E2-Filtros

Instrucciones

En respuesta a las preguntas que se plantean a continuación, se debe entregar en la plataforma de enseñanza virtual (por separado, no en un archivo comprimido)

- un archivo con extensión .ipynb creado con notebook Jupyter de Anaconda que contenga, tanto las respuestas teóricas, como el código programado;
- el archivo anterior en .pdf (File->Print preview->guardar como pdf);
- una declaración de autoría firmada por el alumno/a según la plantilla proporcionada;
- las imágenes/vídeos usados.

La resolución de todos los ejercicios (tanto teóricos como prácticos) debe ser **razonada** y el código desarrollado debe ser **explicado en detalle, justificando** todos los parámetros escogidos y referenciando las fuentes.

Objetivo

El objetivo de este entregable es dominar la manipulación de filtros.

Ejercicio 1. Ruido [0,6 ptos.]

Usando OpenCV, coge una imagen cualquiera a color y pásala a escala de grises (usando cv2.cvtColor).

Dado el siguiente código en Python:

```
import cv2

import numpy as np

import random


def ruido(image, a, b):

    aleat=np.random.randint(1,101, size = (image.shape[0], image.shape[1]))

    image2 = np.where(aleat <=100*a, 0, image)

    out = np.where(aleat > 100-100*b, 255, image2)

    return out
```

Explica detalladamente (línea a línea) qué hace la función implementada y relaciónalo con los ruidos explicados en el tema, considerando como datos de entrada una imagen en escala de grises y dos parámetros a y b entre 0 y 1, de manera que a+b también es un valor entre 0 y 1. ¿Cómo se modifican los píxeles de la imagen de entrada?, ¿qué píxeles se ven afectados y cómo varía la cantidad de píxeles afectados en función de a y b?

Aplícalo de dos formas distintas a una imagen I, con los dos pares de parámetros (a1=0.15, b1=0.15), para obtener la imagen con ruido I1 y (a2=0.3, b2=0), para obtener la imagen con ruido I2. Compara las dos imágenes obtenidas con respecto a la imagen original usando la medida MSE dada en clase (apartado 1.5). Para ello, resta las dos matrices (matriz de la imagen y matriz obtenida después de introducir el ruido), eleva la diferencia al cuadrado (**2) y calcula su media (np.mean). Explica por qué los valores obtenidos en los dos casos son parecidos.

Ejercicio 2. Filtros de suavizado [0,6 ptos.]

Considera una de las imágenes obtenidas I1 ó I2 anteriores.

a) Para tratar los píxeles del marco de la imagen al aplicar un filtro de suavizado, averigua y explica cuáles son las diferentes opciones implementadas en OpenCV y cuál es la que usa por defecto.
b) Aplica un filtro de media de orden 3x3 y otro de orden 5x5 para obtener una imagen suavizada del mismo tamaño que la original. ¿Qué diferencia hay entre este filtro y el Gaussiano?
c) Razona si estos filtros son apropiados o no para eliminar el ruido que se había introducido en la imagen I. En caso de que no sean la mejor opción, usa uno que sí lo sea, razonando tu elección. Calcula el parámetro SNR (apartado 1.5 del contenido de la asignatura) en cada caso para comparar la imagen recuperada con la imagen original.

Ayuda: https://docs.opencv.org/3.3.1/d4/d13/tutorial_py_filtering.html

Ejercicio 3. Filtros de realce [0, 65 ptos.]

Carga una imagen y pásala a escala de grises. Queremos obtener una imagen en escala de grises ([0,255]) en la que aparezcan con valores “altos” aquellos píxeles de la imagen original en cuyo entorno hay cambios bruscos en los niveles de gris (y con valores más bajos u oscuros aquellos que no). Para ello:

a. Usa la aproximación del gradiente basada en el operador de Sobel (**cv2.Sobel()**) apropiadamente.
b. Usa el Laplaciano (**cv2.Laplacian()**) apropiadamente para no perder la información de los resultados negativos.
c. Ahora prueba a realizar un suavizado gaussiano primero (explicando sus parámetros y eligiendo los más convenientes) y Sobel / Laplaciano después. Compara los

resultados con los anteriores (y entre ellos) razonando el porqué de las diferencias. ¿Para qué puede ser conveniente realizar el suavizado gaussiano previo?

Ejercicio 4. Filtros en el dominio de la frecuencia [0,65 ptos.]

Considera la imagen con ruido I2 del primer ejercicio.

Aplica la Transformada Discreta de Fourier con la ayuda del paquete FFT de numpy. Aplica un filtro en el dominio de la frecuencia, anulando las frecuencias apropiadas para poder eliminar el ruido. Vuelve al dominio del espacio para ver el resultado.

a) Explica los pasos detalladamente, comentando, además, cada línea del código aportado.
b) Explica el resultado en la imagen, relacionándolo con los filtros del dominio del espacio a los que se parece.
c) Prueba a variar la cantidad de frecuencias anuladas para obtener un resultado óptimo. ¿Qué efecto tiene en la imagen? ¿Por qué?

Ayuda: https://docs.opencv.org/master/de/dbc/tutorial_py_fourier_transform.html