

Práctica: Histograma y Ecualización del Histograma

Objetivo:

Analizar y aplicar la ecualización del histograma sobre una imagen en escala de grises, comparando visualmente la imagen original, su histograma, la imagen ecualizada y el histograma resultante.

Material Utilizado:

- Lenguaje de programación: Python 3
- Librerías: OpenCV, NumPy, Matplotlib
- Imagen: Reloj (watch.jpg)

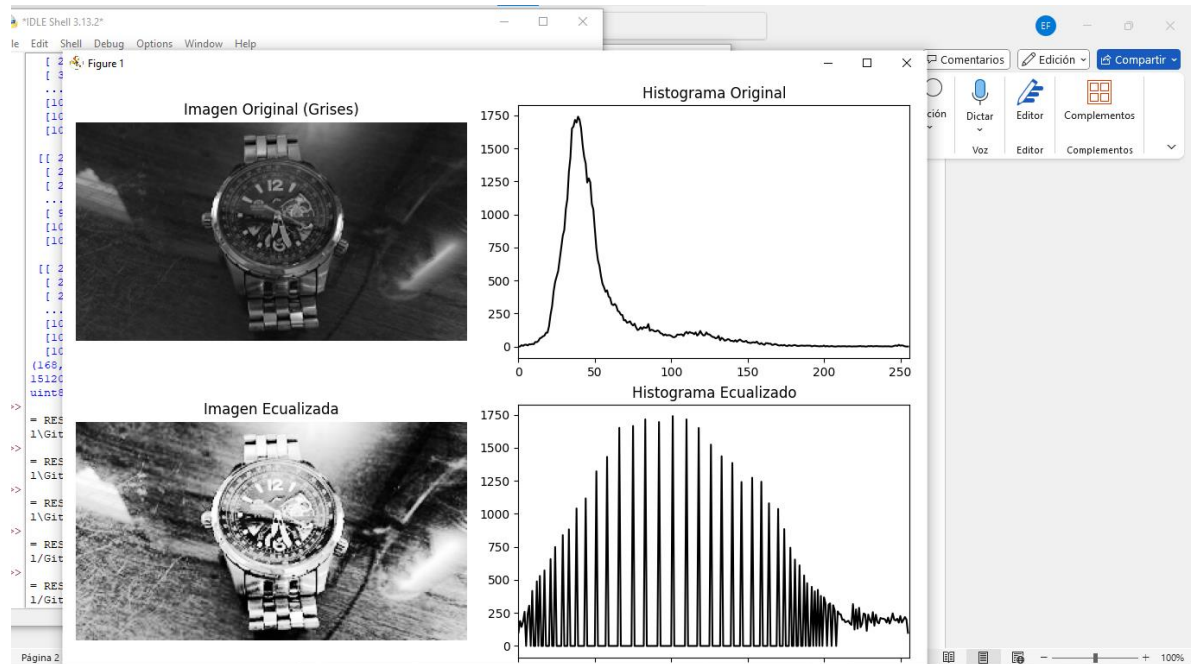
Desarrollo:

1. Se cargó una imagen en formato color utilizando la función `cv2.imread()`.
2. La imagen fue convertida a escala de grises con `cv2.cvtColor()`, ya que los histogramas de grises permiten un análisis más claro de la distribución de intensidades.
3. Se aplicó la función `cv2.equalizeHist()` para realizar la ecualización del histograma, proceso que mejora el contraste al redistribuir las intensidades de los píxeles.
4. Se calcularon los histogramas antes y después de la ecualización mediante `cv2.calcHist()`.
5. Se visualizó todo en una sola ventana utilizando `matplotlib.pyplot`:
 - Imagen original en escala de grises.
 - Histograma de la imagen original.
 - Imagen ecualizada.
 - Histograma de la imagen ecualizada.

Resultados:

- La imagen original presenta una distribución de intensidades concentrada en un rango estrecho, lo que genera bajo contraste.
- Tras aplicar la ecualización, la imagen muestra un contraste mejorado y un histograma distribuido de manera más uniforme.

- La visualización conjunta permite comprender el efecto directo de la ecualización sobre los niveles de intensidad y su impacto en la calidad visual de la imagen.



Conclusiones:

- La ecualización del histograma es una herramienta efectiva para mejorar el contraste en imágenes con iluminación desigual.
- El uso de matplotlib facilita la comparación visual al permitir mostrar imágenes y gráficas en una misma ventana.
- Esta práctica demuestra la importancia del preprocesamiento de imágenes en tareas de visión artificial.

Fragmento de Código Empleado:

```
img_color = cv2.imread('watch.jpg')
img_gray = cv2.cvtColor(img_color, cv2.COLOR_BGR2GRAY)
img_eq = cv2.equalizeHist(img_gray)
hist_orig = cv2.calcHist([img_gray], [0], None, [256], [0, 256])
hist_eq = cv2.calcHist([img_eq], [0], None, [256], [0, 256])
```

Visualización con Matplotlib:

- Se utilizaron `plt.imshow()` para mostrar las imágenes.
- Se graficaron los histogramas con `plt.plot()`.
- Las subventanas fueron organizadas con `plt.subplot()` y mostradas con `plt.show()`.