Procesamiento digital de imágenes **Octubre, 2023-2**

Tarea 2: Filtros de convolución y detección de bordes.

Profesora: Pamela Guevara *Ayudantes: Martín López, Paula Inostroza*

Instrucciones generales

Escribir en Python (debe indicar versión usada) el desarrollo de los ejercicios enunciados a continuación, de forma ordenada y lo más eficiente posible, con las explicaciones y demostraciones pertinentes (comentarios en líneas de código relevantes y despliegue de los resultados obtenidos. Cada imagen e histograma debe incluir su respectivo título). Puede usar las bibliotecas de procesamiento de imágenes vistas en clases, disponibles en Python (PILLOW,OpenCV, etc).

La tarea tiene un valor de 60 puntos, es en grupos de máximo 2 personas, y debe subir a CANVAS:

- El código y todos los archivos necesarios para su ejecución en un archivo comprimido con el formato: apellido-integrante1_apellido integrante2_tarea2.zip. Este archivo debe contener sólo un archivo .py o .ipynb con el código y las imágenes necesarias para compilar su código.
- Indicar en la parte superior del código, un comentario con el nombre de los integrantes.

Las CONSULTAS sobre la tarea deberán ser realizadas por TEAMS, en el horario de Lunes a Viernes de las 8:00hrs hasta las 19:00hrs. Se recibirán consultas hasta el **Viernes 10 de Noviembre**

Fecha de entrega: **Lunes 13 de Noviembre**

Nota: Tanto la profesora como los ayudantes se reservan el derecho de realizar interrogaciones para comprobar la autenticidad del trabajo entregado. Se podrán interrogar a ambos integrantes para corroborar la participación de ambos en el desarrollo de las tareas. De haber discrepancia entre los conocimientos de ambos integrantes y con la nota del trabajo, la nota de la tarea se promediará con la nota de la interrogación de cada integrante. Trabajos muy similares de distintos grupos serán calificados con la nota mínima. En caso de atraso, se descontarán 10 puntos por día de atraso.

Imágenes:

Las imágenes con las cuales se pide trabajar en el desarrollo de la tarea se encuentran disponibles en el siguiente link:

https://drive.google.com/drive/folders/1z95R6ptQMo_jYQzMexEOWnLwcFW1Bclu?usp=share_link

Ejercicio 1

Para la imagen “numbers.jpg”, realice los siguientes procesamientos.

(a) Cargue la imagen y obtenga una imagen con un efecto de espejo de los tres primeros números, como se muestra en la Figura 1. Para esto puede apoyarse en el uso de transformaciones geométricas. Despliegue la imagen espejo y la original en una sola figura.

(b) Genere y despliegue la imagen original, con el link que se encuentra en la esquina inferior derecha ilegible (suavizado), como se muestra en la Figura 2. Para esto aplique el filtro de media con una ventana que permita obtener el resultado deseado.

(c) Convierta la imagen original a escala de grises para aplicar el método multi-otsu de 3 clases. Muestre en una sola figura la imagen original, la imagen en escala de grises y el resultado de aplicar multi-otsu. Explique la distribución de intensidades del resultado, para esto apóyese en el concepto del método multi-otsu y del histograma.

(d) Aplique a la imagen en escala de grises detección de bordes con el filtro sobel. Obtenga solo los bordes externos de los números, como se muestra en la Figura 3. Explique el procedimiento de este filtro.

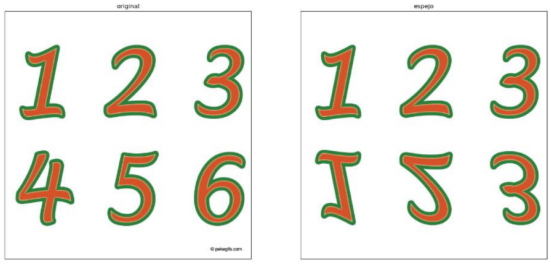

Figura 1: Imagen números original y espejo.

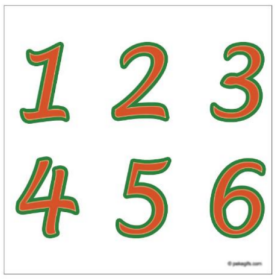

Figura 2: Imagen números con marca de agua suavizada.

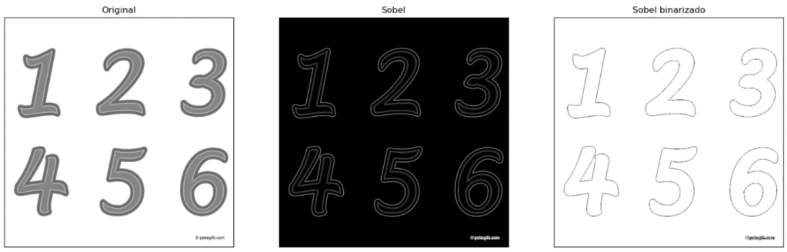

Figura 3: Aplicación de filtros de detección de borde y binarización para obtener bordes externos.

Ejercicio 2

Para la imagen "eye.jpg" realice los siguientes procesamientos.

(a) Cargar la imagen "eye.jpg" y convertirla a escala de grises. Luego, obtener una máscara binaria de las venas y arterias. Desplegar la máscara. Si usted desea obtener una máscara de mejor calidad, puede dividir la imagen en partes iguales con tal de escoger un umbral de binarización por zona. Por ejemplo, si dividimos la imagen en 4 partes iguales, es posible obtener 4 umbrales, correspondientes a los más precisos para cada zona. Para realizar la máscara, puede orientarse a partir de la Figura 4.

(b) Aplicar la máscara obtenida a la imagen original y desplegar la imagen original junto a la imagen obtenida, ambas en escala de grises.

(c) Aplicar tres filtros detectores de bordes basados en gradientes vistos en clases (Roberts, Sobel y Prewitt), por separado, a la imagen obtenida en el paso 2.b) y desplegar los resultados. Comparar el desempeño de cada uno de los filtros y comentar los resultados. Para uno de los filtros, debe utilizar la aplicación de la matríz de convolución vista en clases. Los otros dos, los puede aplicar usando su respectivo comando de cv2. Puede ver un ejemplo de lo anterior en la Figura 5. Debe comentar sobre el comportamiento de estos filtros respecto a su matríz de convolución, y dar una noción del gradiente de magnitud y dirección de cada uno de estos filtros.

(d) Mostrar los bordes sobre la imagen original. Para ello debe crear una imagen RGB donde los bordes queden resaltados con tonos amarillos y la imagen original aparezca en tonalidades verdes. Escoja la imagen de bordes obtenida en 2.c) que considere mejor. Recuerde que para realizar esto, es necesario trabajar cada canal por separado. El resultado esperado lo puede ver en la Figura 6.

(e) Dada la imagen original cargada en el paso a) ("eye.jpg") obtenga una imagen del ojo sin los vasos sanguíneos de este. Para lograr este resultado, se puede apoyar en la máscara de las venas y arterias obtenida en el paso a). Luego, a la imagen resultante, aplicar un filtro de suavizado para desenfocarlo y despliegue los resultados. Utilizar como fondo la imagen desenfocada que obtuvo en el paso anterior. Finalmente, debe lograr que las venas y las arterias se resalten

con color verde y que el fondo difuminado sea de color azul. Puede realizar esto asignando la máscara y el fondo por separado a los canales de una imagen RGB utilizando el comando *np.dstack(R,G,B)*, para unificar distintos canales en una sola imagen. Puede ver en la Figura 7 los pasos a seguir para obtener la figura RGB con venas resaltadas y fondo azul.

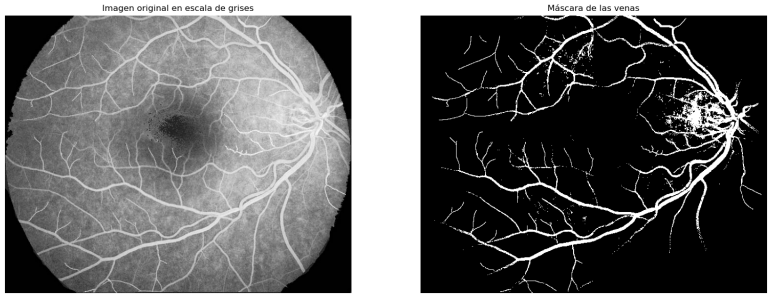

Figura 4: Se puede apreciar la imagen original y la máscara esperada.

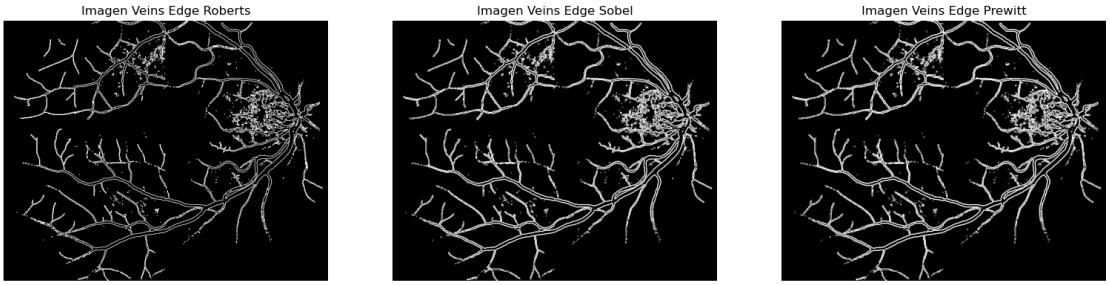

Figura 5: Aplicación de los filtros de detección de borde. Recuerde aplicar uno de estos filtros con su respectiva matríz de convolución y comentar sobre el comportamiento de estos dependiendo de su matríz de convolución.

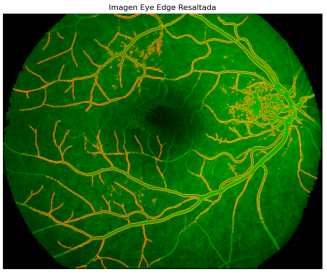

Figura 6: Imagen esperada con los bordes de los vasos sanguíneos resaltados.

Figura 7: Imagen complemento de los vasos, complemento suavizada e imagen con la aplicación de las operaciones de canales.

Ejercicio 3

Para las imágenes "tortugas1.jpg" y "tortugas2.jpeg" realice los siguientes procesamientos:

(a) Cargue y despliegue las imágenes, en escala de grises.

(b) Aplique los filtros de media, mediana y gaussiano vistos en clases (con el menor suavizado posible para reducir el ruido), a cada imagen. Despliegue los resultados de cada imagen en escala de grises y muestre los resultados en subplots con los nombre correspondientes. Elija el que considere mejor para cada imagen y explique la selección de cada filtro.

(c) Cree dos matrices de unos, con dimensiones de 3x3 y 5x5, respectivamente. Luego, mediante convolución, aplíquelas por separado sobre las imágenes originales en escala de grises. Muestre los resultados en subplots. Comente los resultados obtenidos. ¿Qué efecto realizaron las máscaras de convolución sobre el ruido de las imágenes?

(d) Cuantifique la imagen que considere mejor filtrada, obtenida de los incisos b) o c), con 16 niveles de gris y despliegue el resultado en una figura.

(e) Cree y despliegue una imagen de 3 canales a partir de la imagen cuantificada en el inciso anterior con los colores del mapa de colores 'cool' de la biblioteca matplotlib. Para obtener el mapa de colores utilice la función *get_cmap* disponible en la biblioteca matplotlib. Asigne los valores RGB del mapa a los píxeles que comparten el mismo nivel de cuantificación, como se muestra en la Figura 8.

(f) Aplicando umbralización adaptativa a la imagen filtrada, genere una imagen donde solo se muestre el contorno de la tortuga y despliegue la como se ven en la Figura 9.

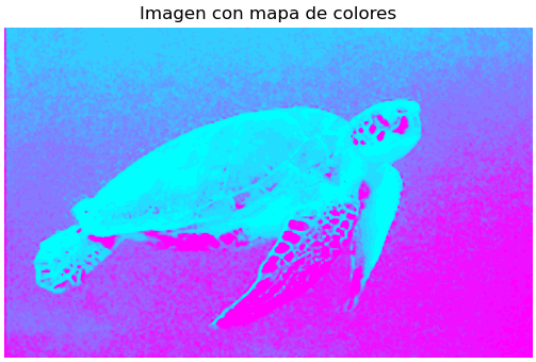

Figura 8: Imagen con mapa de colores aplicado

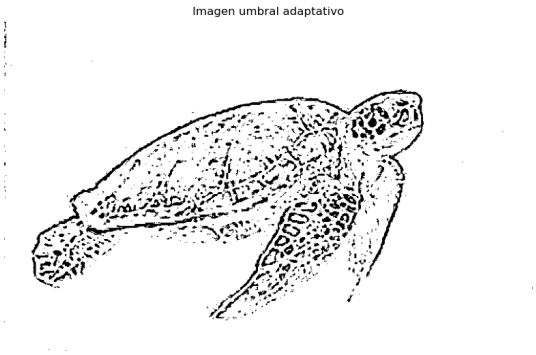

Figura 9: Imagen con umbralización adaptativa

Rúbrica

Ejercicio 1	20
Obtención de imagen espejo	4
suavizado de marca de agua	4
Aplicación de Multiotsu de 3 clases	4
Obtención de imagen con bordes externos	4
Código ordenado, comentarios y nombres de variables adecuados	2
Código sin errores de ejecución	2
Ejercicio 2	**20**
Obtención de la máscara	4
Aplicación de los filtros de detección de borde, usando matriz de convolución de alguno de estos.	4
Obtención de imagen con realce de bordes	4
Obtención de imagen de fondo difuminado y realce de vasos	4
Código ordenado, comentarios y nombres de variables adecuados	2
Código sin errores de ejecución	2
Ejercicio 3	**20**
Aplicación de filtro de reducción de ruido respaldado con comentarios	4
Aplicación de kernels de distintas dimensiones	4
Cuantización a 16 niveles de grises	4
Aplicación del mapa de colores cool.	2
Umbralización adaptativa	4
Código ordenado, comentarios y nombres de variables adecuados	1
Código sin errores de ejecución	1