Reporte de Práctica: Procesamiento de Imágenes con OpenCV y Matplotlib
Nombre: Eduardo Correa Flores
Materia: Visión Artificial
Tema: Visualización de imágenes y gráficos con Matplotlib
Lenguaje: Python

Objetivo:

Aprender a utilizar las librerías OpenCV y Matplotlib para:

1. **Cargar y visualizar imágenes en escala de grises.**
2. **Dibujar elementos gráficos (líneas) sobre imágenes.**
3. **Comparar las capacidades de visualización de OpenCV y Matplotlib.**

Descripción de la Práctica:

Se utilizó la imagen watch.jpg como base para realizar las siguientes operaciones:

1. Carga de la Imagen

- **Se cargó la imagen en escala de grises usando cv2.imread() con el flag cv2.IMREAD_GRAYSCALE, lo que convierte la imagen en una matriz 2D (sin canales RGB).**

python

Copy

Download

```
img = cv2.imread('watch.jpg', cv2.IMREAD_GRAYSCALE)
```

2. Visualización con OpenCV (Opcional)

- **El código incluía una sección comentada para mostrar la imagen con OpenCV:**

python

Copy

Download

 # cv2.imshow('image', img)

 # cv2.waitKey(0)

 # cv2.destroyAllWindows()

- **cv2.imshow(): Abre una ventana con la imagen.**
- **cv2.waitKey(0): Espera a que el usuario presione una tecla para cerrar.**
- **cv2.destroyAllWindows(): Cierra todas las ventanas de OpenCV.**

3. Visualización con Matplotlib

- **Se usó plt.imshow() para mostrar la imagen con:**
  - **Mapa de color: cmap='gray' (escala de grises).**
  - **Interpolación: interpolation='bicubic' (suavizado de imagen).**
  - **Ejes ocultos: plt.xticks([]), plt.yticks([]) para eliminar las marcas de los ejes.**

python

Copy

Download

```
plt.imshow(img, cmap='gray', interpolation='bicubic')
```

```
plt.xticks([]), plt.yticks([])
```

4. Dibujo de una Línea sobre la Imagen

- **Se agregó una línea cyan (turquesa) con plt.plot():**
  - **Coordenadas X: [50, 80, 40] (puntos en el eje horizontal).**
  - **Coordenadas Y: [70, 100, 50] (puntos en el eje vertical).**
  - **Estilo: Color 'c' (cyan), grosor linewidth=5.**

python

Copy

Download

```
plt.plot([50,80,40], [70,100,50], 'c', linewidth=5)
```

5. Visualización Final

- **La imagen modificada se mostró con:**

python

Copy

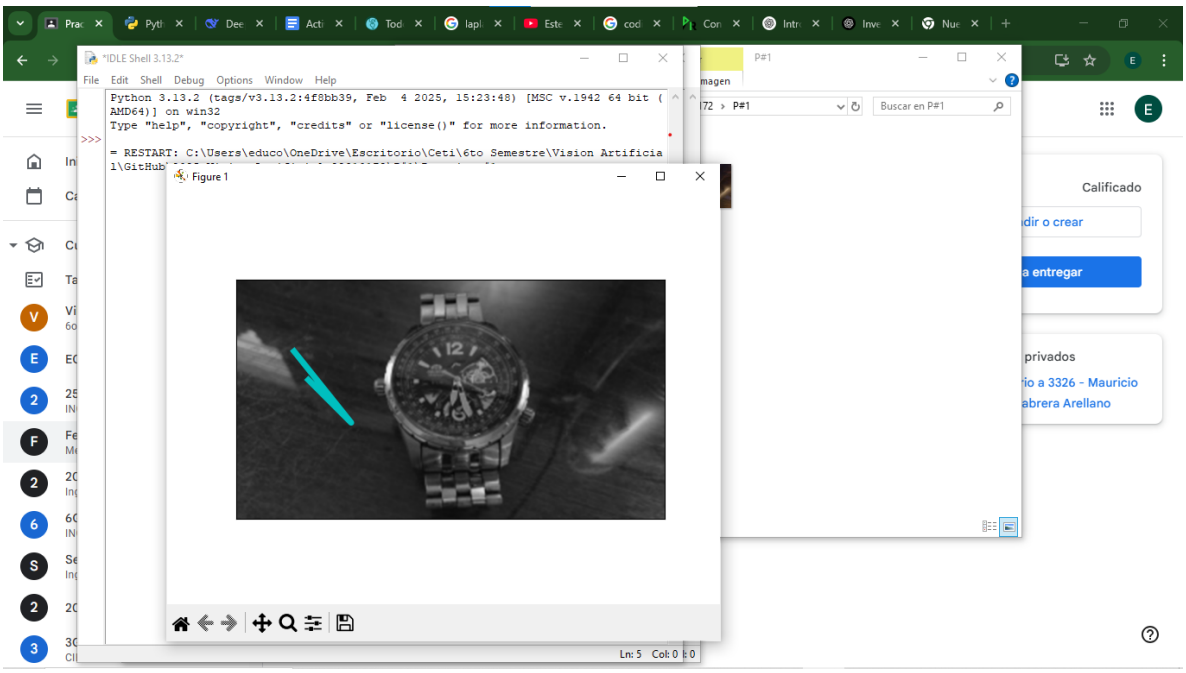
Download

plt.show()

Resultados Obtenidos:

1. **La imagen watch.jpg se cargó correctamente en escala de grises.**
2. **Se visualizó con Matplotlib, aplicando interpolación bicúbica para un mejor suavizado.**
3. **Se dibujó una línea cyan que conecta los puntos (50,70), (80,100) y (40,50).**
4. **Los ejes X e Y se ocultaron para una presentación más limpia.**

Ejemplo de salida:



Conclusión:

- **OpenCV es eficiente para cargar y procesar imágenes, mientras que Matplotlib ofrece mayor flexibilidad para visualizaciones personalizadas (como gráficos superpuestos).**
- **La interpolación bicubic mejora la calidad visual al renderizar la imagen.**
- **Dibujar elementos con Matplotlib es útil para anotaciones o resaltar características en análisis de imágenes.**