderosa para la visión artificial**, pero es importante conocer cómo maneja los colores y la iluminación.

-**El formato BGR vs. RGB es clave** cuando trabajamos con otras bibliotecas de visualización como Matplotlib.

-**El ojo humano tiene diferentes mecanismos de percepción**, y conocerlos ayuda a mejorar el procesamiento digital de imágenes en sistemas de visión artificial.

