



## ***Visión Artificial***

***No. de Practica: 3***

***Título: Histograma y ecualizado del histograma***

***Nombre: Aldo Misael Osuna Rodríguez***

***Registro: 22310221***

***6°G***

***15-Mayo-2025***

## Objetivo:

Histograma y ecualizado del histograma de las imágenes de la práctica 2.

## Código:

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

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

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

```
img = cv2.imread('3D-Matplotlib.png', cv2.IMREAD_GRAYSCALE)
```

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

```
histx1 = cv2.calcHist([img], [0], None, [256], [0, 256])
histx2 = cv2.calcHist([img_eq], [0], None, [256], [0, 256])
```

```
histy1 = np.sum(img, axis=0)
histy2 = np.sum(img_eq, axis=0)
```

```
fig = plt.figure(figsize=(14, 6))
gs = fig.add_gridspec(3, 4)
```

```
ax1 = fig.add_subplot(gs[0, 0])
ax1.plot(histx1, color='black')
ax1.set_title('Histograma X OG')
ax1.set_xlim([0, 256])
ax1.set_xticks([])
ax1.set_yticks([])
```

```
ax2 = fig.add_subplot(gs[0, 2])
ax2.plot(histx2, color='black')
ax2.set_title('Histograma X Ecualizada')
ax2.set_xlim([0, 256])
ax2.set_xticks([])
ax2.set_yticks([])
```

```
ax3 = fig.add_subplot(gs[1, 0])
ax3.plot(histy1, color='black')
ax3.set_title('Histograma Y OG')
ax3.set_ylim([0, np.max(histy1)])
ax3.set_yticks([])
ax3.set_xticks([])
```

```
ax4 = fig.add_subplot(gs[1, 1])
ax4.imshow(img, cmap='gray')
ax4.set_title('Original')
ax4.axis('off')
```

```
ax5 = fig.add_subplot(gs[1, 2])
ax5.plot(histy2, color='black')
ax5.set_title('Histograma Y Ecualizada')
ax5.set_ylim([0, np.max(histy2)])
ax5.set_yticks([])
ax5.set_xticks([])
```

```
ax6 = fig.add_subplot(gs[1, 3])
ax6.imshow(img_eq, cmap='gray')
ax6.set_title('Result')
ax6.axis('off')
```

```
plt.tight_layout()
plt.show()
```

### Comentarios:

OpenCv nos proporciona una función a través de la cuál podemos simplemente aplicar ecualización de histograma sobre una imagen, llamada `equilizeHist()`. Es sencilla de aplicar a una imagen en escala de grises ya que el método realmente ecualiza el histograma de una imagen en escala de grises, pero en nuestro caso

tenemos tres canales (RGB) para cada pixel y no podemos simplemente aplicar ecualización de histograma sobre tres canales de manera separada.

Una buena solución con la que me encontré en el libro Python: Real World Machine Learning es convertir nuestra imagen al espacio de color YUV, ecualizar el canal Y, y finalmente convertir el resultado a RGB. Así que la primera cosa que hacemos es convertir nuestra imagen a YUV. Esto puede hacerse usando el método `cvtColor()`, el cuál convierte la imagen de un espacio de color a otro:

```
"img_to_yuv = cv2.cvtColor(img,cv2.COLOR_BGR2YUV)"
```

Nota que usamos BGR en lugar de RGB aquí, ya que OpenCV carga las imágenes en formato BGR.

Ahora aplicamos el método de ecualización de histograma sobre el canal Y usando el método `equalizeHist()`:

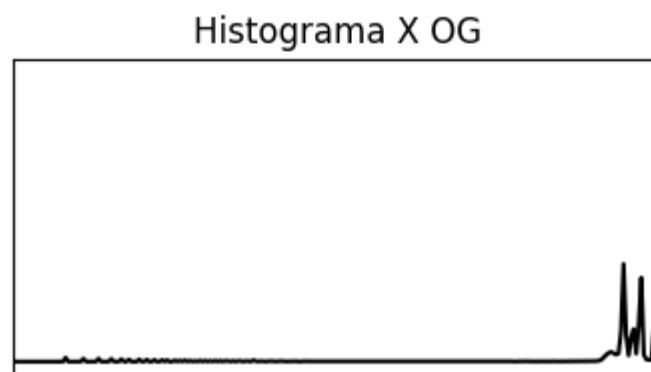
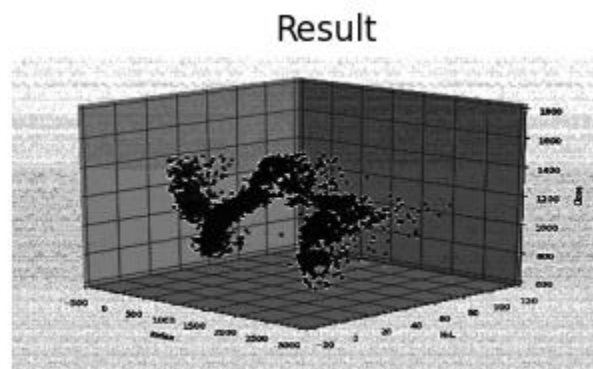
```
"img_to_yuv[:, :, 0] = cv2.equalizeHist(img_to_yuv[:, :, 0])"
```

Finalmente, convertimos el canal Y a RGB (BGR en OpenCV), como sigue:

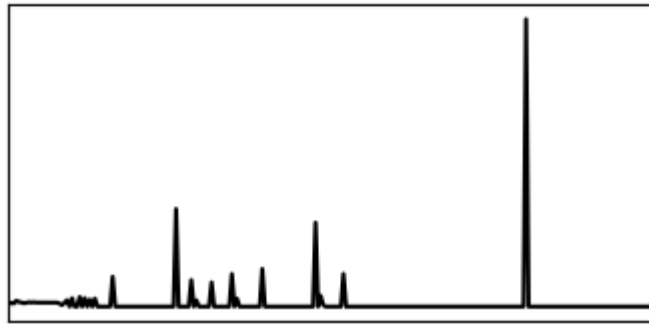
```
"hist_equalization_result = cv2.cvtColor(img_to_yuv, cv2.COLOR_YUV2BGR)"
```

Y listo hemos aplicado ecualización de histograma a la Imagen.

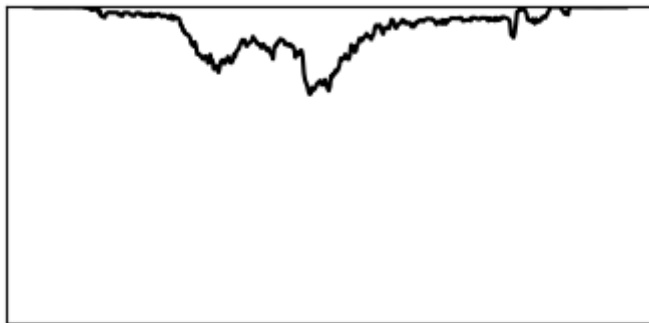
## Resultados:



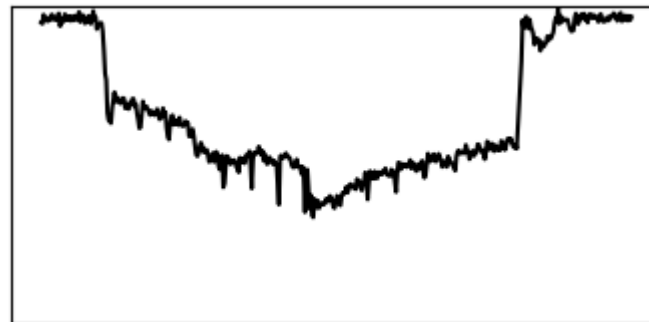
Histograma X Ecualizada



Histograma Y OG



Histograma Y Ecualizada



## Bibliografía

Ali, A.-R. (30 de Enero de 2018). *envatotuts+*. Obtenido de envatotuts+:

<https://code.tutsplus.com/es/histogram-equalization-in-python--cms-30202t>

Vélez, J. F. (2003). *Visión por computador*. Ana Belén Moreno Díaz,. Obtenido de

[https://www.researchgate.net/publication/31759521\\_Vision\\_por\\_computador\\_JF\\_Vel ez\\_Serrano\\_et\\_al](https://www.researchgate.net/publication/31759521_Vision_por_computador_JF_Vel ez_Serrano_et_al)