

Rodrigo Hernández Zavala

15 de marzo de 2025

Ing. Mecatrónica

22310215

Operaciones bit a bit

Visión Artificial

🔗 Explicación del Código

Este código en **Python** usa **OpenCV** para **superponer un logo (img2) sobre otra imagen (img1)**, eliminando el fondo blanco del logo mediante **operaciones bit a bit**.

🔍 Paso a Paso del Código

📦 Importar Librerías

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

- ◆ **cv2 (OpenCV)** → Para procesamiento de imágenes.
 - ◆ **numpy (np)** → Para manipulación de matrices de píxeles.
-

📂 Cargar las Imágenes

```
python
CopiarEditar
img1 = cv2.imread('3D-Matplotlib.png')
img2 = cv2.imread('mainlogo.png')
```

- ◆ **img1** → Imagen de fondo donde se colocará el logo.
 - ◆ **img2** → Logo que se superpondrá en **img1**.
-

📏 Definir la Región de Interés (ROI)

```
python
CopiarEditar
rows, cols, channels = img2.shape
roi = img1[0:rows, 0:cols]
```

- ◆ `rows, cols, channels = img2.shape` → Obtiene dimensiones del logo (**img2**).
- ◆ `roi = img1[0:rows, 0:cols]` → **Extrae una región de img1** del mismo tamaño que **img2** para colocar el logo.

✅ **Ejemplo:** Si el logo mide **100x200 píxeles**, la región de interés (*roi*) será `img1[0:100, 0:200]`.

4 Convertir el Logo a Escala de Grises

```
python
CopiarEditar
img2gray = cv2.cvtColor(img2, cv2.COLOR_BGR2GRAY)
```

- ◆ Convierte `img2` en **escala de grises** para facilitar la creación de la máscara.
-

5 Crear una Máscara para Separar el Logo del Fondo

```
python
CopiarEditar
ret, mask = cv2.threshold(img2gray, 220, 255, cv2.THRESH_BINARY_INV)
```

◆ `cv2.threshold(img2gray, 220, 255, cv2.THRESH_BINARY_INV)` → Aplica **umbralización**:

- **Píxeles > 220** (blancos) → Se convierten en **negro (0)**.
- **Píxeles <= 220** (parte del logo) → Se convierten en **blanco (255)**.
 - ◆ **Inversión (THRESH_BINARY_INV)** → Para que el logo sea blanco y el fondo negro.

✅ **Ejemplo de `mask` (resultado de la umbralización):**

```
plaintext
CopiarEditar
Logo blanco sobre fondo negro.
```

6 Crear la Máscara Invertida

```
python
CopiarEditar
mask_inv = cv2.bitwise_not(mask)
```

- ◆ **Invierte la máscara** → Ahora el logo es negro y el fondo es blanco.

✅ **Ejemplo de `mask_inv`:**

```
plaintext
```

```
CopiarEditar
Fondo blanco, logo negro.
```

🗂 Crear el Fondo sin el Logo

```
python
CopiarEditar
img1_bg = cv2.bitwise_and(roi, roi, mask=mask_inv)
```

- ◆ **Elimina la parte del logo en la región de interés (roi).**
- ◆ `bitwise_and()` mantiene solo los píxeles donde la máscara es **blanca**.
- ◆ Como `mask_inv` tiene el **logo en negro**, el resultado es solo el **fondo sin el logo**.

✅ **Ejemplo de `img1_bg`:**

```
plaintext
CopiarEditar
Imagen de fondo con un espacio negro donde irá el logo.
```

🗂 Extraer Solo el Logo sin el Fondo

```
python
CopiarEditar
img2_fg = cv2.bitwise_and(img2, img2, mask=mask)
```

- ◆ `bitwise_and()` mantiene solo los píxeles del logo donde la **máscara es blanca**.
- ◆ **El fondo desaparece y queda solo el logo.**

✅ **Ejemplo de `img2_fg`:**

```
plaintext
CopiarEditar
Solo el logo, sin fondo.
```

🗂 Superponer el Logo sobre la Imagen

```
python
CopiarEditar
dst = cv2.add(img1_bg, img2_fg)
img1[0:rows, 0:cols] = dst
```

- ◆ `cv2.add(img1_bg, img2_fg)` → **Combina la imagen sin logo (`img1_bg`) con el logo (`img2_fg`).**
- ◆ `img1[0:rows, 0:cols] = dst` → Coloca el resultado en `img1`, en la misma posición.

✓ Resultado (img1):

```
plaintext
CopiarEditar
El logo se superpone sobre la imagen sin su fondo blanco.
```

10 Mostrar las Imágenes

```
python
CopiarEditar
cv2.imshow('res', img1)
cv2.imshow('mask_inv', mask_inv)
cv2.imshow('img1_bg', img1_bg)
cv2.imshow('img2_fg', img2_fg)
cv2.imshow('dst', dst)
```

◆ Muestra diferentes pasos del proceso:

- "res" → Imagen final con el logo insertado.
- "mask_inv" → Máscara invertida.
- "img1_bg" → Imagen de fondo sin logo.
- "img2_fg" → Solo el logo sin fondo.
- "dst" → Imagen fusionada antes de insertarse en img1.

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

◆ Espera a que el usuario presione una tecla y cierra todas las ventanas.

Objetivos de la Práctica

- ✓ Superponer imágenes con eliminación de fondo blanco.
 - ✓ Usar máscaras para separar objetos de su fondo.
 - ✓ Aplicar operaciones bit a bit (`bitwise_and`, `bitwise_not`).
 - ✓ Combinar imágenes sin perder información de fondo.
-

Posibles Mejoras

- ◆ Permitir mover el logo con coordenadas dinámicas.
- ◆ Ajustar el umbral (220) para adaptarse a diferentes fondos.
- ◆ Redimensionar el logo antes de insertarlo.

```
python
CopiarEditar
img2 = cv2.resize(img2, (100, 100)) # Ajusta tamaño
```

- ◆ **Guardar la imagen final:**

```
python
CopiarEditar
cv2.imwrite('resultado.png', img1)
```

Conclusión

- ✦ Este código demuestra cómo eliminar el fondo de una imagen y superponerla sobre otra usando OpenCV.
- ✦ Es útil para insertar logotipos, marcas de agua o realizar edición de imágenes.
- ✦ Puede mejorarse para soportar distintos colores de fondo o posiciones dinámicas.