

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 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 realizar algunos ejercicios de transformaciones básicas y aplicación de filtros.

- Tutoriales básicos:
 - NumPy quickstart: <https://numpy.org/devdocs/user/quickstart.html>
 - OpenCV-Python Tutorials:
https://docs.opencv.org/master/d6/d00/tutorial_py_root.html
 - Matplotlib Quick start
https://matplotlib.org/stable/users/explain/quick_start.html#quick-start

Ejercicio 1. Histograma y transformaciones de píxeles [0,7 puntos]

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

https://docs.opencv.org/4.9.0/de/db2/tutorial_py_table_of_contents_histograms.html

Considera una imagen fotográfica en escala de grises cuyos niveles de gris estén ampliamente **repartidos** en el intervalo $[0,255]$ (y sea, por tanto, una imagen de mucho contraste).

- a) Calcula y visualiza su histograma. Aplica una *transformación de intensidad lineal* 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.

- b) Razona qué transformación de intensidades **no lineal** (tipo log, exp, raíz, potencias) podría mejorar el contraste de la imagen obtenida en el apartado a).
- c) 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).

Ejercicio 2. Filtros de suavizado [0,8 ptos.]

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

Copia el siguiente código en Python:

```
import cv2
```

```
import numpy as np
```

```
import random
```

```
def transf(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
```

Considerando como datos de entrada una imagen en escala de grises y dos parámetros a y b entre 0 y 1 (los dos), de manera que a+b también es un valor entre 0 y 1:

- a) ¿Qué efecto tendrá en la imagen de entrada la función implementada? ¿Qué significado tendrán los valores a y b? ¿Qué píxeles se ven afectados por la transformación y cómo varía la cantidad de píxeles afectados en función de a y b?

Aplicalo de dos formas distintas a una imagen I, con los dos pares de parámetros (a1=0.1, b1=0.1), para obtener la imagen I1 y (a2=0.2, b2=0), para obtener la imagen I2. Compara las dos

imágenes obtenidas con respecto a la imagen original usando la medida MSE dada en clase (apartado 1.5).

Considera una de las imágenes obtenidas I1 ó I2 anteriores.

- b) Aplica un filtro de media de orden 3x3 y otro de orden 5x5 para obtener una imagen filtrada del mismo tamaño que la original. Explica el efecto. ¿Qué diferencia hay entre este filtro y el Gaussiano?
- c) Razona si estos filtros (media, Gaussiano) son apropiados o no para restaurar la imagen I. En caso de que no sean la mejor opción, usa uno que sí lo sea, razonando tu elección. Calcula el parámetro MSE (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/4.9.0/d4/d13/tutorial_py_filtering.html