

# Visión por Computadora 2025

## Lab 2

13.febrero.2025

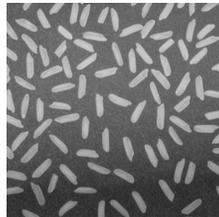
En este lab trabajaremos operadores morfológicos en imágenes binarias y escala de grises. Para referencia, revisar el **capítulo 9** del libro *Digital Image Processing* de González y Woods.

1. Leer la sección 9.4 sobre el operador *Hit or Miss*. Explicar con sus propias palabras cómo funciona este operador, y cómo se construye la operación  $A \otimes B$  a partir de las operaciones básicas de dilatación, erosión, opening, closing y diferencia.

Mostrar un ejemplo con una imagen binaria de su elección en donde se muestre que el operador *Hit or Miss* detecta la localización de uno (o varios) objetos específicos dentro de una imagen binaria  $I$ .

2. Consideramos de nuevo la imagen `rice.jpg`. Hacer lo siguiente:

- a) Binarizar la imagen mediante el método de Otsu, o usando algún otro método no local.
- b) Aplicar operaciones morfológicas adecuadas para limpiar el ruido de la binarización. Explicar por qué está eligiendo estas transformaciones morfológicas.
- c) Aplicar un algoritmo de componentes conexas sobre la binarización y contar cuántos granos de arroz hay en la imagen.



3. Aplicar diferentes operaciones morfológicas a las siguientes imágenes en escala de grises (si la imagen no está en grises, primero convertir a escala de grises): `butterfly.jpeg`, `quetzalgris.png`, `chestXray.jpeg`.

Explicar

- ¿Cuál es el efecto de aplicar dilatación y erosión a estas imágenes?
- ¿Cuál es el efecto de aplicar opening y closing a estas imágenes?
- ¿Qué hace el white top-hat? ¿Para qué puede ser útil?

4. Obtener el gradiente morfológico  $\nabla(I) = (I \oplus B) - (I \ominus B)$  de la imagen `brain-scan.jpeg`.

Explicar cuál es el resultado obtenido y explicar por qué se le llama gradiente. (Sugerencia: experimentar con otras imágenes en escala de grises para entender el resultado de esta operación).



5. Sobre la imagen `microscope.png`, aplicar los siguientes pasos:

- Binarizar la imagen (si no está binarizada).
- Obtener la componente conexas de menor tamaño. ¿Cuántos píxeles tiene?
- Recortar la componente conexas de mayor tamaño, y remover el resto de componentes. Mostrar una imagen binaria donde sólo quede la mayor componente conexas.
- Aplicar operaciones morfológicas para contar cuántas células hay en total. ¿Comparar el número obtenido contra el número verdadero de células? ¿Son iguales?

6. Para la imagen `wheat.png`, hacer lo siguiente:

- Segmentar la imagen de forma que se identifique de forma separada cada grano de trigo mediante el método de *Watershed*.
- Construir el mapa de clases, pintando de color negro el fondo, y cada grano con un color distinto.
- Combinar la imagen original con una versión pálida del mapa de colores (modifique el canal  $\alpha$  del mapa de clases y súmelo con la imagen original), para mostrar los colores obtenidos en el mapa de clases sobre cada grano, como se ilustra en la siguiente figura.



7. Leer la subsección **Skeletons** de la sección 9.5 del libro, y explique con sus palabras cómo se construye el esqueleto de un objeto binario mediante operaciones morfológicas.

Investigue como se implementan en Python las operaciones de (i) esqueletización, (ii) transformada de distancia desde el borde, y (iii) transformada de distancia desde el esqueleto. Puede usar Scikit-image, OpenCV, o alguna otra librería de su preferencia.

Calcule estas tres transformaciones para una imagen binaria de su elección.

8. Leer la subsección **Prunning** y explique con sus palabras cómo se eliminan las ramificaciones no deseadas de un objeto binario mediante operaciones morfológicas.

Aplicar estas ideas para remover las ramificaciones no deseadas del esqueleto en el ejercicio anterior.