



Rodrigo Hernández Zavala

15 de marzo de 2025

Ing. Mecatrónica

22310215

Manipulación de Píxeles

Visión Artificial

Explicación del Código

Este código en **Python** usa **OpenCV** para manipular píxeles en una imagen. Se modifican colores, se copian regiones y se muestran propiedades de la imagen.

Paso a Paso del Código

Importar Librerías

```
python
CopiarEditar
import cv2
import numpy as np
```

- ◆ **cv2 (OpenCV)** → Para leer y manipular imágenes.
 - ◆ **numpy (np)** → Para manejar matrices de píxeles.
-

Cargar la Imagen

```
python
CopiarEditar
img = cv2.imread('watch.jpg', cv2.IMREAD_COLOR)
```

- ◆ **Carga la imagen "watch.jpg" en color (BGR).**
- ◆ Si la imagen no se encuentra, `img` será `None`.

✅ **Mejora:** Verificar si la imagen se cargó correctamente.

```
python
CopiarEditar
if img is None:
    print("Error: No se pudo cargar la imagen.")
    exit()
```

Acceder y Modificar un Píxel Específico

```
python
CopiarEditar
px = img[55,55] # Obtiene el color del píxel en (55,55)
img[55,55] = [255,255,255] # Cambia el píxel a blanco
px = img[55,55] # Vuelve a leer el píxel modificado
print(px) # Imprime el nuevo valor del píxel
```

- ◆ **Antes de modificar:** `px` almacena el color original del píxel (B, G, R).
- ◆ **Después de modificar:** El píxel (55,55) se vuelve blanco [255, 255, 255].

✅ **Ejemplo de salida antes y después:**

```
css
CopiarEditar
[34, 56, 200] # (Ejemplo de color antes de la modificación)
[255, 255, 255] # Píxel cambiado a blanco
```

4. Seleccionar y Modificar una Región de la Imagen

```
python
CopiarEditar
px = img[100:150,100:150] # Selecciona una región de 50x50 píxeles
print(px) # Imprime los valores de los píxeles en la región
img[100:150,100:150] = [255,255,255] # Convierte toda la región en blanco
```

- ◆ `img[100:150,100:150]` → Accede a un **bloque de píxeles** dentro del rango (**y:100-150, x:100-150**).
 - ◆ Se reemplaza **cada píxel en esa área por [255,255,255] (blanco)**.
-

5. Imprimir Propiedades de la Imagen

```
python
CopiarEditar
print(img.shape)
print(img.size)
print(img.dtype)
```

- ◆ `img.shape` → Devuelve (**altura, ancho, número de canales**).
- ◆ `img.size` → Número total de píxeles en la imagen.
- ◆ `img.dtype` → Tipo de datos de los píxeles (normalmente `uint8`).

✅ **Ejemplo de salida:**

```
scss
CopiarEditar
(300, 400, 3) # Imagen de 300x400 píxeles con 3 canales (BGR)
360000 # Total de píxeles (300x400x3)
uint8 # Cada valor de píxel está almacenado como un número de 8 bits (0-255)
```

📄 Copiar y Mover una Región de la Imagen

```
python
CopiarEditar
watch_face = img[37:111,107:194] # Extrae una región de la imagen
img[0:74,0:87] = watch_face # Copia esa región a la esquina superior
izquierda
```

- ♦ `watch_face = img[37:111,107:194]` → **Selecciona un fragmento de la imagen.**
- ♦ `img[0:74,0:87] = watch_face` → **Mueve esa sección a otra parte de la imagen (superior izquierda).**

✅ Ejemplo:

Si la imagen es un reloj, esta línea copia la parte del **rostro del reloj** y la pega en la esquina superior izquierda.

📄 Mostrar la Imagen Modificada

```
python
CopiarEditar
cv2.imshow('image', img)
cv2.waitKey(0)
cv2.destroyAllWindows()
```

- ♦ **Muestra la imagen modificada en una ventana.**
 - ♦ `cv2.waitKey(0)` → Espera hasta que el usuario presione una tecla.
 - ♦ `cv2.destroyAllWindows()` → Cierra la ventana.
-

🎯 Objetivos de la Práctica

- ✅ **Acceder y modificar píxeles individuales en una imagen.**
 - ✅ **Seleccionar y editar regiones específicas de una imagen.**
 - ✅ **Copiar y mover partes de una imagen a otras posiciones.**
 - ✅ **Leer información clave sobre la imagen (tamaño, tipo de datos, dimensiones).**
-

🚀 Posibles Mejoras

- ♦ **Convertir la imagen en escala de grises antes de modificarla.**

```
python
CopiarEditar
img = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
```

- ◆ **Permitir al usuario seleccionar píxeles o regiones con el mouse.**
- ◆ **Guardar la imagen después de editarla.**

```
python
CopiarEditar
cv2.imwrite('imagen_modificada.jpg', img)
```

- ◆ **Mostrar la región copiada en otra ventana para comparación.**
-



Conclusión

- ✦ **Este código muestra cómo manipular imágenes directamente con OpenCV, modificando píxeles y regiones específicas.**
- ✦ **Es útil en aplicaciones de edición de imágenes, segmentación y reconocimiento de patrones.**
- ✦ **Se pueden agregar mejoras para hacer la edición interactiva o automatizada.**