Intente resolver la mayoría de los ejercicios a continuación, a través de realizar recorridos por los píxeles en vez de usar comandos pre fabricados que hagan toda la tarea por si mismos.

Procesamiento de Imagenes: Segmentación y Filtros

Histogramas

1. Diseñar una función que genere el histograma de una imagen en escala de grises, que se le dé como argumento. Graficar el histograma de alguna imagen cualquiera, que haya convertido previamente a escala de grises, utilizando la librería matplotlib. Puede optar por implementarla como un *procedimiento* que directamente realice la gráfica del mismo, al ser llamada. O podría ser una función cuyo retorno sea el histograma, y luego en el programa principal realizar la gráfica. Lo que crea más conveniente para su uso actual.

   Notar que en este caso, estaremos haciendo la versión más simple de histograma, que es efectivamente la frecuencia de todos los valores. *(No se hará una agrupación por rangos de valores o bins)*

2. Diseñar una función que genere el histograma de un solo color, el que se elija, dada una imagen color pasada como argumento.

3. Investigar la sintaxis de las funciones de OpenCV, Numpy y Matplotlib para generar histogramas. Listar la sintáxis de cada una.

Thresholding

4. Implementar una función llamada *bn* que genere, a partir de una imagen y un valor llamado *umbral* una versión en blanco y negro de la misma. El valor de retorno de la función, será la nueva imagen. Recuerde que la función de threshold, la más básica sería aquella que para un valor dado, si supera un umbral, devuelve el valor máximo, sino 0:
   ** deberá adaptarla a imagenes.*

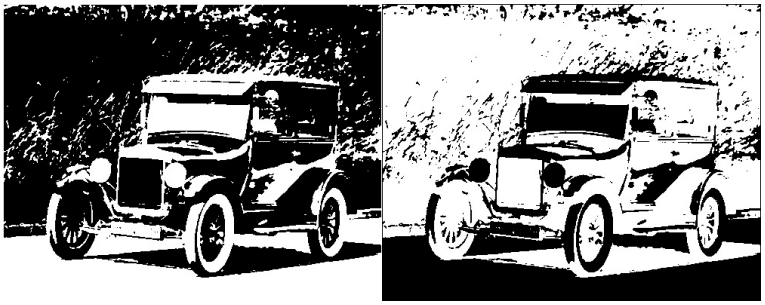
$$f(x) = \begin{cases} maxvalor & \text{si} \quad x > umbral \\ 0 & \text{si} \quad sino \end{cases}$$

5. Dada la imagen llamada *ciudad.jpg* utilice el histograma, para analizar cual sería un buen valor de umbral para separar los edificios del cielo.

6. Dada la imagen llamada *actriz.png* genere una versión binaria de la misma. Refine su versión analizando el histograma de la misma.

7. Con la imagen binara generada en el punto anterior, utilicela como una máscara para recortar la actriz del fondo, y generar una imagen nueva, que tenga la actriz sobre un fondo anaranjado por ejemplo. Podrá observar fácilmente, el desafío presente en separar figura de fondo.

8. Implementar una función que genere una imagen binaria invertida, dada una imagen en escala de grises y un valor de umbral, pasados como argumentos. Por ejemplo, para la imagen $fordt.png$ se podría apreciar la versión binaria estándar a la izquierda, y la versión binaria invertida a la derecha.



9. Dada la imagen original *actor.png* en color, generar la siguiente imagen, ¿Qué pasos debería hacer para lograrlo?

Convoluciones

10. Implementar un programa que dada una imagen en escala de grises, genere una imagen suavizada *(Blurring)*. Realice dos versiones, la primera con el filtro conocido como Box Filter de la figura 1 *(promedio de los pixeles vecinos)* y la segunda versión con el filtro Gaussiano de la figura 2. Compare sus resultados.

$$\frac{1}{9}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

Figura 1: Box Filter de 3x3, normalizado.

$$\frac{1}{16}\begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$$

Figura 2: Filtro Gaussiano de 3x3, normalizado.

11. Implementar un programa que dada una imagen en escala de grises *(o en color, pero convertida a grises)* aumente su nitidez, utilizando alguno de los kernel de Laplace dados a continuación. *(Sharpen)*

0	-1	0
-1	4	-1
0	-1	0

0	-1	0
-1	5	-1
0	-1	0

12. ¿En qué cree que puede consistir darle relieve a una imagen? Piense. Diseñe un kernel de 3x3 que considere debiera lograr un efecto de relieve en la imagen. Recuerde que los valores del kernel representan cuanta importancia se le dará o quitará a cada pixel vecino, y al pixel en cuestión del medio que se está calculando para la nueva imagen. Implemente un programa con dicho kernel y una imagen en escala de grises que usted elija. Experimente y muestre al profesor sus resultados :)

13. Implemente ahora un programa que aplique el filtro Gaussiano de 5x5. ¿Encuentra diferencias con la versión de 3x3?

1/273

1	4	7	4	1
4	16	26	16	4
7	26	41	26	7
4	16	26	16	4
1	4	7	4	1

Figura 3: Filtro Gaussiano de 5x5, normalizado.

14. Implemente un programa que ejecute el filtro Gaussiano de 3x3 de suavizado en una imagen color de 3 canales. ¿Qué piensa debería hacer para adaptar el algoritmo con imagenes de escala de grises, a una imagen color?

15. Implemente un programa que permita ingresar una imagen, y detecte los bordes de la misma, realice las siguientes versiones:

    a) Detección de Bordes de Laplace.

    b) Detección de Bordes de Sobel, sin suavizar.

    c) Detección de Bordes de Sobel, con previo suavizado.

    Un kernel de Laplace para detectar bordes es:

-1	-1	-1
-1	8	-1
-1	-1	-1

    y los de Sobel son:

1	2	1
0	0	0
-1	-2	-1

-1	0	1
-2	0	2
-1	0	1

Variado

16. Implementar una función llamada *mascara* que tome 3 argumentos: una imagen, y dos tuplas BGR que representarán un rango de color, y que genere una imagen en blanco y negro, de mismo tamaño que la original, tal que si el pixel en la imagen original se encuentra en el rango de color dado, la imagen mascara tendrá el color blanco en ese píxel, y sino tendrá el color negro en ese píxel.

17. Crear una función *rotar* que tome dos argumentos: una imagen color y un ángulo de rotación. Devolverá una nueva imagen, que será la original rotada dicho ángulo. El ángulo solo podrá tomar los valores 90, 180, y 270. Generar una versión artesanal.

18. Crear una función llamada *flip* que tome dos parámetros: una imagen y un eje *(eje será una variable del tipo str, que contendrá uno de dos valores posibles, horizontal o vertical)*. Devolver una nueva imagen que sea la versión espejada de la imagen original, sobre el eje vertical o sobre el eje horizontal según corresponda.

Referencias

[1] Mark Newman, *Convolution and the Fourier Transform explained visually.* Mark Newman (2022) *(Video)*.

[2] 3Blue1Brown, *But what is a convolution?* (2022) *(Video)*.

[3] Prathima Guruprasad, *Overview of different Thresholding Methods in image processing* (2020) *(Artículo)*.