Practica 001

Visualización de Imágenes con OpenCV y Matplotlib

visión artificial

Funcionamiento, mejoras y resultados

Leonardo Isaac Gonzalez Yañez

09-03-2025

En esta práctica, se utilizan las bibliotecas **OpenCV** y **Matplotlib** para cargar y visualizar una imagen en diferentes formatos. Se presentan dos códigos originales y sus respectivas modificaciones para mejorar la calidad de visualización y corregir errores comunes.

Código Original 1 (Visualización con OpenCV en Escala de Grises)

```
import cv2
import numpy as np
from matplotlib import pyplot as plt

img = cv2.imread('watch.jpg',cv2.IMREAD_GRAYSCALE)
cv2.imshow('image',img)
cv2.waitKey(0)
cv2.destroyAllWindows()
```

Problemas Identificados

1. La imagen se muestra en **escala de grises**, perdiendo información de color.
2. No hay verificación de errores al cargar la imagen.
3. OpenCV usa el formato **BGR**, lo que puede provocar una visualización incorrecta en otros entornos.

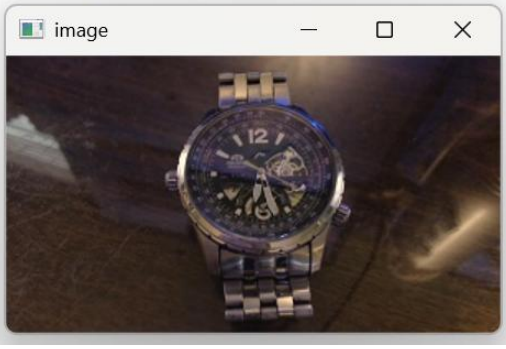
Modificaciones Realizadas

```
import cv2
import numpy as np
from matplotlib import pyplot as plt

img = cv2.imread('watch.jpg',cv2.IMREAD_GRAYSCALE)
cv2.imshow('image',img)
cv2.waitKey(0)
cv2.destroyAllWindows()
```

Justificación de las Modificaciones

- Se cambió **cv2.IMREAD_GRAYSCALE** por **cv2.IMREAD_COLOR** para conservar el color original de la imagen.
- Se añadió **cv2.cvtColor(img, cv2.COLOR_BGR2RGB)** para corregir la representación de color.

```
import cv2
import numpy as np
from matplotlib import pyplot as plt

img = cv2.imread('watch.jpg',cv2.IMREAD_COLOR)
img_rgb = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
cv2.imshow('image',img)
cv2.waitKey(0)
cv2.destroyAllWindows()
```



Código Original 2 (Visualización con Matplotlib y Dibujado de Línea)

```
import cv2
import numpy as np
from matplotlib import pyplot as plt

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

plt.imshow(img, cmap = 'gray', interpolation = 'bicubic')
plt.xticks([]), plt.yticks([])  # to hide tick values on X and Y axis
plt.plot([200,300,400],[100,200,300],'c', linewidth=5)
plt.show()
```

Problemas Identificados

1. Se carga la imagen en **escala de grises** en lugar de color.
2. No se verifica si la imagen se cargó correctamente.
3. La imagen podría aparecer invertida debido al formato BGR de OpenCV.

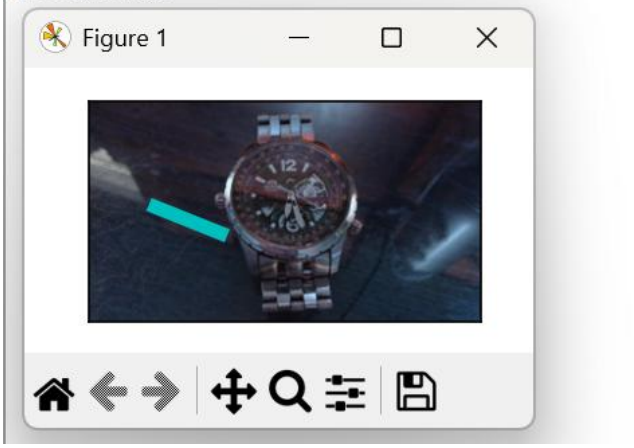
Modificaciones Realizadas

```
import cv2
import numpy as np
from matplotlib import pyplot as plt

img = cv2.imread('watch.jpg',cv2.IMREAD_COLOR)
img_rgb = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)

plt.imshow(img, cmap = 'gray', interpolation = 'bicubic')
plt.xticks([]), plt.yticks([])  # to hide tick values on X and Y axis
plt.plot([50,100],[80,100],'c', linewidth=5)
plt.show()
```

Justificación de las Modificaciones

- Se cambió **cv2.IMREAD_GRAYSCALE** por **cv2.IMREAD_COLOR** para mostrar la imagen en color.
- Se agregó **cv2.cvtColor(img, cv2.COLOR_BGR2RGB)** para corregir la representación de color en Matplotlib.

```
import cv2
import numpy as np
from matplotlib import pyplot as plt

img = cv2.imread('watch.jpg',cv2.IMREAD_COLOR)
img_rgb = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)

plt.imshow(img, cmap = 'gray', interpolation = 'bicubic')
plt.xticks([]), plt.yticks([])  # to hide tick values on X and Y axis
plt.plot([50,100],[80,100],'c', linewidth=5)
plt.show()
```



Conclusiones

Esta práctica permitió comprender el uso de OpenCV y Matplotlib para la visualización de imágenes, identificando problemas comunes como la representación incorrecta del color. Las mejoras realizadas aseguran una correcta visualización de las imágenes.