

Universidad Pontificia de Comillas

ICAI

## LABORATORIO 2

Visión por Ordenador I

Lydia Ruiz Martínez, Pablo Tuñón Laguna

## PROCESAMIENTO DE IMÁGENES



Curso 2024-2025

## Índice

1. Introducción	2
2. Segmentación por color	3
3. Filtro Gaussiano y Detección de bordes: Sobel y Canny	4
4. Operadores morfológicos	4

# 1. Introducción

El procesamiento de imágenes es un área fundamental de la visión por ordenador que se enfoca en la manipulación y el análisis de imágenes digitales. A diferencia de un ser humano, que puede interpretar imágenes de manera intuitiva, un ordenador necesita algoritmos específicos que le permitan extraer información útil de las matrices de píxeles que componen cada imagen. Estos algoritmos permiten a los sistemas de visión por ordenador realizar tareas como la segmentación, la detección de bordes y la aplicación de filtros, convirtiendo datos visuales en información comprensible.

La segmentación de imágenes por color es una técnica crucial que permite identificar y aislar objetos de interés en una imagen basándose en sus características cromáticas. Este proceso es esencial en múltiples aplicaciones, desde la automatización industrial hasta la medicina, donde la identificación precisa de regiones de interés puede marcar la diferencia en el análisis y la toma de decisiones. La capacidad de un ordenador para diferenciar colores y reconocer patrones en el espacio de color adecuado es un paso inicial vital en la interpretación de las imágenes.

Además, la aplicación de filtros es otra herramienta poderosa en el procesamiento de imágenes. Filtros como el gaussiano, Sobel y Canny se utilizan para suavizar imágenes, eliminar el ruido y detectar bordes, respectivamente. La detección de bordes es especialmente relevante, ya que resalta las transiciones significativas en una imagen, facilitando la identificación de formas y estructuras. Al aplicar y comparar estos filtros, se pueden evaluar sus efectos en la calidad de la imagen y la efectividad del análisis posterior.

Finalmente, el uso de operadores morfológicos complementa estas técnicas al permitir la manipulación de la estructura de los objetos en las imágenes a nivel de píxel. Este enfoque es fundamental para mejorar los resultados de la segmentación y la detección, proporcionando una comprensión más profunda de las características presentes en las imágenes. En esta práctica, se implementarán estas técnicas utilizando Python y las librerías OpenCV e Imageio principalmente.

## 2. Segmentación por color

La segmentación de imágenes por colores es una técnica sencilla y útil en aplicaciones en las que se tiene un gran control sobre las condiciones de contorno: iluminación, tipo de objeto que se espera encontrar, color de fondo, etc. En esta práctica, se enfocará en la segmentación de los colores blancos y naranjas en imágenes de peces.

Para comenzar, se procede a cargar las imágenes desde la carpeta donde se almacenan los archivos de las imágenes a procesar. Una vez cargadas, se implementan métodos para visualizar y guardar las imágenes, permitiendo una fácil manipulación y verificación de los resultados.

El siguiente paso consiste en cambiar el espacio de color de las imágenes de *BGR* a *HSV*, lo cual es esencial para una mejor discriminación de los colores de interés. Este cambio de espacio de color transforma la representación de los colores, separando la crominancia de la intensidad, y facilita la identificación de los tonos deseados. Utilizando el método `cv2.cvtColor()` de OpenCV, se transformarán las imágenes para prepararlas para la segmentación. Este cambio es similar a convertir un sistema de coordenadas cartesianas a uno polar, donde los colores se pueden manejar de manera más intuitiva.

Una vez que las imágenes estén en el espacio de color adecuado, se procede a generar máscaras para los colores naranjas y blancos, utilizando los métodos `cv2.inRange()` y `cv2.bitwise_and()` de OpenCV. Estas máscaras permiten segmentar las partes relevantes de las imágenes, destacando los píxeles de interés y eliminando aquellos que no corresponden.



Figura 1: Imagen 1 original, dataset.



Figura 2: Imagen 1 corregida, autoría propia.



Figura 3: Imagen 2 original, dataset.

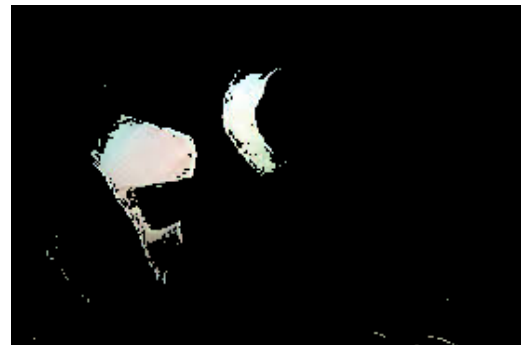


Figura 4: Imagen 2 corregida, autoría propia.

### 3. Filtro Gaussiano y Detección de bordes: Sobel y Canny

### 4. Operadores morfológicos

En esta sección, se implementan los operadores morfológicos de dilatación y erosión utilizando un kernel de  $3 \times 3$  con valores de 255 (blanco) como structuring element.

Se define el método de binarización, el cual convierte una imagen a escala de grises a una imagen binaria utilizando un umbral de 127. Este método emplea la función `cv2.threshold` de OpenCV para realizar la conversión.

Se implementa el método de dilatación, que consiste en ampliar las áreas blancas de la imagen binaria. Para ello, se utiliza un padding de la imagen original para mantener las dimensiones tras el procesamiento. En este método, se recorren todos

los píxeles de la imagen, y se reemplaza cada píxel por el valor máximo de sus vecinos en el kernel.

Asimismo, se implementa el método de erosión, que reduce las áreas blancas de la imagen. Al igual que en la dilatación, se utiliza un padding para conservar las dimensiones. En este caso, cada píxel se reemplaza por el valor mínimo de sus vecinos en el kernel.



Figura 5: Ajuste fino de intersección de cuadrícula, autoría propia.



Figura 6: Ajuste fino de intersección de cuadrícula, autoría propia.