E1-Introducción y preprocesamiento

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 Jupyter notebook 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.

IMPORTANTE: 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 realizar algunos ejercicios de transformaciones básicas y aplicación de filtros.

Instalación de OpenCV

Trabajaremos básicamente con la librería OpenCV en Python. Una forma sencilla de instalación es usando Anaconda https://www.anaconda.com/download e instalando OpenCV https://anaconda.org/conda-forge/opencv.

- Tutoriales básicos:
    - NumPy quickstart: https://numpy.org/doc/stable/user/quickstart.html
    - OpenCV-Python Tutorials: https://docs.opencv.org/4.x/d6/d00/tutorial_py_root.html
    - Matplotlib Quick start https://matplotlib.org/stable/users/explain/quick_start.html# Con matplotlib se pueden visualizar imágenes incrustadas en el notebook, pero hay que tener cuidado, porque se trata de una librería de visualización de gráficos que pueden estar renderizados para mejorar la visualización (por ejemplo, una imagen de poco contraste podría parecer que tiene buen contraste.

Ejercicio 1. Histograma y transformaciones de píxeles [0,8 puntos]

Consulta la información sobre cálculo y visualización del histograma de una imagen:

https://docs.opencv.org/4.x/de/db2/tutorial_py_table_of_contents_histograms.html

a) Considera una imagen fotográfica de mucho contraste. Cárgala en escala de grises y comprueba que tiene mucho contraste basándote en su histograma.
b) Aplica una *transformación de intensidad lineal* (da su expresión matemática) que la convierta en una imagen con poco contraste **muy clara,** sin que haya saturación. Diseña la transformación de manera que se pueda asegurar, a priori que el resultado se encuentra en [0,255] y manteniendo, en la medida de lo posible, "la forma" del histograma. Visualiza ahora su histograma para comprobar que el resultado es coherente.

Nota: consideraremos niveles de gris "muy claros" los que estén por encima de 175. La solución que se pide debe aclarar, pero no "desnaturalizar" la imagen.

Sugerencia: Asegúrate que las operaciones que realizas no cambian el tipo de dato de las imágenes (que es uint8 por defecto). De ser así, busca la forma de devolverlos a ese tipo de dato. También hay que tener precaución con las operaciones que se realizan con arrays de datos uint8, puesto que el resultado puede estar dándose módulo 256.

c) **Razona** (en base al crecimiento de la función) qué transformación de intensidades **no lineal** (tipo log, exp, raíz, potencias) podría mejorar el contraste de la imagen obtenida en el apartado b) y con qué formulación explícita para poder controlar el intervalo de salida de los niveles de gris. Implementa dicha transformación de manera que el resultado se ajuste, a priori, al intervalo [0,255], tratando de perder la menor información posible (de forma razonada).
d) Realiza una ecualización del histograma de la imagen obtenida en b). ¿Es mejor solución para mejorar el contraste que la transformación del apartado c)? ¿Por qué? Compara los histogramas para razonarlo.
e) Usa el MSE como medida para decidir qué operación recupera mejor la imagen original en a), si la transformación de c) o la ecualización de d).

Ejercicio 2. Filtros [0,4 ptos.]

Escoge una imagen que contenga, entre otras cosas, líneas diagonales (45 grados), horizontales y verticales. Cárgala en escala de grises.

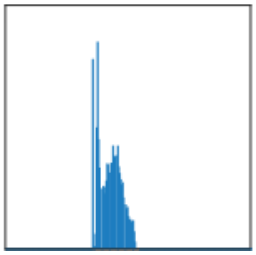
a) Diseña razonadamente un filtro 5x5 que dé una respuesta fuerte (valores claros) sobre los píxeles en las líneas diagonales, pero no en el resto (donde debe producir valores oscuros). Impleméntalo y explica funciones y parámetros usados.
a) Aplica otro tipo de filtro, de los explicados en clase y ya definidos en OpenCV que pueda realzar todos los tipos de líneas que aparecen en la imagen (píxeles claros sobre las

líneas, píxeles oscuros en el resto), razonando el por qué ocurre esto y si se puede conseguir sólo el realce de las líneas rectas.

Ayuda: https://docs.opencv.org/4.x/d4/d13/tutorial_py_filtering.html
https://docs.opencv.org/4.x/d5/d0f/tutorial_py_gradients.html

Ejercicio 3. [0,3 puntos]

1. En el estándar de compresión JPEG:
    a. ¿En qué paso se produce la pérdida de información?
    b. ¿En qué consisten los distintos grados de compresión que se pueden realizar en la imagen?
2. Razona si puede ser beneficioso usar Huffmann para codificar los niveles de gris de la imagen de 8 bits correspondiente a este histograma.



¿Cómo podríamos comprobar "cuánto de beneficioso" sería?

3. Razona por qué el modelo de color HSV puede ser más útil que el RGB para detectar objetos de un color determinado en una imagen. Pruébalo con una imagen con un objeto de color fuerte. Cárgala, pásala a HSV y explica cómo detectar el objeto de color en la imagen.