



ING. Mecatrónica

visión Artificial

Saul Isaac Limon Bautista

22310278

## Documento explicativo: Histograma y Ecuación de Histograma en Imágenes

Este documento explica paso a paso el funcionamiento del código que realiza el procesamiento de una imagen en escala de grises, su ecualización de histograma y la visualización de los resultados usando matplotlib.

---

### 1. Importación de librerías

```
import cv2
```

```
import numpy as np
```

```
from matplotlib import pyplot as plt
```

Se importan tres librerías esenciales:

- cv2 para el procesamiento de imágenes (OpenCV).
  - numpy para operaciones numéricas.
  - matplotlib.pyplot para graficar imágenes y sus histogramas.
- 

### 2. Carga y conversión de imagen a escala de grises

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

- Se carga la imagen watch.jpg en escala de grises.
  - El modo cv2.IMREAD\_GRAYSCALE convierte la imagen a un solo canal (intensidad de grises).
- 

### 3. Validación de carga de imagen

```
if img is None:
```

```
    print("Error al cargar la imagen.")
```

```
    exit()
```

- Se verifica que la imagen haya sido cargada correctamente. Si no, se termina la ejecución.
-

#### 4. Ecualización del histograma

```
img_eq = cv2.equalizeHist(img)
```

- `cv2.equalizeHist()` mejora el contraste de la imagen al redistribuir la intensidad de los píxeles.
  - La salida `img_eq` es la imagen con mejor contraste.
- 

#### 5. Cálculo de histogramas

```
hist_orig = cv2.calcHist([img], [0], None, [256], [0, 256])
```

```
hist_eq = cv2.calcHist([img_eq], [0], None, [256], [0, 256])
```

- `cv2.calcHist` genera un histograma con 256 niveles (0-255) para la imagen original y la ecualizada.
  - El histograma muestra la frecuencia de cada nivel de intensidad.
- 

#### 6. Configuración de la visualización con matplotlib

```
plt.figure(figsize=(10, 6))
```

- Se crea una figura de 10x6 pulgadas para organizar los subgráficos.
- 

#### 7. Visualización de la imagen original

```
plt.subplot(2, 2, 1)
```

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

```
plt.title('Imagen Original')
```

```
plt.axis('off')
```

- Se muestra la imagen original en una grilla 2x2, posición 1.
- 

#### 8. Visualización del histograma original

```
plt.subplot(2, 2, 2)
```

```
plt.plot(hist_orig, color='black')
```

```
plt.title('Histograma Original')
```

```
plt.xlim([0, 256])
```

- Se grafica el histograma original en la posición 2.
- 

## 9. Imagen ecualizada

```
plt.subplot(2, 2, 3)
```

```
plt.imshow(img_eq, cmap='gray')
```

```
plt.title('Imagen Ecualizada')
```

```
plt.axis('off')
```

- Se muestra la imagen con contraste mejorado en la posición 3.
- 

## 10. Histograma ecualizado

```
plt.subplot(2, 2, 4)
```

```
plt.plot(hist_eq, color='black')
```

```
plt.title('Histograma Ecualizado')
```

```
plt.xlim([0, 256])
```

- Se grafica el histograma de la imagen ecualizada.
- 

## 11. Mostrar la ventana con todos los elementos

```
plt.tight_layout()
```

```
plt.show()
```

- `tight_layout()` ajusta automáticamente los espacios entre gráficos.
  - `plt.show()` muestra la ventana con todos los elementos.
- 

## Resultado:

Una ventana que contiene:

- Imagen original.
- Histograma original.
- Imagen ecualizada.
- Histograma ecualizado.

Esto permite visualizar claramente el efecto de la ecualización del histograma sobre el contraste de la imagen.

Resultado:

