

# Práctica 1

**Asignatura:** Visión por Computadora

**Docentes:**

- Brugé, Lucas
- Ferrucci, Constantino
- Manson, Juan Pablo

Esta práctica está diseñada para introducirte en las técnicas fundamentales de procesamiento de imágenes necesarias para preparar datasets efectivos para proyectos de Computer Vision.

## Ejercicios

### Ejercicio 1: Creación de Imagen con Cuatro Cuadrantes de Color

Genera una imagen de 200x200 píxeles utilizando `numpy`, donde cada cuadrante tenga un color diferente:

- **Superior izquierdo:** Rojo (255, 0, 0)
- **Superior derecho:** Verde (0, 255, 0)
- **Inferior izquierdo:** Azul (0, 0, 255)
- **Inferior derecho:** Blanco (255, 255, 255)

Muestra la imagen generada utilizando `matplotlib`.

---

### Ejercicio 2: Conversión de Imagen a Escala de Grises

1. Carga una imagen desde un archivo o desde una URL usando `OpenCV`.
  2. Escribe dos funciones para convertir la imagen a escala de grises:
    - **Método 1:** Promediando los canales RGB manualmente.
    - **Método 2:** Usando `cv2.cvtColor()` de OpenCV.
  3. Muestra las tres imágenes juntas (original, método 1 y método 2) y compara los resultados.
-

## Ejercicio 3: Extracción de Metadatos EXIF

1. Carga una imagen tomada con una cámara o un celular.
  2. Utiliza `exifread` y `Pillow` para extraer los metadatos EXIF.
  3. Imprime estos valores y explica qué significan.
- 

## Ejercicio 4: Comparación de efectos de compresión

1. Carga una imagen en formato JPG usando `OpenCV`.
2. Convierte la imagen a RGB para su correcta visualización en `Matplotlib`.
3. Guarda la imagen en formato JPG con distintos niveles de calidad:
  - **Calidad alta:** 95%
  - **Calidad media:** 50%
  - **Calidad baja:** 10%
4. Carga las imágenes comprimidas y conviértelas a RGB para su correcta visualización.
5. Muestra las imágenes en una sola figura comparando la imagen original con las versiones comprimidas.

**Pregunta de análisis:** ¿Qué efectos visuales observas en las imágenes con menor calidad? (artefactos de compresión, pérdida de detalles, ruido).

---

## Ejercicio 5: Comparación de distintos formatos de imagen

1. Carga una imagen en formato JPG usando `OpenCV`.
2. Guarda la imagen en diferentes formatos con distintos niveles de compresión:
  - **PNG** con máxima compresión ( `compression level 9` ).
  - **WEBP** con calidad del 90%.
3. Obtén el tamaño de los archivos generados en KB y compáralos con el original.
4. Visualiza las imágenes convertidas en una figura comparativa junto con la imagen original.

**Pregunta de análisis:** ¿Qué formato resultó en el archivo más liviano y cuál en el más pesado? Justificar.

---

## Ejercicio 6: Preparación de imágenes para datasets

1. Carga una imagen en formato `.png` o `.jpg`.
  2. Redimensiona la imagen a 256 píxeles de ancho manteniendo la relación de aspecto.
  3. Convierte la imagen a formato `.jpg` si no lo está.
  4. Verifica si la imagen está corrupta al intentar cargarla con `OpenCV`.
  5. Normaliza los valores de los píxeles dividiendo entre 255.0 y muestra la imagen resultante.
- 

## Ejercicio 7: Detección de Imágenes Corruptas

1. Escribe una función que recorra una carpeta con imágenes y detecte cuáles están corruptas o no pueden abrirse con `OpenCV`.
  2. Prueba la función con un conjunto de imágenes, incluyendo una dañada o con extensión incorrecta.
- 

## Ejercicio 8: Recorte y Padding para Estandarizar Tamaños

1. Carga una imagen con dimensiones aleatorias.
  2. Recórtala para hacerla cuadrada manteniendo el centro de la imagen.
  3. Si es más pequeña que 256x256 píxeles, agrégale padding negro hasta alcanzar ese tamaño.
  4. Muestra la imagen original y la ajustada.
- 

## Ejercicio 9: Mantenimiento de Relación de Aspecto al Redimensionar

1. Carga una imagen con una relación de aspecto no cuadrada.
  2. Redimensiona la imagen a 256 píxeles de alto, ajustando el ancho proporcionalmente.
  3. Agrega padding si es necesario para que la imagen final sea 256x256 sin deformarse.
- 

## Ejercicio 10: Aumentación de Datos Básica

1. Carga una imagen y crea 5 variaciones usando técnicas básicas de aumentación:

- Rotación (15 grados)
- Espejo horizontal
- Cambio de brillo (+20%)
- Recorte aleatorio del 10%
- Ligeramente desenfocado gaussiano

2. Visualiza las imágenes originales y aumentadas en una sola figura.

---

## Tarea intermedia

A partir de este punto vamos a hacer una recolección de imágenes previo a avanzar ante los próximos ejercicios por lo que, como consigna intermedia, se solicita:

- Cargar un conjunto de imágenes (Pueden ser obtenidas a través de internet)

Una vez realizado, se podrá continuar con los ejercicios.

## Ejercicio 11: Organización de Dataset en Carpetas

1. Escribe un script que organice automáticamente un conjunto de imágenes en carpetas según:
    - Dimensiones (pequeñas, medianas, grandes)
    - Orientación (horizontal, vertical, cuadrada)
  2. Genera un informe sencillo de cuántas imágenes hay en cada categoría.
- 

## Ejercicio 12: Estandarización de Nombres de Archivos

1. Renombra todas las imágenes de una carpeta siguiendo un patrón consistente:
    - Formato: `imagen_001.jpg` , `imagen_002.jpg` , etc.
    - Asegúrate que la numeración sea consecutiva
  2. Guarda un archivo CSV con el mapeo entre nombres originales y nuevos.
-

## Ejercicio 13: Eliminación de Duplicados

1. Escribe una función que identifique imágenes duplicadas o muy similares en un dataset.
  2. Usa un método simple como comparar histogramas de color.
  3. Muestra las imágenes detectadas como posibles duplicados.
- 

## Ejercicio 14: Generación de Miniaturas para Previsualización Rápida

1. Crea versiones en miniatura (64x64 píxeles) de todas las imágenes de un dataset.
  2. Guárdalas en una subcarpeta llamada "thumbnails".
  3. Genera una hoja de contacto que muestre todas las miniaturas en una sola imagen grande.
- 

## Ejercicio 15: Extracción de Características Básicas

1. Calcula y grafica histogramas RGB para un conjunto de imágenes.
  2. Clasifica las imágenes como "oscuras", "brillantes" o "balanceadas" según los histogramas.
  3. Organiza o etiqueta el dataset según estas características.
- 

## Ejercicio 16: Filtrado por Calidad Básica

1. Implementa un filtro simple que descarte imágenes con problemas evidentes:
    - Demasiado oscuras (promedio de brillo  $< 50/255$ )
    - Demasiado brillantes (promedio de brillo  $> 220/255$ )
    - Demasiado pequeñas (menos de 100x100 píxeles)
    - Muy baja varianza (posiblemente imágenes en blanco o de un solo color)
- 

## Ejercicio 17: División en Conjuntos de Entrenamiento y Validación

1. Divide automáticamente un dataset en conjuntos de entrenamiento (80%) y validación (20%).
2. Asegúrate que la división mantenga la proporción de clases en ambos conjuntos.
3. Guarda los conjuntos en carpetas separadas conservando la estructura original.