

**Universidad Nacional de San Antonio Abad del Cusco**  
**Departamento Académico de Ing. Informática**  
**VISION COMPUTACIONAL**  
**Práctica N° 04**

**HISTOGRAMAS**

Iván C. Medrano Valencia

**1. OBJETIVO.**

- Obtener el histograma de una imagen en escala de grises o a color.
- Conocer la utilidad del histograma de una imagen.

**2. ACTIVIDAD I. Obtener el histograma de una imagen utilizando matplotlib.pyplot.hist**

**Ejercicio 4.1.** En este ejercicio, leemos una imagen en escala de grises, mostrándola y luego trazando su histograma 2D con 256 bins.

```
import cv2
import matplotlib.pyplot as plt
#--leer la imagen
img= cv2.imread('images\sunflower.jpg', cv2.IMREAD_GRAYSCALE)
cv2.imshow('grayscale image', img)
#--calcular histograma y mostrar
#--El eje X del histograma de la imagen tendrá un rango de 0-255
#--y el eje Y rastrea el recuento de píxeles contra cada intensidad.
ax = plt.hist(img.ravel(), bins = 256)
plt.show()
#--esperar
cv2.waitKey(0)
cv2.destroyAllWindows()
```

**Ejercicio 4.2. Trazar una imagen de muestra y su histograma con 10 bins.**

```
import cv2
import matplotlib.pyplot as plt
#--leer la imagen
img= cv2.imread("images\sunflower.jpg",cv2.IMREAD_GRAYSCALE)
cv2.imshow('grayscale image', img)
#--calcular el histograma, por defecto 10 bins
plt.hist(img.ravel())
plt.show()
#--esperar
cv2.waitKey(0)
cv2.destroyAllWindows()
```

### Ejercicio 4.3. Histograma de una imagen a color.

Para trazar los tres histogramas 2D de los tres canales (B, G y R) en un solo trazado pero en diferentes colores, se puede llamar al comando `plt.show()` después de usar el comando `plt.hist()` en cada canal individual. Se usa un valor de transparencia de 0.5 para mostrar los tres canales en sus respectivos colores en el gráfico del histograma:

```
import cv2
import matplotlib.pyplot as plt
#--leer la imagen
img= cv2.imread('../images/sunflower.jpg')
cv2.imshow("color image",img)

im= img[:, :, 2]
plt.hist(im.ravel(), bins = 256, color = 'Red', alpha = 0.5)
im= img[:, :, 1]
plt.hist(im.ravel(), bins = 256, color = 'Green', alpha = 0.5)
im= img[:, :, 0]
plt.hist(im.ravel(), bins = 256, color = 'Blue', alpha = 0.5)
plt.show()
#--esperar
cv2.waitKey(0)
cv2.destroyAllWindows()
```

### 3. ACTIVIDAD II. HISTOGRAMAS CON OpenCV.

Para obtener el histograma de una imagen con OpenCV utilizaremos la siguiente instrucción:

```
cv2.calcHist([images], [channels], mask, [histSize],[ranges])
```

en la que:

**[images]** : Esta es la lista que contiene sus imágenes de entrada. Puede ser 2D o 3D.

**[channels]** : Éste es el índice del plano de la imagen cuyo histograma quiere calcular. Para una imagen en escala de grises o binaria, solo puede ser 0. También se proporciona como una lista, [], de modo que para una imagen BGR, puede proporcionar el índice de canal individual, como en [0], [1] o [2], o al calcular histogramas de varios planos juntos, puede proporcionar valores en consecuencia: por ejemplo, [0, 1] para los planos B y G combinados o [0, 1, 2] para los tres planos combinados.

**mask**: Aquí podemos proporcionar una máscara. Si se proporciona una máscara, se calculará un histograma solo para píxeles enmascarados. Si no tenemos una máscara o no queremos aplicar una, solo podemos proporcionar un valor de **None**.

**[histSize]**: Este es el número de contenedores (bins) que desea utilizar. Si tiene un solo canal, esta lista tiene un solo número (por ejemplo, [10]). Al calcular un histograma de dos canales, esta lista tendrá dos números (por ejemplo, [10,10]), y al calcular un histograma de tres canales juntos, esta lista tendrá tres números (por ejemplo, [10,10,10]).

**[ranges]** : Éste es el rango de límites de intervalo de histograma en cada dimensión. También se presenta como una lista, []. Para un solo canal, puede dar un valor único como [256], pero para más de un canal juntos, debe proporcionar rangos completos para cada canal (por ejemplo, [0, 255, 0, 255] para dos canales y [0, 255, 0, 255, 0, 255] para tres canales).

#### Ejercicio 4.4. Histograma de una imagen en escala de grises

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

#--leer la imagen
img = cv2.imread('../images/lena.jpg',cv2.IMREAD_GRAYSCALE)
cv2.imshow('lena.jpg', img)
#--determinar el histograma
hist = cv2.calcHist([img], [0], None, [256], [0, 256])
#--plotear el histograma
plt.plot(hist, color='gray' )
plt.xlabel('intensidad de iluminacion')
plt.ylabel('cantidad de pixeles')
plt.show()

cv2.destroyAllWindows()
```

- **Ejercicio 4.5. Histograma de una imagen a color.**

```
import cv2
import numpy as np
from matplotlib import pyplot as plt
#--leer la imagen
img = cv2.imread('../images/lena.jpg')
cv2.imshow('lena.jpg', img)
color = ('b','g','r')
#--calcular el histograma
for i,col in enumerate(color):
    histr = cv2.calcHist([img],[i],None,[256],[0,256])
    plt.plot(histr,color = col)
    plt.xlim([0,256])
plt.show()
```

- **Ejercicio 4.6. Histograma con máscara.**

```
import cv2
import numpy as np
from matplotlib import pyplot as plt
img = cv2.imread('../images/lena.jpg',0)

# crear máscara
mask = np.zeros(img.shape[:2], np.uint8)
mask[10:140, 100:200] = 255
```

```

masked_img = cv2.bitwise_and(img,img,mask = mask)

# Calcular el histograma con máscara y sin máscara
hist_full = cv2.calcHist([img],[0],None,[256],[0,256])
hist_mask = cv2.calcHist([img],[0],mask,[256],[0,256])

plt.subplot(221), plt.imshow(img, 'gray')
plt.subplot(222), plt.imshow(mask, 'gray')
plt.subplot(223), plt.imshow(masked_img, 'gray')
plt.subplot(224), plt.plot(hist_full), plt.plot(hist_mask)
plt.xlim([0,256])

plt.show()

```

#### 4. ACTIVIDAD III. ECUALIZACION DE UN HISTOGRAMA

Para mejorar el contraste de una imagen modificando su histograma, se debe asegurar de que el histograma no se concentre alrededor de ninguna intensidad en particular. Esto se logra extendiendo el gráfico del histograma en todos los valores de intensidad de 0 a 255; esto le dará una imagen de alto contraste y se llama ecualización de histograma. La biblioteca OpenCV de Python nos proporciona un comando muy útil que nos permite lograr esto en una sola línea:

```
imgOut = cv2.equalizeHist (imgIn)
```

Aquí, `imgIn` es la imagen de entrada en escala de grises cuyo histograma desea ecualizar, e `imgOut` es la imagen de salida (histograma ecualizado).

##### Ejercicio 4.7. Ecualización de histograma de una imagen en escala de grises

En este ejercicio, leemos una imagen en escala de grises, mostrándola y luego se traza su histograma 2D con 256 bins. Después de eso, le aplicaremos la ecualización de histograma y luego trazaremos el histograma de la imagen ecualizada.

```

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

#--leer la imagen
img= cv2.imread('../images/dark_image1.png', 0)
cv2.imshow('Imagen original', img)

#--obtener el histograma de la imagen original
ax = plt.hist(img.ravel(), bins= 256)
plt.title('Histograma de la imagen original')
plt.show()

#--aplicando la ecualización
histequ = cv2.equalizeHist(img)
#--muestra la imagen ecualizada
cv2.imshow('Histograma de la imagen ecualizada', histequ)
#--obtiene el histograma de la imagen ecualizada

```

```

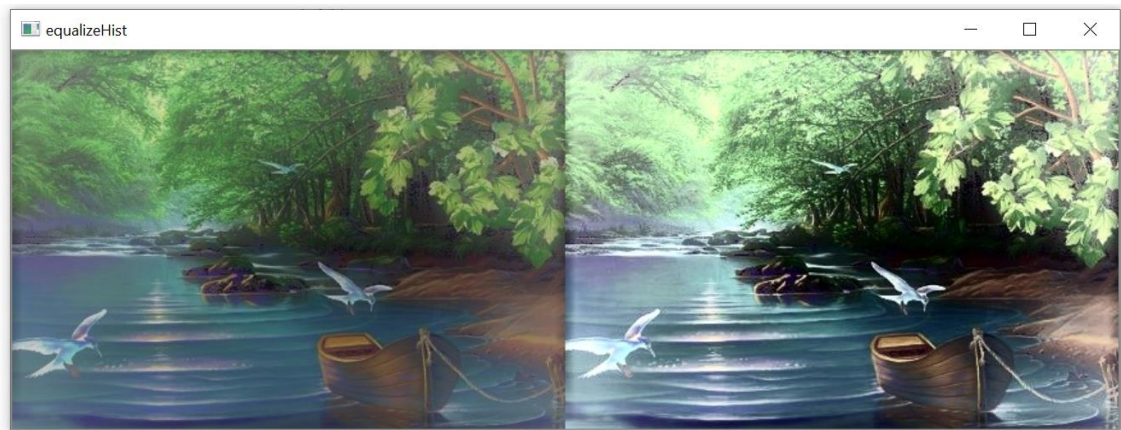
ax = plt.hist(histequ.ravel(), bins= 256)
plt.title('Histogram of Histogram Equalized Image')
plt.show()

img_with_histequ = np.hstack((img,histequ))
cv2.imshow('Comparison', img_with_histequ)
cv2.waitKey(0)
cv2.destroyAllWindows()

```

## 5. TAREA,

Ecualizar una imagen a colores como en el ejemplo:



## 6. REFERENCIAS BIBLIOGRÁFICAS

- Dawson-Howe, K. (2014). A Practical Introduction to Computer Vision With OpenCV, Wiley & Sons Ltd.
- Hafsa Asad, W. R., Nikhil Singh (2020). The Computer Vision Workshop. Birmingham U.K., Pack Publishing.
- Szeliski, R. (2011). Computer Vision Algorithms and Applications. London, Springer.