

# Visión por Computador

## Introducción a OpenCV

**Fecha de entrega: 30 de Septiembre (PRADO)**

**Bonus: hasta 3 puntos**

Esta práctica, de la que los estudiantes deben entregar código y memoria explicativa, está dirigida a familiarizarse con el uso de OpenCV y Python en dos escenarios: en el ordenador portátil y en la herramienta “Colab” de Google. En el ordenador portátil se puede trabajar con IDEs como Spyder o Jupyter notebook. En “Colab” solo se puede trabajar con un IDE web muy similar a Jupyter. Usar el manual de referencia y la ayuda on-line de OpenCV ([https://docs.opencv.org/4.4.0/d6/d00/tutorial\\_py\\_root.html](https://docs.opencv.org/4.4.0/d6/d00/tutorial_py_root.html)) para el uso de sus funciones.

- **Tarea 1:** Instalar en su portátil Anaconda, OpenCV y Spider usando la Guía de instalación proporcionada.
- **Tarea 2:** Darse de alta como usuario Google o verificar que se dispone de una cuenta para acceder a la herramienta Drive de Google. Una vez en Drive iniciar “Colab”. Crear un directorio en su drive de Google y subir a él las imágenes de la práctica. La ruta debe ser './drive/My Drive/Colab Notebooks/images/'
- **Tarea 3:** Escribir un script en Python que resuelva los ejercicios mostrados más abajo. Ejecutar el script en Colab, empleando una celda por ejercicio (de modo que se puedan ejecutar los distintos ejercicios de modo independiente) y mostrar que se obtienen los mismos resultados que en el ordenador personal. Los alumnos deben entregar tanto el código .py como el código .ipynb. El siguiente documentos puede resultar de utilidad de cara a escribir código de visualización de imágenes compatible en ambos IDEs (<https://colab.research.google.com/drive/1RWGmqoEQdeyh5TssGtsXsFk8hbLgtWp>)
- **Ejercicios:**
  1. Escribir una función que lea el fichero de una imagen y permita mostrarla tanto en grises como en color (`im=leeimagen(filename, flagColor)`). *flagColor* es la variable que determina si la imagen se muestra en escala de grises o en color.
  2. Escribir una función que permita visualizar una matriz de números reales cualquiera/arbitraria, tanto monobanda como tribanda (`pintaI(im)`). Para ello se deberá escalar el rango de cada banda al intervalo [0,1] sin pérdida de información.
  3. Escribir una función que visualice varias imágenes distintas a la vez (concatenando las imágenes en una última imagen final<sup>1</sup>): `pintaMI(vim)`. (*vim* será una secuencia de imágenes) ¿Qué pasa si las imágenes no son todas del mismo tipo (nivel de gris, color, blanco-negro)?

---

<sup>1</sup> Es decir, se desea obtener una imagen final concatenando las imágenes insertadas en el vector que se pasa como argumento a la función

4. Escribir una función que modifique el color en la imagen de cada uno de los elementos de una lista de coordenadas de píxeles<sup>2</sup>. En concreto, los alumnos deben insertar un cuadrado azul de 100x100 píxeles en el centro de la imagen a modificar.
5. Una función que sea capaz de representar varias imágenes con sus títulos en una misma ventana.

Usar las imágenes del directorio *images* en *practica-0\_python.rar*. Además, este fichero contiene un ejercicio sobre el lenguaje Python y algunas ayudas con OpenCV para aquellos que nunca antes hayan usado Python u OpenCV.

---

<sup>2</sup> Recuérdese que (fila, columna) es lo contrario a (x,y). Es decir fila=y, columna=x