

Rodrigo Hernández Zavala

15 de marzo de 2025

Ing. Mecatrónica

22310215

Escribir y Dibujar en imagenes

Visión Artificial

🔗 Explicación del Código

Este código en **Python** usa **OpenCV** para **dibujar formas geométricas y texto sobre una imagen**.

🔍 Paso a Paso del Código

📦 Importar Librerías

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

- ◆ **cv2 (OpenCV)** → Para trabajar con imágenes y dibujar formas.
 - ◆ **numpy (np)** → Para manejar coordenadas de polígonos.
-

📂 Cargar una Imagen en Color

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

- ◆ **cv2.imread('watch.jpg', cv2.IMREAD_COLOR)** → Carga la imagen "watch.jpg" en **color**.
- ◆ 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()
```

🖌 Dibujar una Línea

```
python
CopiarEditar
cv2.line(img, (0,0), (150,150), (255,255,255), 15)
```

- ◆ **Dibuja una línea blanca** desde (0,0) hasta (150,150).
 - ◆ **Grosor de la línea:** 15 px.
-

4 Dibujar un Rectángulo

```
python
CopiarEditar
cv2.rectangle(img, (15,25), (200,150), (0,255,0), 5)
```

- ◆ **Dibuja un rectángulo verde** desde la esquina superior izquierda (15,25) hasta (200,150).
 - ◆ **Borde de 5 px** de grosor.
-

5 Dibujar un Círculo

```
python
CopiarEditar
cv2.circle(img, (100,63), 55, (0,0,255), -1)
```

- ◆ **Dibuja un círculo rojo** con:
 - **Centro en (100,63).**
 - **Radio de 55 px.**
 - **-1** significa que el círculo estará relleno.
-

6 Dibujar un Polígono (Múltiples Líneas Conectadas)

```
python
CopiarEditar
pts = np.array([[10,5],[20,30],[70,20],[50,10]], np.int32)
#pts = pts.reshape((-1,1,2))
cv2.polylines(img, [pts], True, (0,255,255), 5)
```

- ◆ **Define una serie de puntos** [[10,5], [20,30], [70,20], [50,10]].
 - ◆ **cv2.polylines()** dibuja una **línea continua entre los puntos**.
 - ◆ **True** → Indica que el polígono está **cerrado** (se une el último punto con el primero).
 - ◆ **Color:** Amarillo (0,255,255), **Grosor:** 5 px.
-

📝 Escribir Texto sobre la Imagen

```
python
CopiarEditar
font = cv2.FONT_HERSHEY_SIMPLEX
cv2.putText(img, 'Tal cual Hermano', (0,130), font, 1, (200,255,155), 5,
cv2.LINE_AA)
```

- ◆ **Texto:** "Tal cual Hermano".
 - ◆ **Ubicación:** (0,130) → Coordenadas (esquina inferior izquierda del texto).
 - ◆ **Fuente:** cv2.FONT_HERSHEY_SIMPLEX (puedes probar otras fuentes).
 - ◆ **Tamaño:** 1.
 - ◆ **Color:** (200,255,155).
 - ◆ **Grosor:** 5 px.
 - ◆ **cv2.LINE_AA** → Suaviza los bordes del texto.
-

🖼️ Mostrar la Imagen con los Dibujos

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

- ◆ **Muestra la imagen en una ventana llamada "image".**
 - ◆ **cv2.waitKey(0)** → Espera hasta que el usuario presione una tecla para cerrar.
 - ◆ **cv2.destroyAllWindows()** → Cierra todas las ventanas abiertas por OpenCV.
-

🎯 Objetivos de la Práctica

- ✅ Aprender a **dibujar formas básicas** (líneas, rectángulos, círculos, polígonos) en OpenCV.
 - ✅ Comprender el uso de **coordenadas y colores** en imágenes.
 - ✅ Aplicar **texto sobre imágenes** con diferentes fuentes y tamaños.
-

🚀 Posibles Mejoras

- ◆ **Dibujar figuras con colores aleatorios:**

```
python
CopiarEditar
import random
color = (random.randint(0,255), random.randint(0,255),
random.randint(0,255))
cv2.line(img, (0,0), (150,150), color, 15)
```

- ◆ **Permitir interacción con el mouse para dibujar en tiempo real.**
- ◆ **Guardar la imagen modificada:**

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

Conclusión

- ✦ **Este código demuestra cómo agregar gráficos y texto a una imagen en OpenCV.**
- ✦ **Se puede modificar fácilmente para crear dibujos dinámicos o interactivos.**
- ✦ **Es una base útil para anotaciones en imágenes o generación de gráficos computacionales.**