

Segmentación

Visión Artificial II

Máster Universitario en Informática Industrial y Robótica

Alejandro Paz López

Alma Mallo Casdelo

Contenidos

- Introducción.
- Técnicas de segmentación.
 - Técnicas orientados a regiones.
 - Técnicas orientadas a bordes.
 - Técnicas de clustering.
 - Técnicas basadas en deep learning.
- Validación de la segmentación.

Introducción

Segmentación

- **Segmentar una imagen:** dividirla en zonas disjuntas para diferenciar los diversos objetos (y el fondo).
- Al final de la etapa de segmentación:
 - Cada pixel de la imagen tiene que tener una **etiqueta** de modo que los objetos queden definidos por agrupación de puntos con la misma etiqueta.
 - Tiene que conocerse los objetos existentes para extraer las características de cada uno de ellos. Realmente no son objetos sino **regiones en la imagen** (un objeto podría ocupar varias regiones).

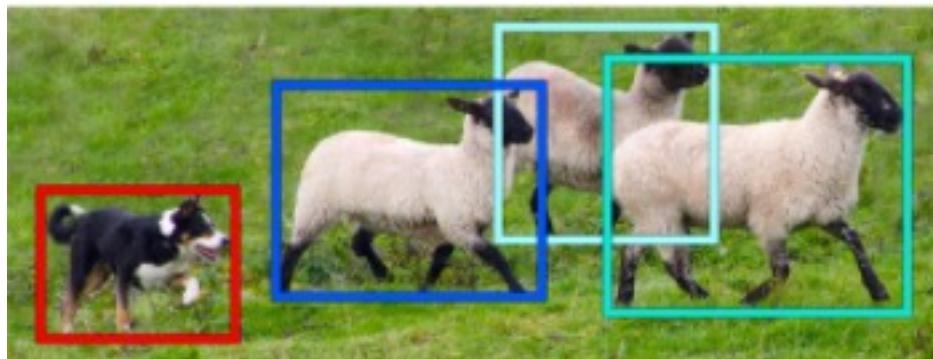
Segmentación

- Agrupar píxeles por algún criterio de **homogeneidad**.
 - Extracción de los objetos de interés insertados en la escena capturada.
 - Se busca una correspondencia entre características de bajo nivel (color, textura, intensidad, bordes, etc.) con objetos (o partes de objetos) del mundo físico.

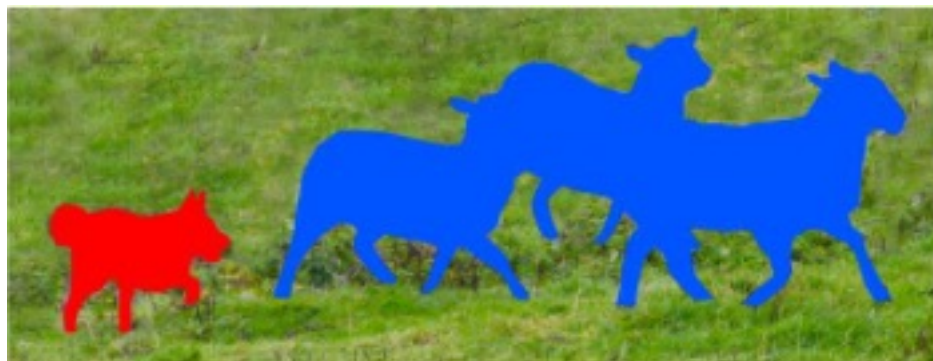


Segmentación

Reconocimiento de objetos
(clasificación + localización con bounding-box)



Segmentación semántica



Segmentación de instancias

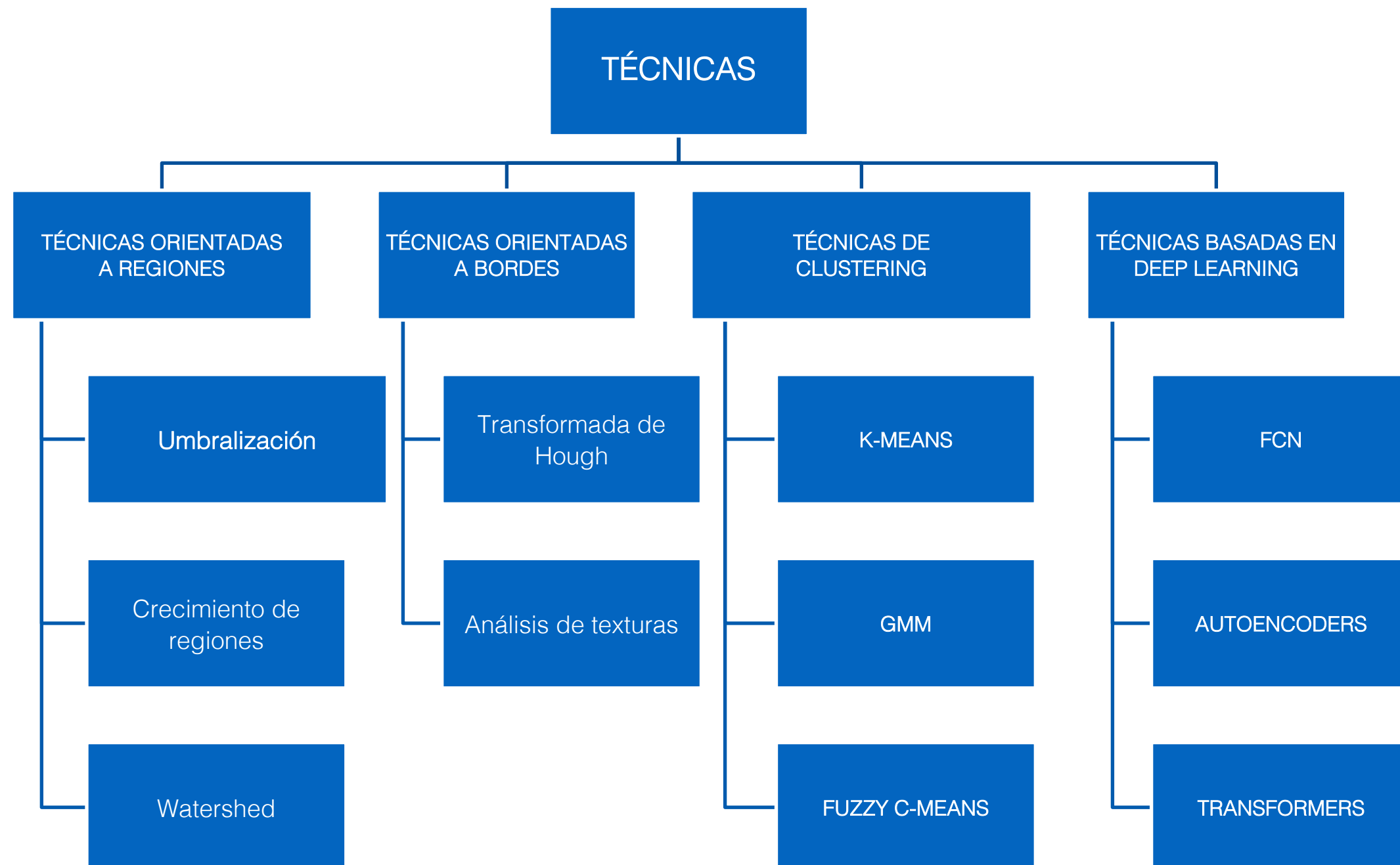


- **Semantic segmentation**
 - Se identifican los píxeles correspondientes a un determinado número de clases (una o más).
- **Instance segmentation**
 - Se diferencian los píxeles correspondientes a diferentes objetos.
 - No se conoce a que clase de objeto pertenece cada una de las regiones, simplemente se identifican como diferentes.
- **Panoptic segmentation**
 - Combinación de segmentación semántica y segmentación de instancia.
 - Se predice la clase de cada región individual respecto a su identidad de objeto.

Segmentación

- División de una imagen en regiones (según algún criterio).
 - Dentro de una región la imagen debe ser uniforme.
 - Regiones adyacentes deben ser distintas.
 - Si conocemos de antemano la forma y tamaño de los objetos el problema se simplifica.
 - En ámbitos industriales los objetos (o regiones de ellos) a menudo tienen formas geométricas relativamente sencillas.
- Problema complejo.
 - No existe una solución general buena.
 - El éxito de unas u otras aproximaciones depende de las características de la imagen.

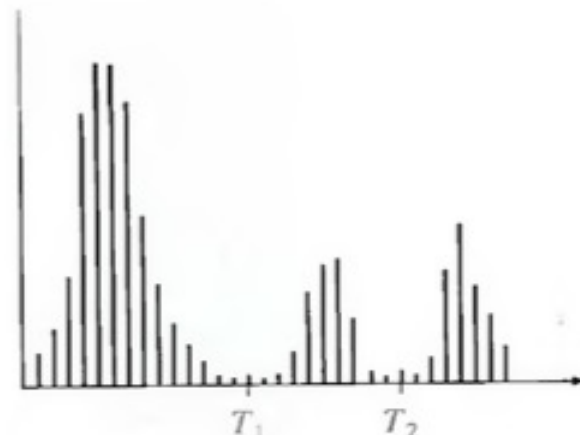
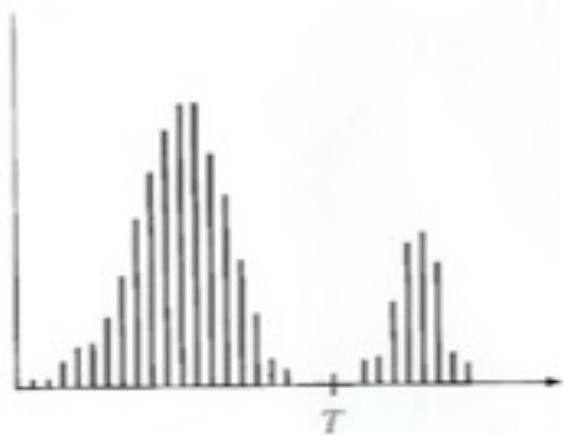
Técnicas de segmentación



Técnicas orientadas a regiones

Umbralización

- El histograma de la imagen mostrará si los píxeles de la imagen se agrupan en zonas.
- Una forma sencilla de segmentar los objetos consiste en determinar el umbral que separa los píxeles claros de los oscuros.
- Si tenemos varios objetos distintos en la imagen será necesario determinar varios umbrales, o bien distinguirlos en base a otras características.



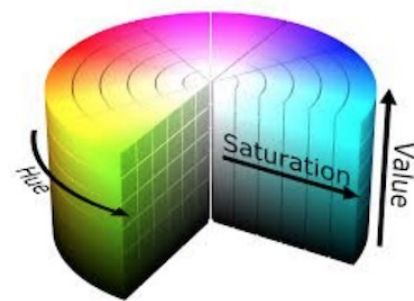
Umbralización

- Tipos:
 - Global: el umbral depende de la intensidad global de la imagen.
 - Algoritmo OTSU (uno de los más habituales). Minimiza la varianza en ambas clases.
 - Local: el umbral depende de ciertas propiedades en un vecindario.

Umbralización basada en el color



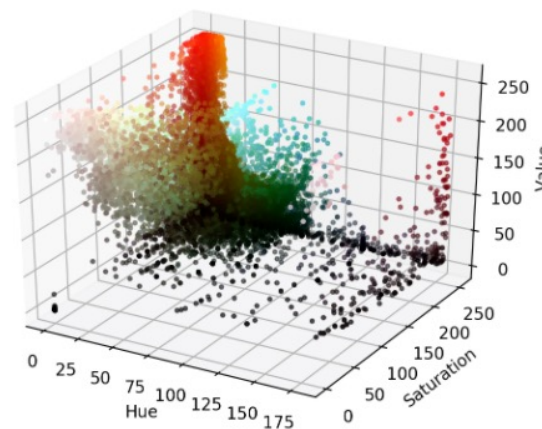
Espacio de color HSV de la imagen



FILTRO POR RANGO DE COLOR HSV

light_orange = (1, 190, 200)

dark_orange = (18, 255, 255)

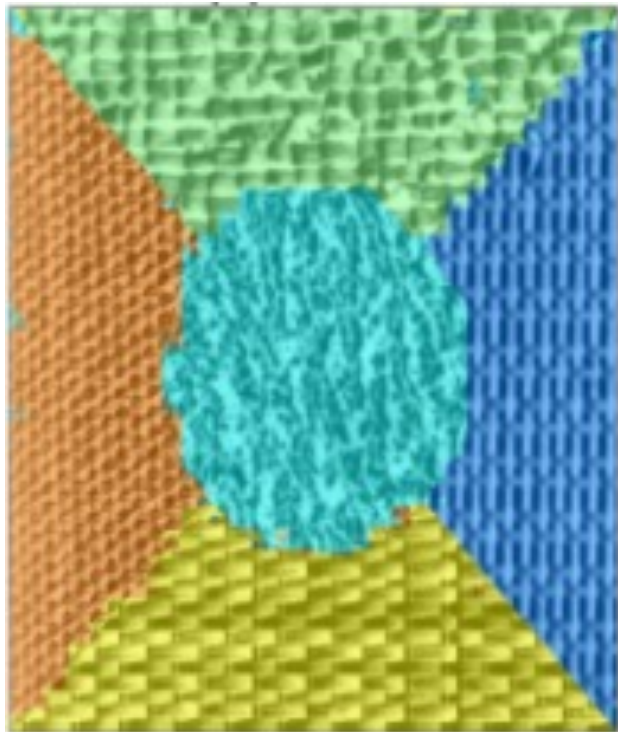


Umbralización

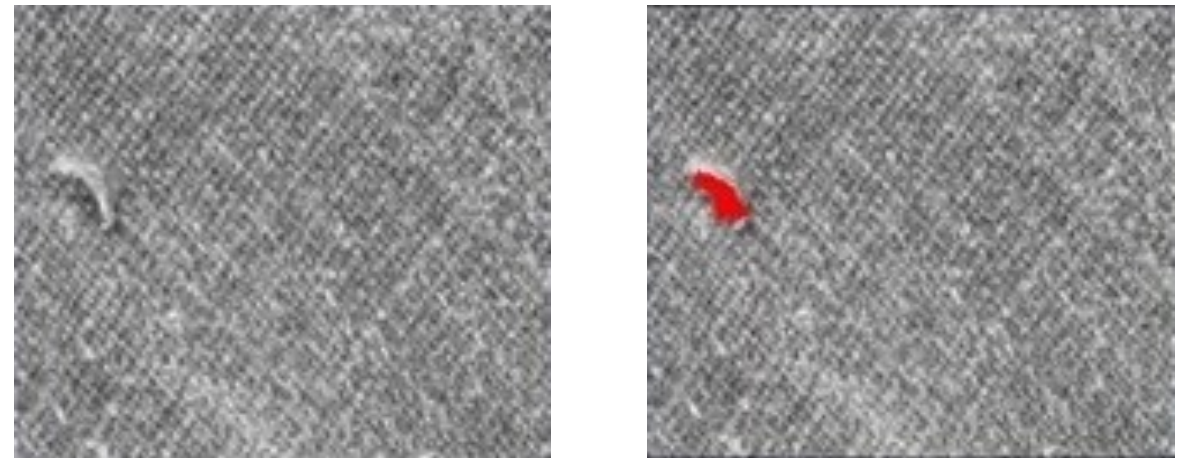
- Dificultades:
 - Umbralización por histograma bimodal: una imagen **no siempre tiene un único objeto** (o tipo de objeto) sobre el fondo.
 - A menudo los objetos también se caracterizan por **propiedades distintas** al nivel de intensidad, como por ejemplo, su textura.

Umbralización

- Dificultades:
 - Ejemplo: imágenes con textura.



Extracción de regiones basada en la textura de la imagen



Identificación de defecto basada en la textura de la imagen

Crecimiento de regiones

- Agrupación de píxeles o subregiones en regiones más grandes en base a un criterio preestablecido.
 - Se parte de un conjunto de **puntos semilla**.
 - Se añade a cada semilla los puntos **vecinos con propiedades similares** a la semilla (ej. media del nivel gris en la región).
- Selección de semillas:
 - Teniendo en cuenta propiedades de interés (picos del histograma, textura, etc.)
- Se pueden introducir otros criterios para mejorar la segmentación (ej. tamaño o forma esperados de la región).

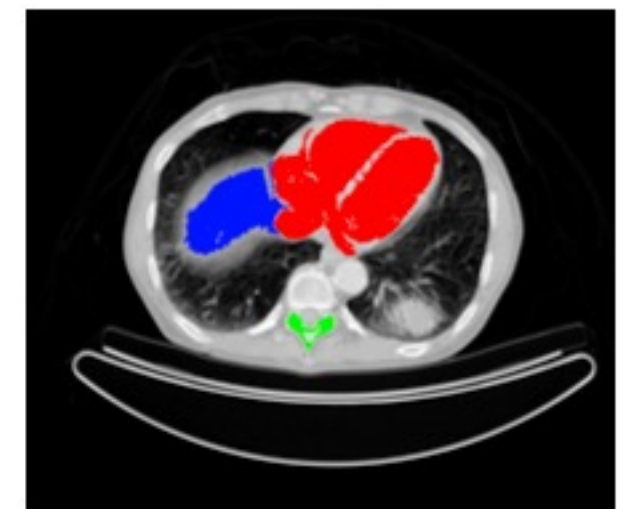
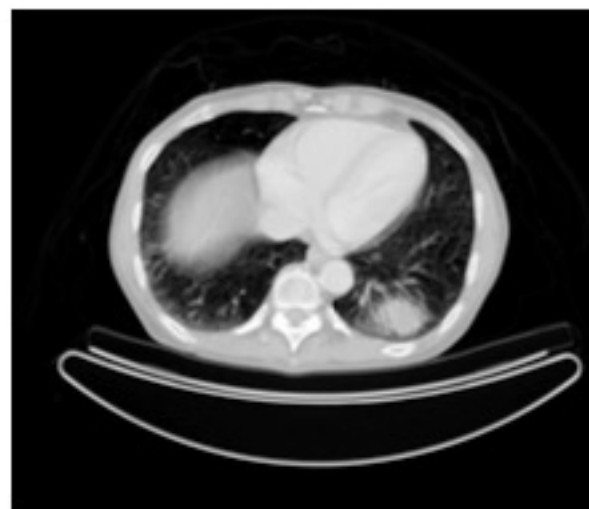
Crecimiento de regiones

1	2	5	8	9
2	4	6	0	3
2	4	5	6	2
8	8	2	9	3
5	7	8	9	2

1	2	5	8	9
2	4	6	0	3
2	4	5	6	2
8	8	2	9	3
5	7	8	9	2

A partir de una semilla se explora iterativamente el vecindario de píxeles en base a un criterio de similitud

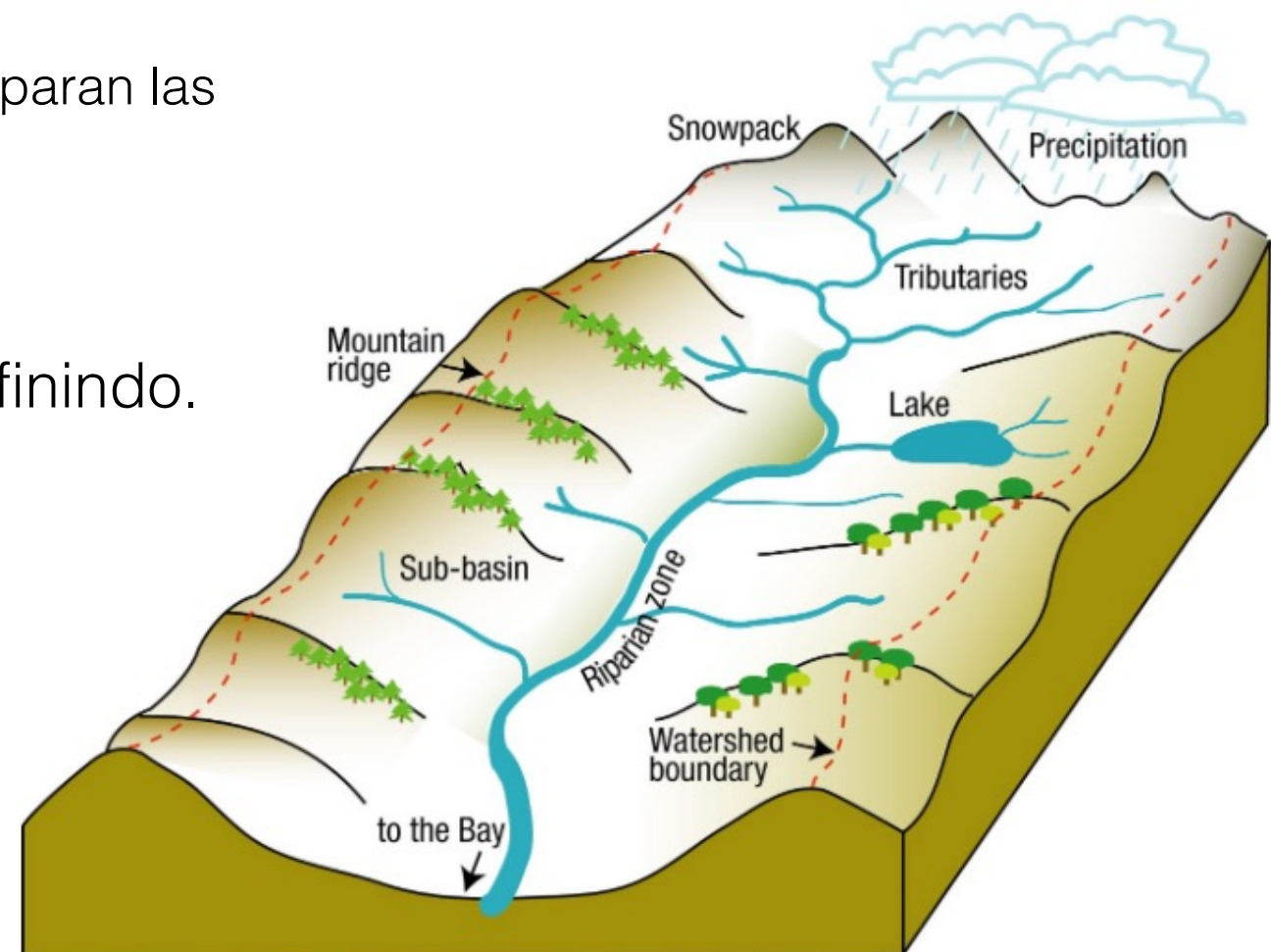
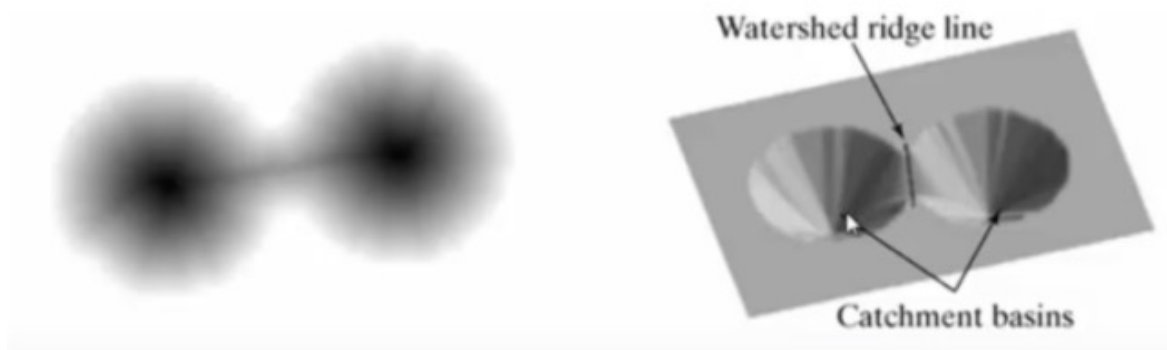
Crecimiento de regiones basado en la media del nivel de gris de la región



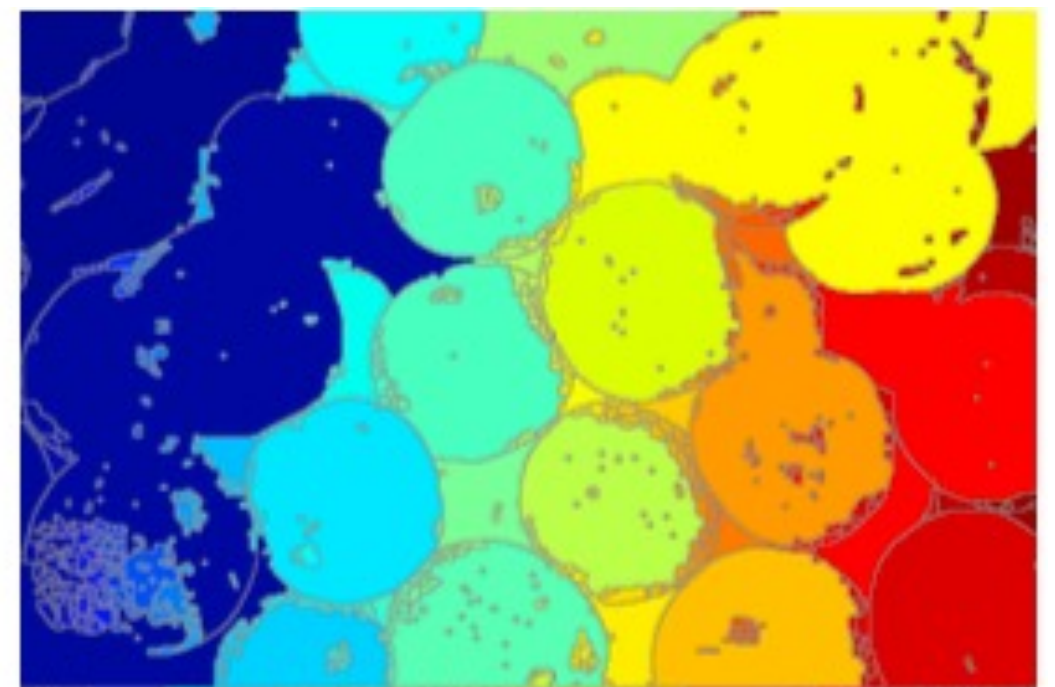
Watershed

- Idea:

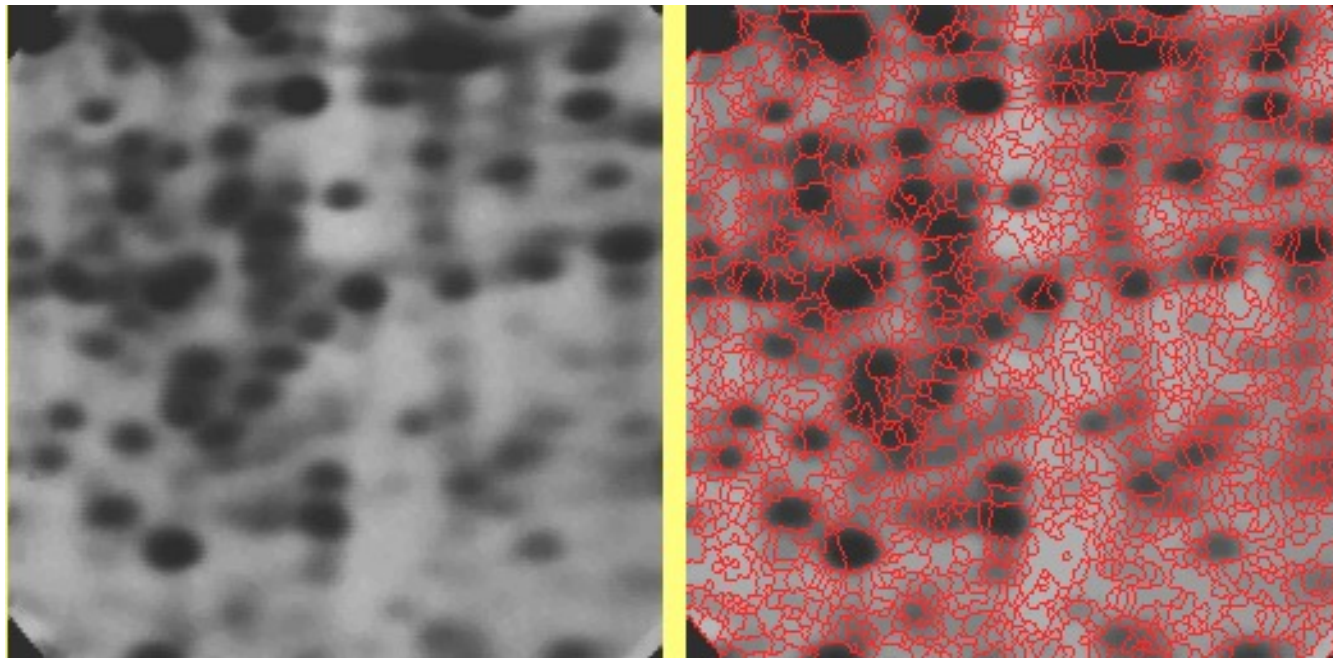
- En cada **región** hay una **cuenca de captación**.
- Watershed line: línea que divide varias cuencas de captación.
 - El objetivo es encontrar las líneas que separan las cuencas.
 - Equivalencia **altura** -> **nivel de gris**.
- El número de regiones no está predefiniendo.



Watershed

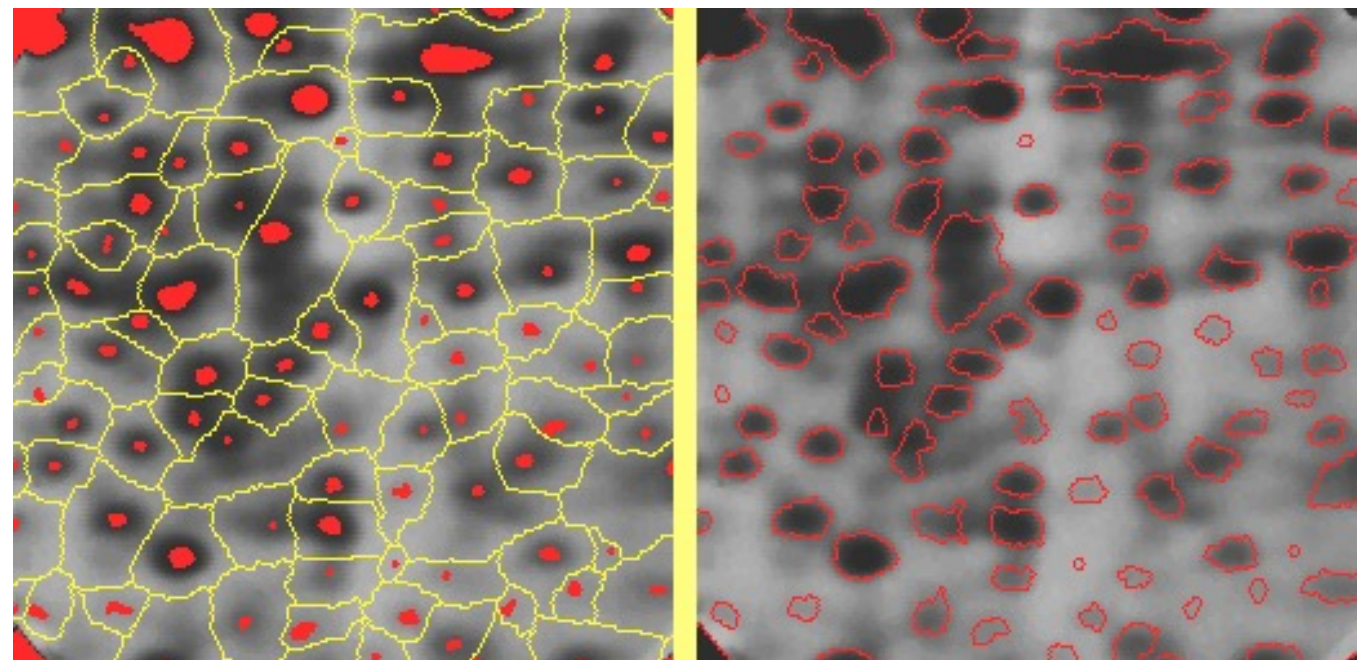


Watershed



Es fácil que se produzca una sobre-segmentación debida a ruido e irregularidades en el gradiente de intensidades de la imagen.

En esos casos es posible mejorar el resultado drásticamente si partimos de una imagen con las regiones parcialmente marcadas.



Técnicas orientadas a bordes

Causas físicas de los bordes

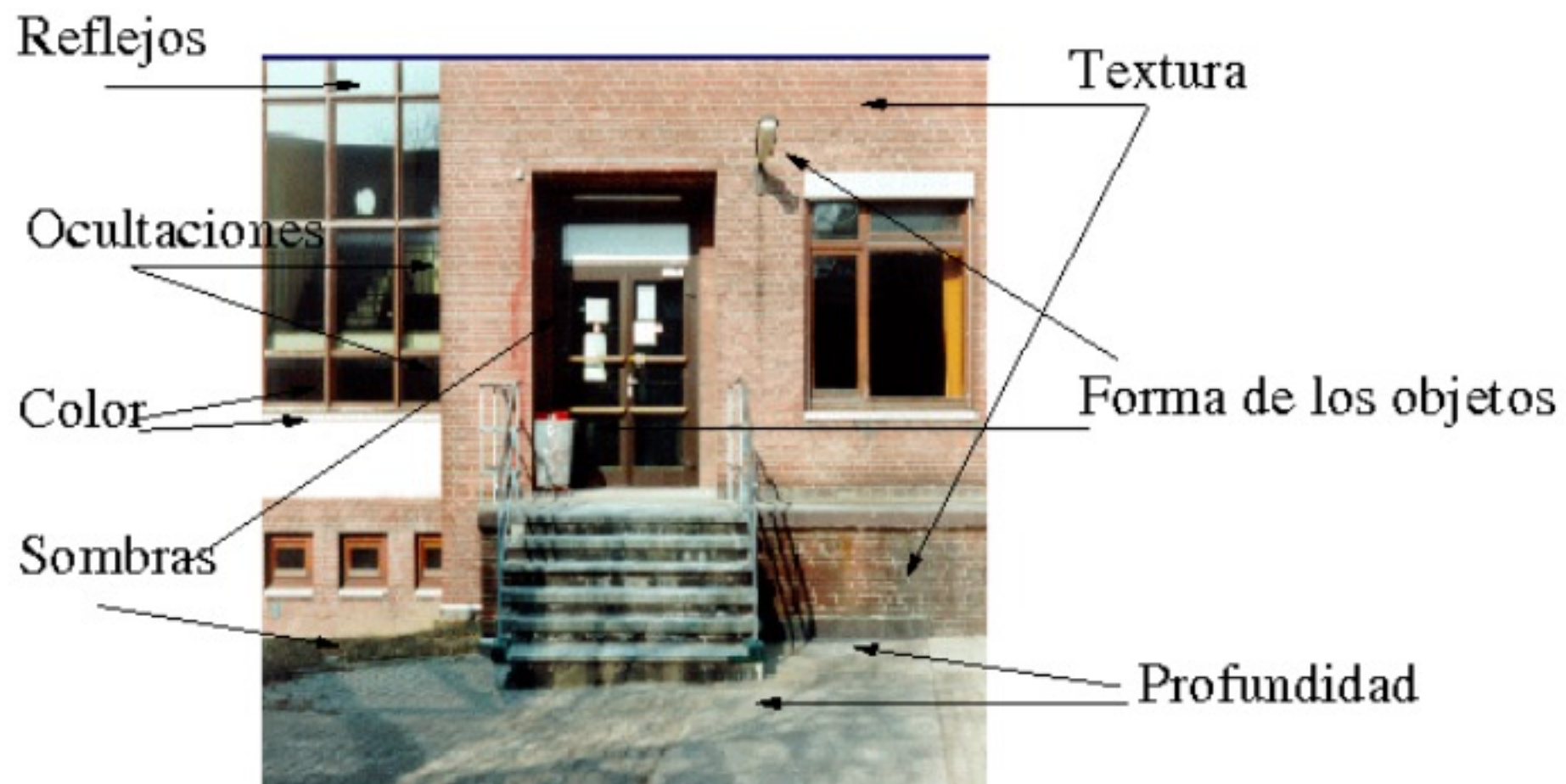


Imagen en color con bordes provocados por diferentes motivos.

Segmentación basada en bordes

- Una vez que se extraen los bordes de la imagen, el objetivo sería poder unirlos para obtener los objetos de interés (delimitados por **contornos**).
- Múltiples algoritmos y aproximaciones.

Contornos

- Los **bordes** son cambios de intensidad pronunciados.
- Un **contorno** es una curva de puntos sin huecos ni saltos. Tiene un principio y el final de la curva termina en ese principio.
- Podemos tratar de analizar todos los bordes detectados y comprobar si son contornos o no.



CONTORNO



NO CONTORNO

Contornos

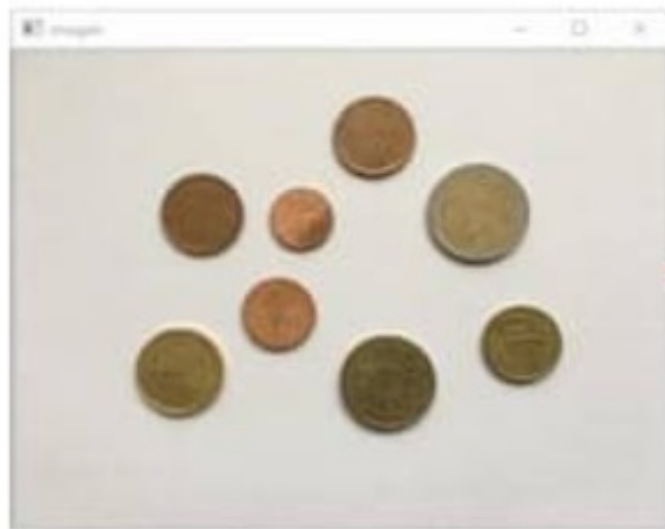


Imagen original



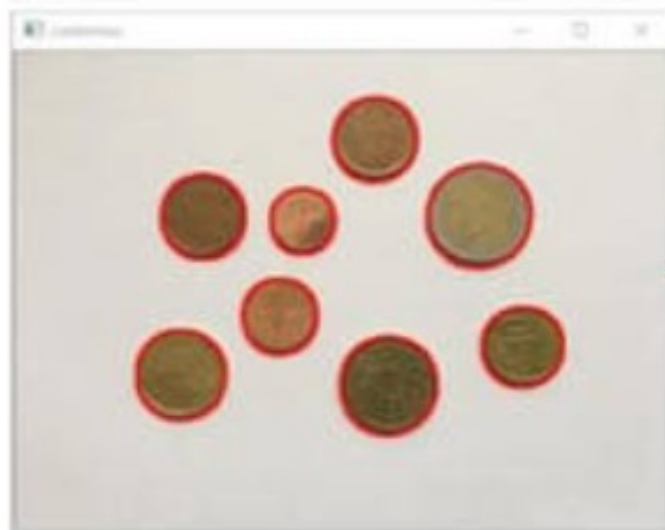
Imagen escala de grises



Filtrado de ruido



Detector de bordes Canny



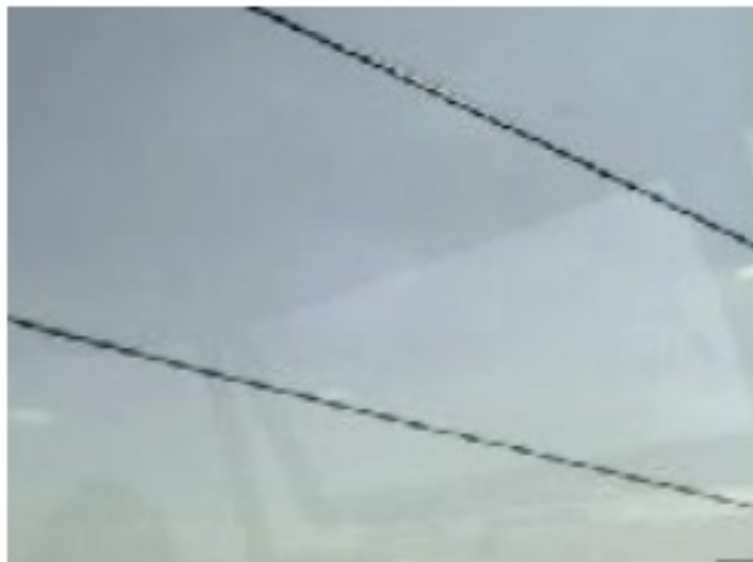
Detección de contornos

Transformada de Hough

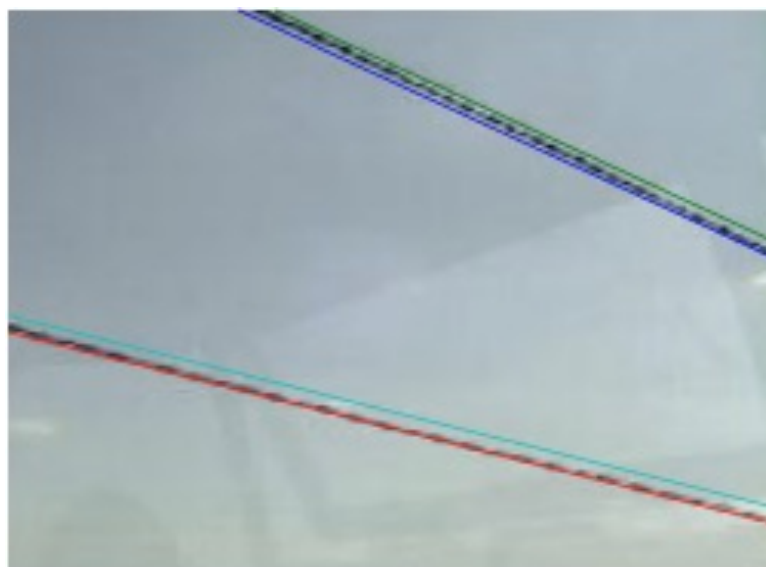
- Algoritmo para detectar patrones de puntos en **imágenes binarias**.
 - Líneas.
 - Curvas.
- Aplicación:
 - Detectar estructuras en una imagen cuando tenemos conocimiento a priori de lo que buscamos (rectas, círculos, etc.).
 - La entrada normalmente será una **imagen binarizada de bordes**.

Hough para líneas

Catenaria

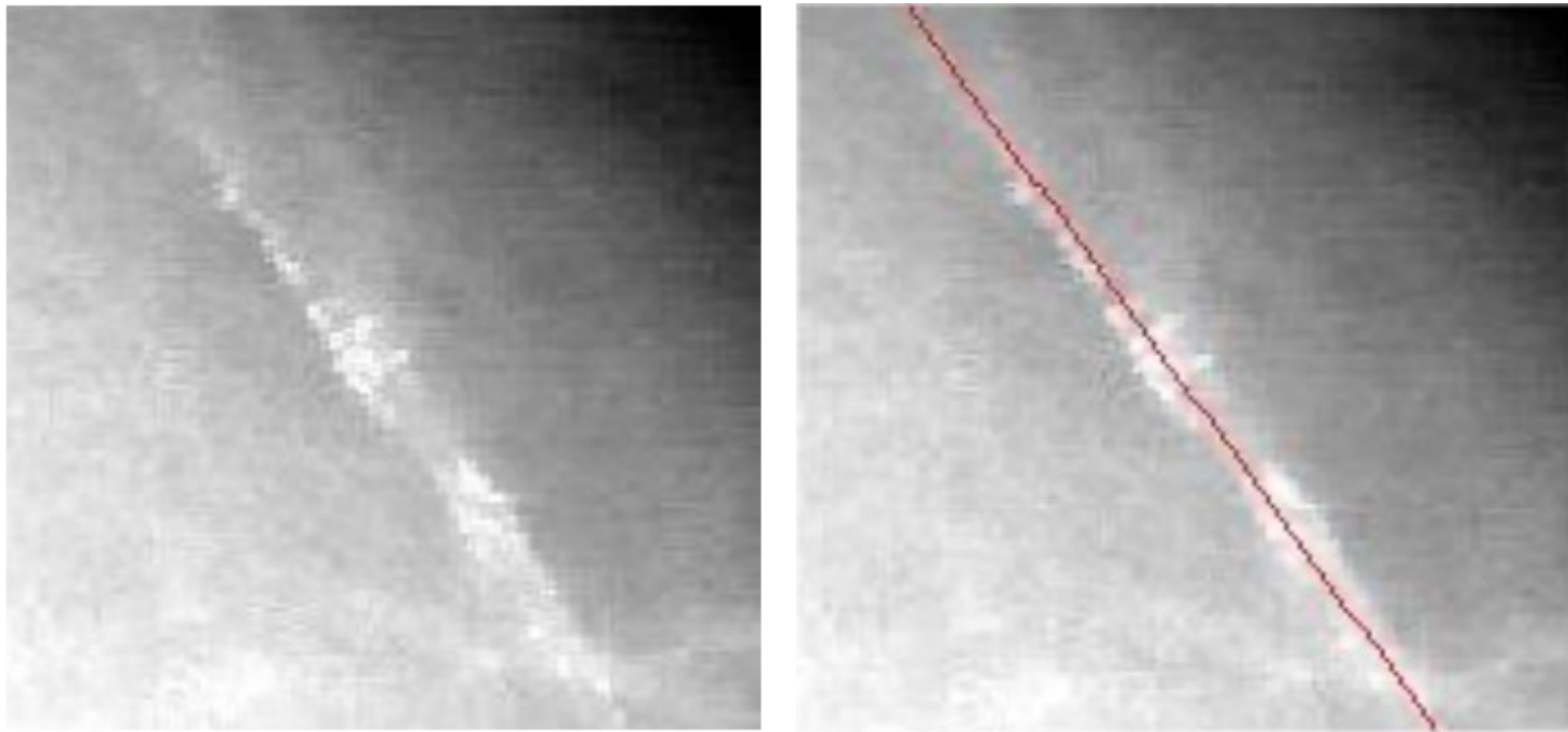


Detector de Canny



Detección de líneas superpuesta sobre imagen original

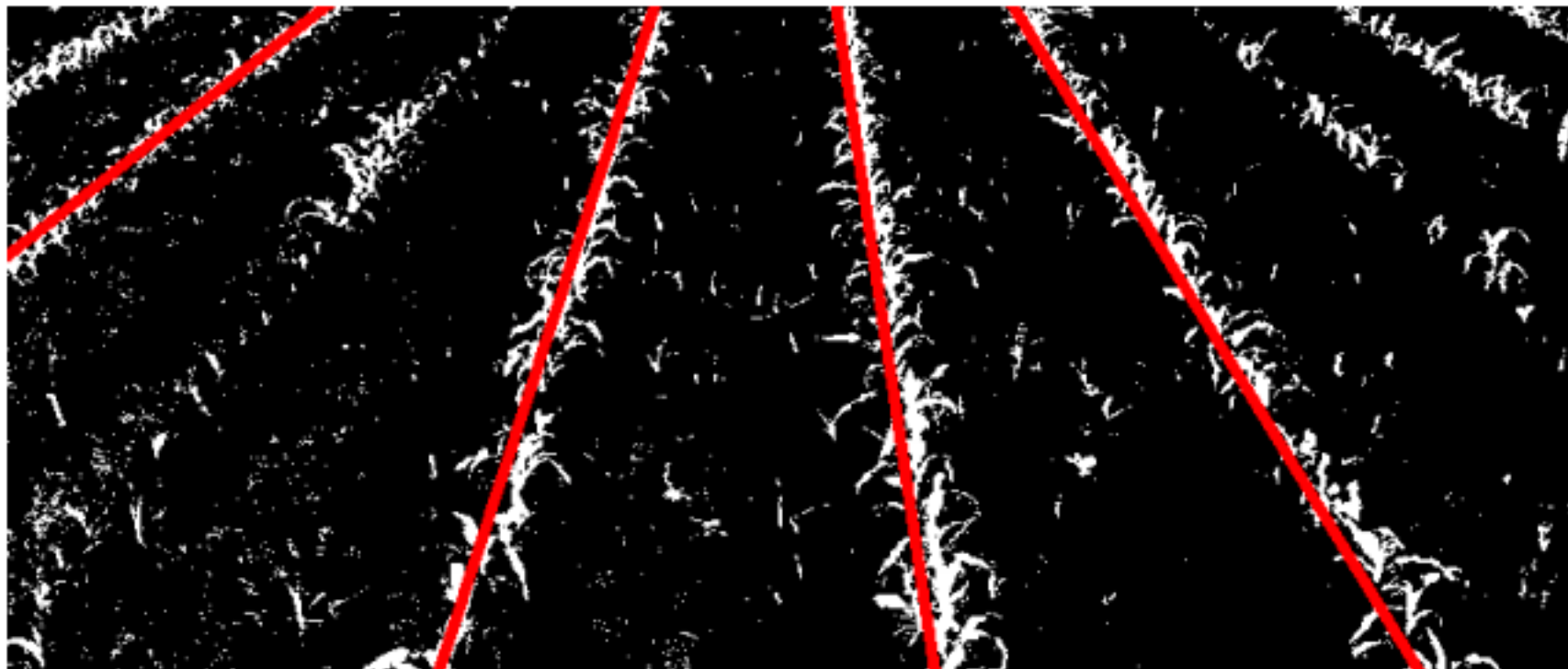
Hough para líneas



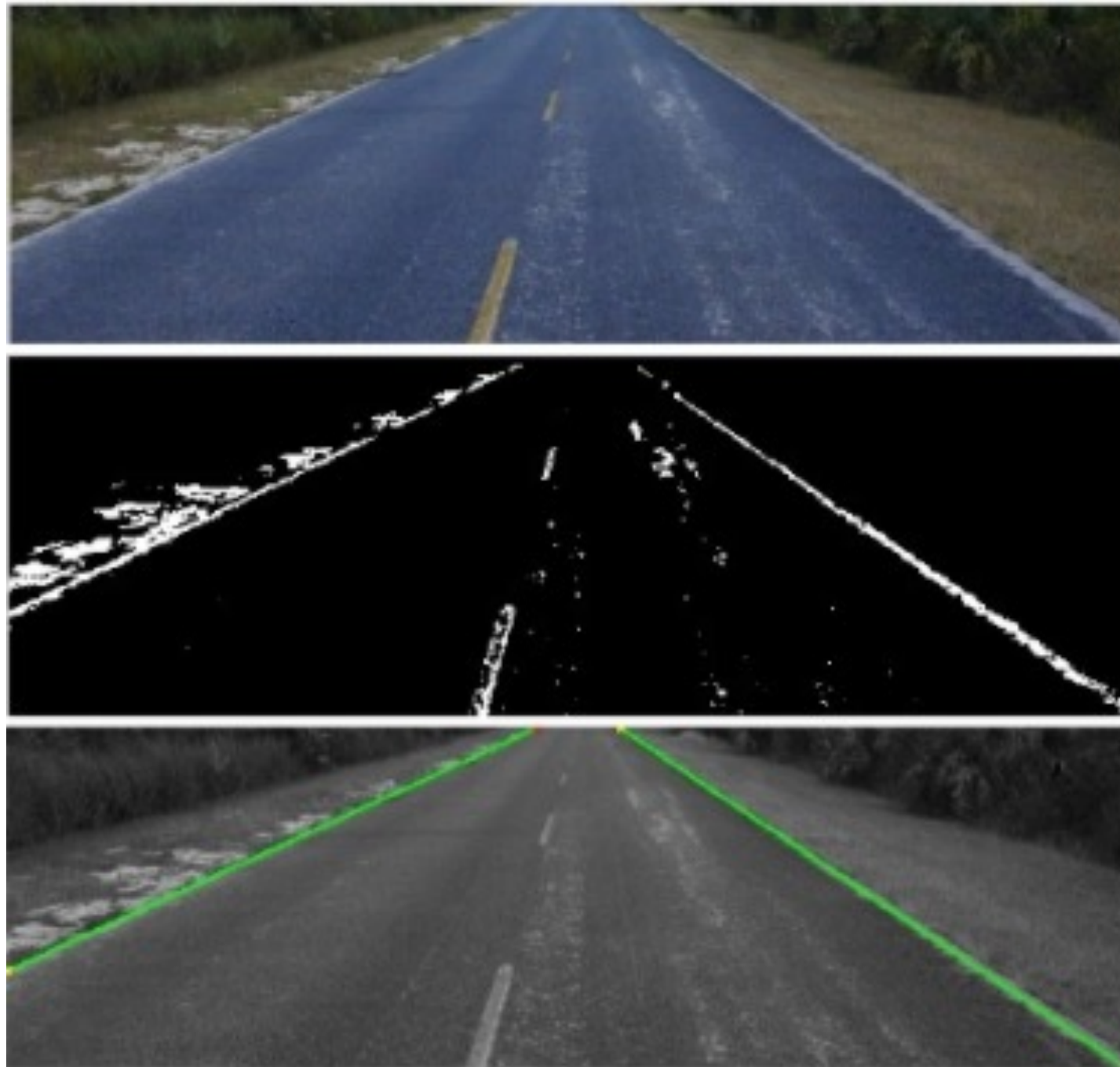
Segmentación de un vaso capilar

Hough para líneas

- Podemos establecer un umbral que controla qué se considera una línea o no en la imagen.



Hough para líneas



Hough para circunferencias

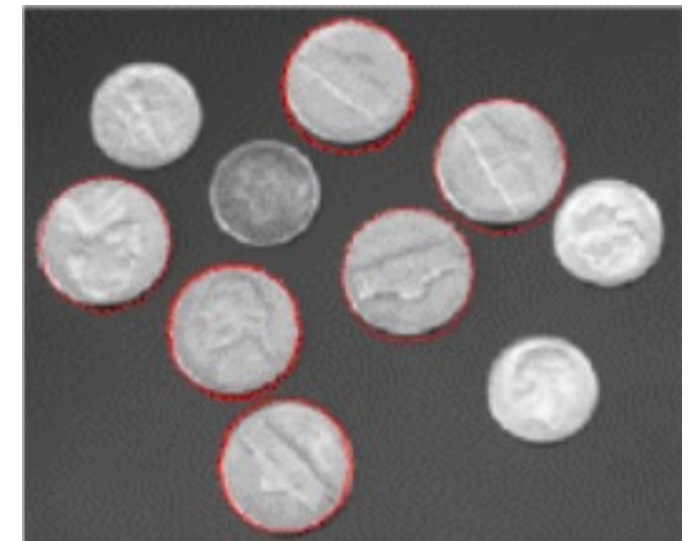
- Se puede mejorar el resultado si se acota el radio de la circunferencia (conocimiento a priori de los objetos que se van a capturar).



Imagen original



Detector de Canny

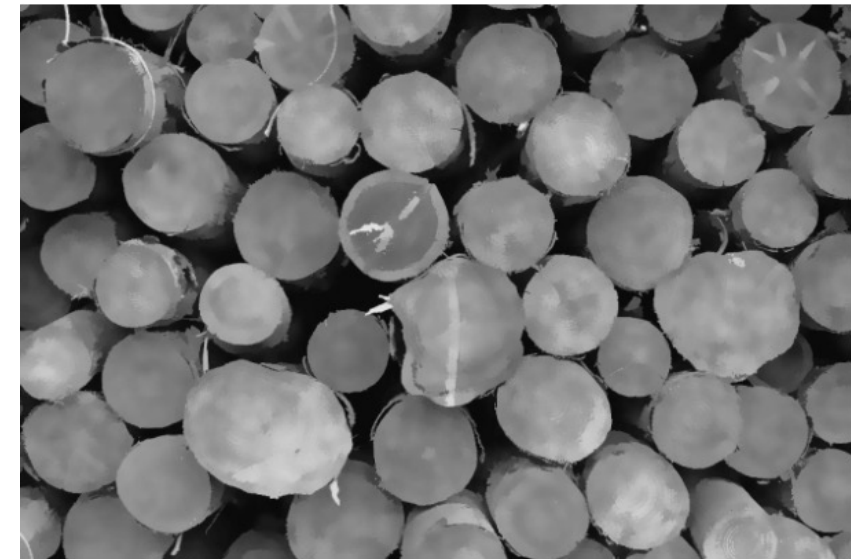


Segmentación de círculos
de 25 píxeles de radio

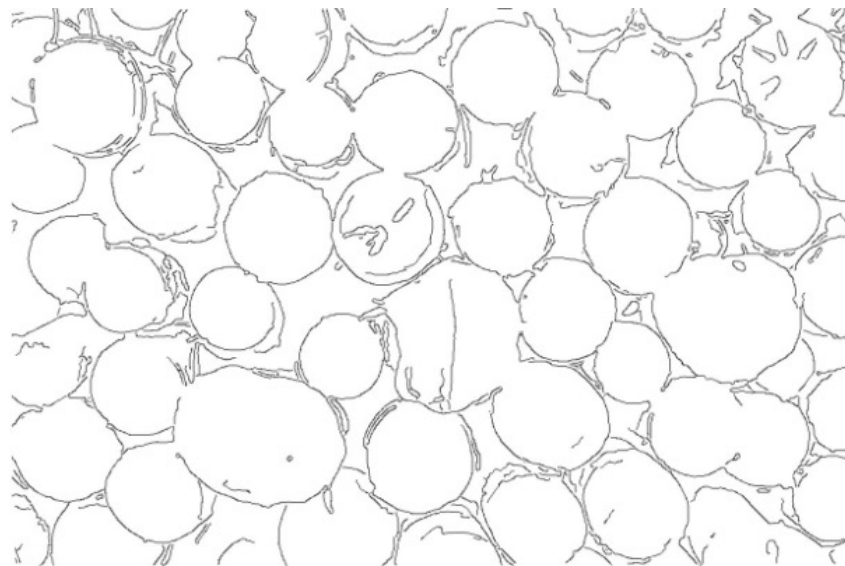
Hough para circunferencias



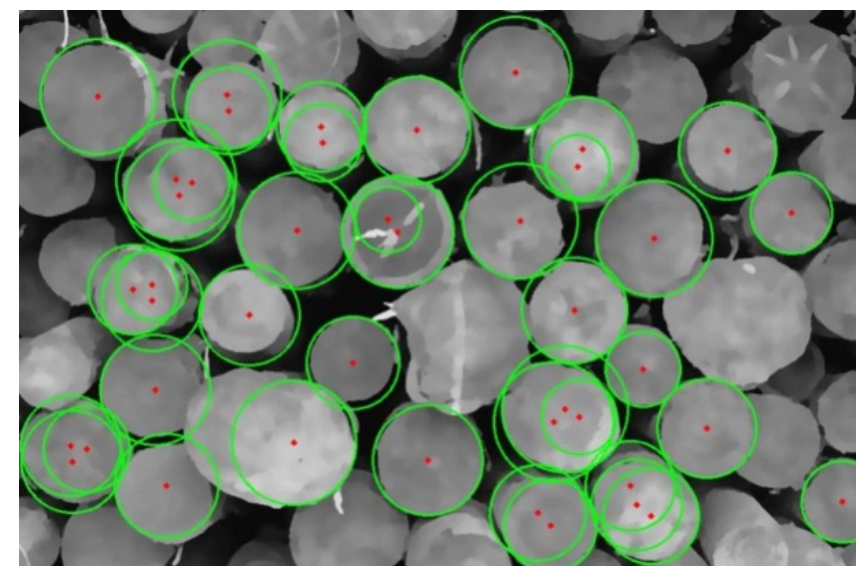
Imagen original



Suavizado y escala de grises



Detección de bordes con Canny



Detección de círculos con Hough

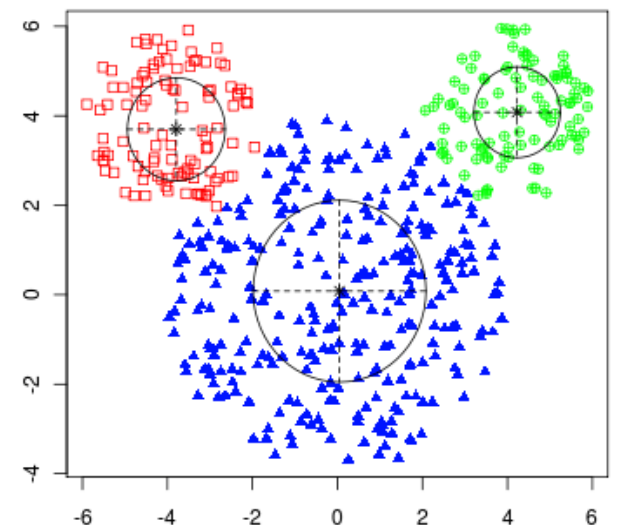
Técnicas de clustering

Clustering

- La **segmentación** es un caso especial de **clustering**: dado un conjunto de píxeles, queremos dividirlos en grupos (regiones).
- En base a características de las imágenes (ejemplo color y textura de un tomate).
- Un algoritmo agrupa automáticamente las observaciones en grupos (clusters) con valores similares de esas características. **Los grupos no están definidos previamente.**
- No se necesita una semilla a partir de la que empezar la segmentación.

Ejemplo: K-means

- Algoritmo de **clustering no supervisado** (no disponemos de información a priori de como deben agruparse los datos).
- Clasifica los datos de entrada en **K clusters** (k predefinido)
 - Los centroides iniciales de los k clusters se generan aleatoriamente.
- Intenta encontrar una partición de las muestras en K clusters, de forma que cada ejemplo pertenezca a uno de ellos, concretamente **a aquel cuyo centroide esté más cerca**.
- El **mejor valor de K** para que la clasificación separe lo mejor posible los ejemplos no se conoce a priori, y depende completamente de los datos con los que trabajemos.
- En el caso de **imágenes**: los datos de entrada se representan mediante vectores que pueden tener varias componentes:
 - Nivel de gris.
 - Componentes de color (en algún espacio de color, RGB/HSV).
 - Componentes de color y coordenadas de los píxeles.

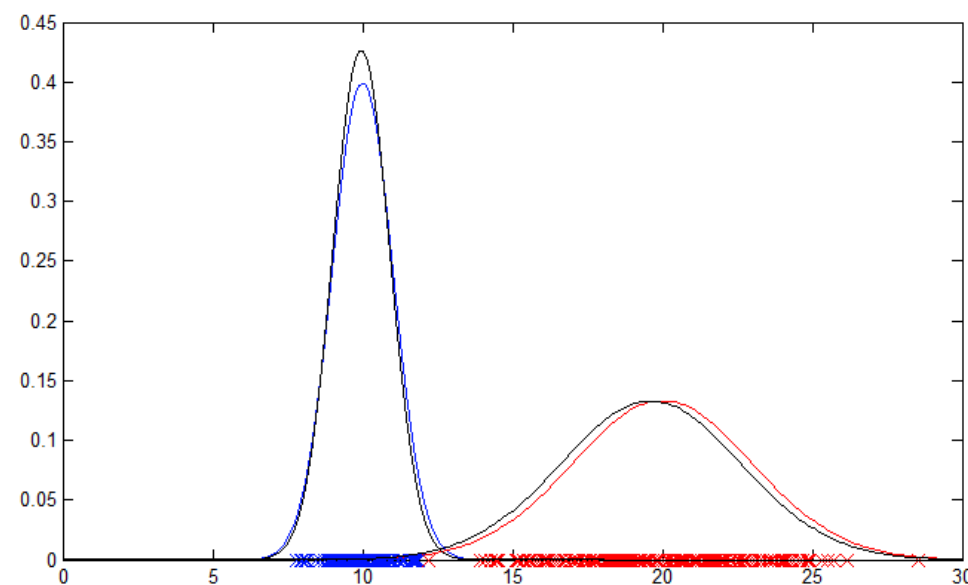


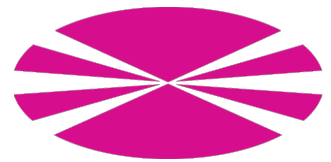
Ejemplo: K-means



Ejemplo: GMM

- Gaussian Mixture Model (GMM)
 - Algoritmo no supervisado que modela cada cluster como una **distribución gaussiana con una media y varianza específica**, en base a un conjunto de datos.
 - Para cada punto tenemos una probabilidad de que pertenezca a cada cluster.
 - Expectation Maximization: técnica que se utiliza para encontrar de forma automática e iterativa los clusters.
 - <https://www.youtube.com/watch?v=kkAirywakmk>
 - <https://scikit-learn.org/stable/modules/generated/sklearn.mixture.GaussianMixture.html>





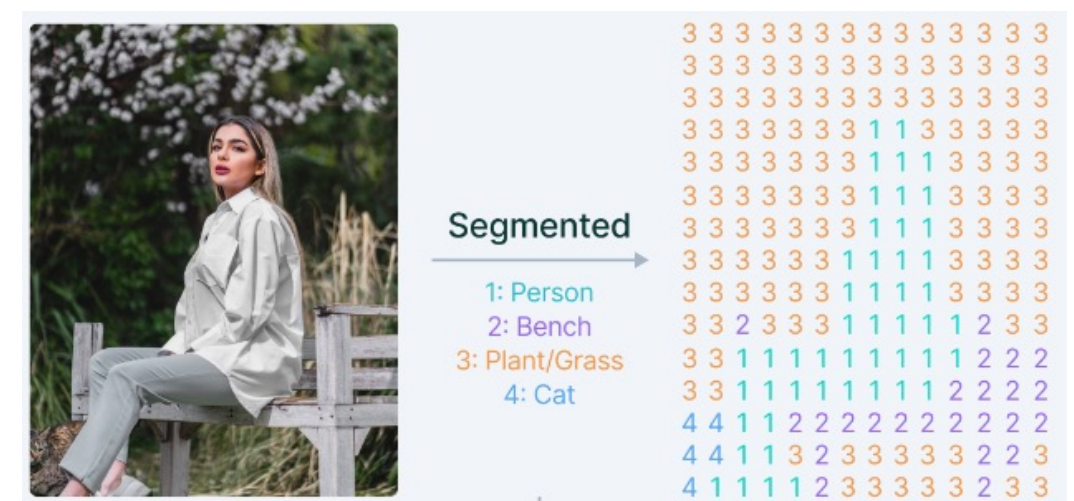
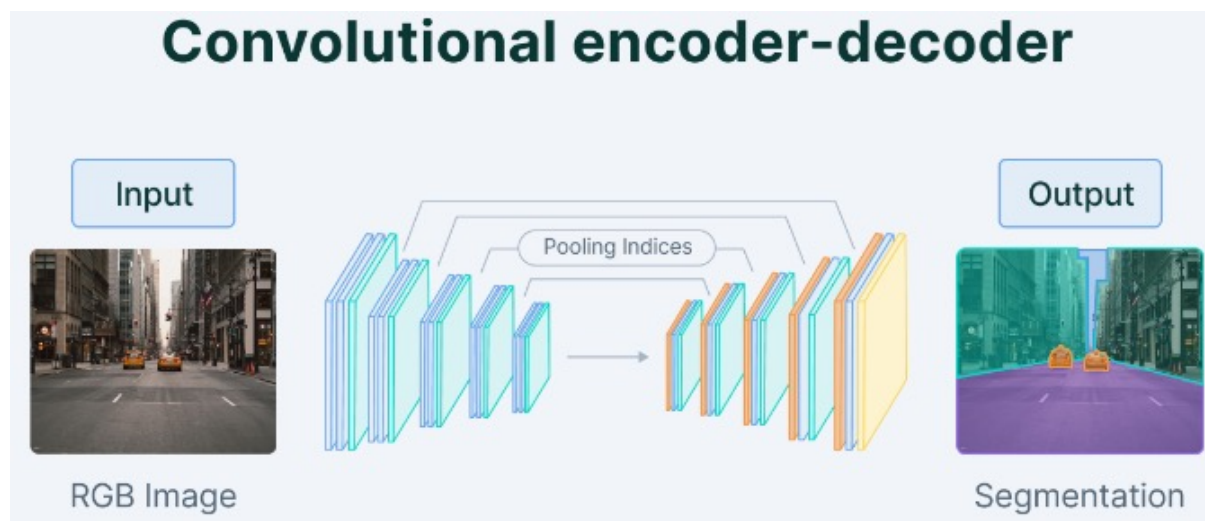
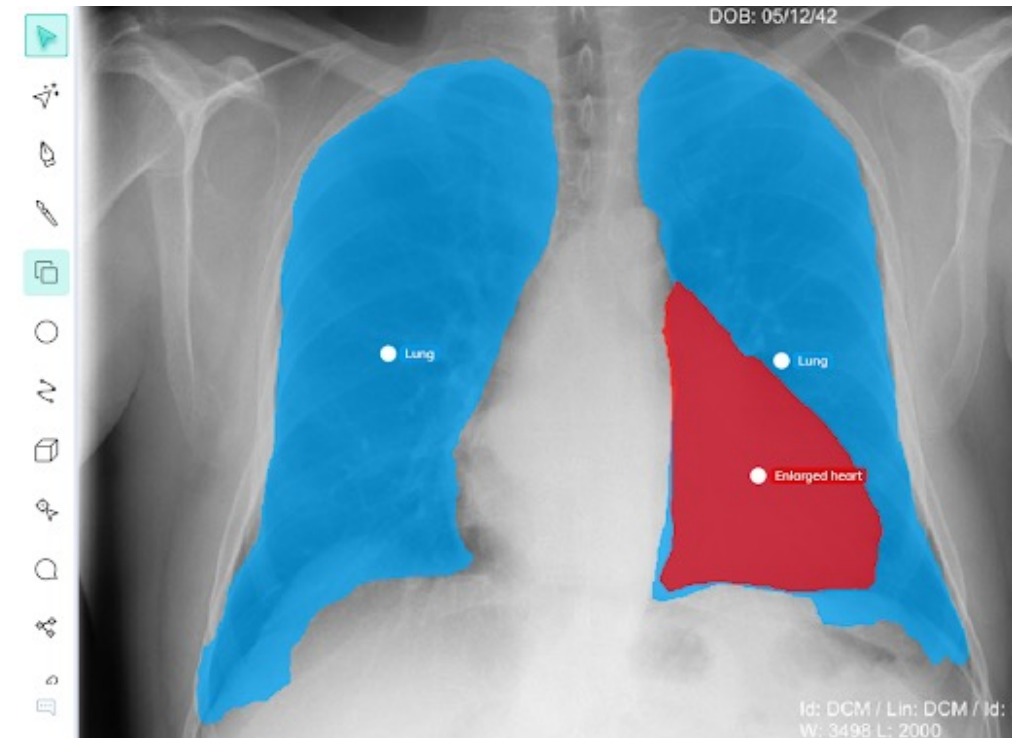
UNIVERSIDADE DA CORUÑA

Técnicas basadas en deep-learning

Segmentación con deep-learning

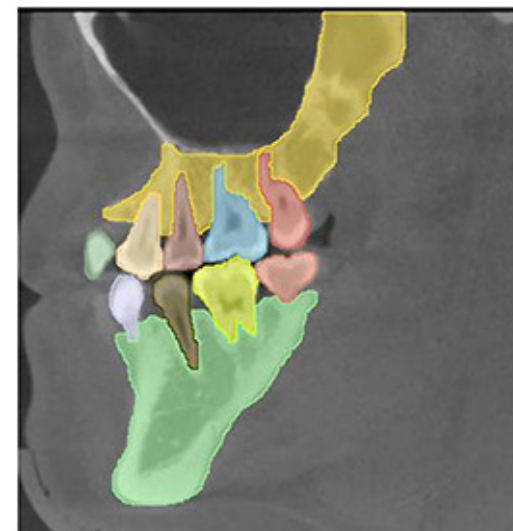
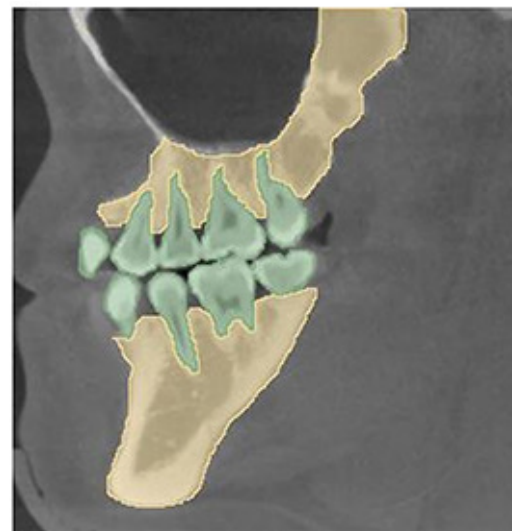
- Generación automática de mapas de píxeles en base al aprendizaje realizado con un conjunto etiquetado de imágenes de entrada sobre una red de neuronas.
- Los métodos de segmentación basados en aprendizaje supervisado necesitan un conjunto de datos (imágenes) previamente etiquetado.
- Existen herramientas software para asistir el proceso manual de etiquetado.
 - Marcado de bounding-boxes de regiones.
 - Generación automática de la región probable dentro de un bounding-box.
 - Refinamiento manual de la región generada.

Etiquetado de imágenes con herramienta software



Segmentación con deep-learning

- Las técnicas basadas en *deep learning* han supuesto también una revolución en la segmentación de imagen.
- Es frecuente que sean este tipo de técnicas las que obtengan mejores resultados sobre conjuntos de datos de referencia.
- Es necesario reunir un **conjunto de datos de calidad** para entrenamiento y validación.
- En general, es necesario disponer del *ground-truth* para cada muestra de los datos de entrenamiento y validación.
- La aproximación varía si el problema requiere de una **segmentación individualizada de instancias**.

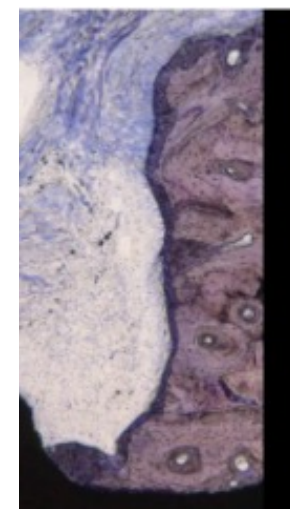


segmentación semántica

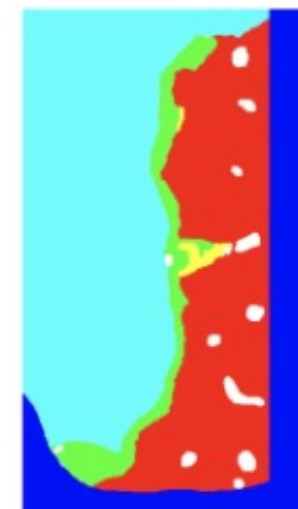
segmentación de instancias

Método

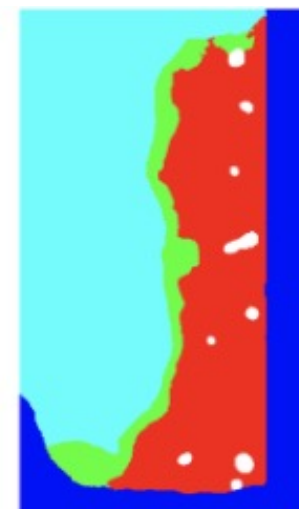
recorte original



objetivo



predicción



1. Reunir **conjunto de datos** inicial.

- Habitualmente se necesitan **expertos** en el ámbito del problema.
- Necesitamos tener todas **clases de regiones** que queremos segmentar presentes en las imágenes del conjunto de datos (adecuadamente balanceadas).
- Es aconsejable comparar imágenes de **ground-truth** generadas por **más de un experto**.
 - Nos dará una idea del sesgo que introduce el propio experto al hacer manualmente la segmentación.
 - Nos permite conocer si el error que comete nuestro algoritmo está dentro de los márgenes habituales debidos al sesgo del experto.

2. Seleccionar **modelos** de deep learning a utilizar.

- Es aconsejable empezar probando modelos de redes no excesivamente grandes, ya que si dan buenos resultados para nuestro problema nos permitirá hacer entrenamientos y pruebas con mayor rapidez. Ejemplos: SegNet, HRNet, U-Net.
- Arquitectura: dimensiones de imágenes de entrada (tamaño, canales), número de clases de salida, función de coste, optimizador, velocidad de aprendizaje, arquitectura interna de la red, ...

3. Iterativamente **validar** los resultados con los expertos y tratar de **refinar** la solución.

4. Considerar el uso de **técnicas de data augmentation** para generar un mayor número de muestras de entrenamiento (y con mayor variabilidad).

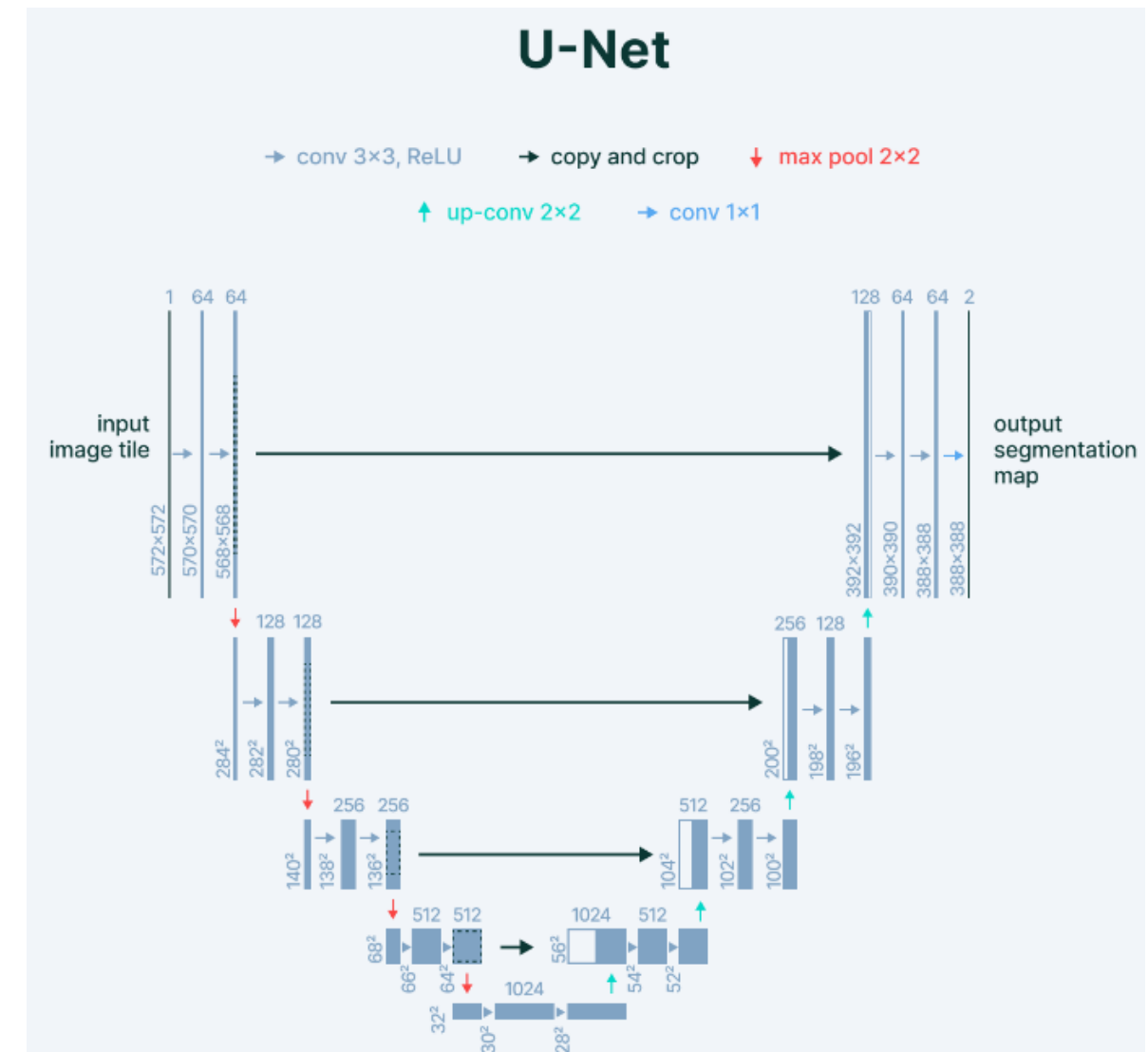
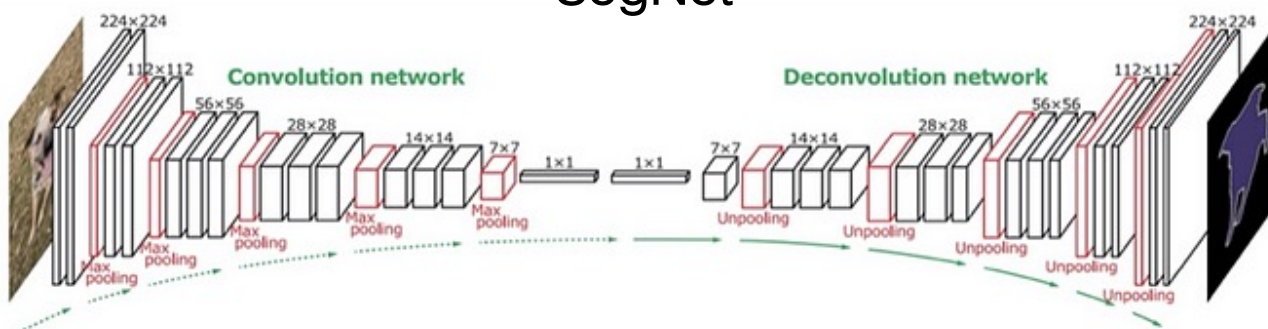
5. Si las aproximaciones probadas no consiguen buenos resultados, considerar **dividir el problema en varios casos** (si es posible) y utilizar distintos modelos para realizar la segmentación de cada caso de forma independiente.

- Requiere que el sistema final pueda conocer el caso concreto que tiene que resolver.
- Ejemplo: varios casos para la segmentación de diente y hueso según la especie animal de que se trate.

Modelos

- Existen múltiples modelos de redes de deep learning que se aplican en tareas de segmentación.
- Algunos de los más populares están en la categoría de redes tipo autoencoder (Ejemplo: SegNet).
- Una combinación de bloques convolucionales y submuestreo para *comprimir* la información de entrada, y a continuación, el un decodificador que reconstruye la información de entrada para generar un mapa de las regiones de la entrada según las clases definidas.
- U-Net: introduce conexiones directas entre capas intermedias del encoder y capas intermedias del decoder. Muy utilizada en imagen médica.
- DeepLab (<https://github.com/tensorflow/models/tree/master/research/deeplab>).
- Transformers (<https://keras.io/examples/vision/segformer/>).

SegNet



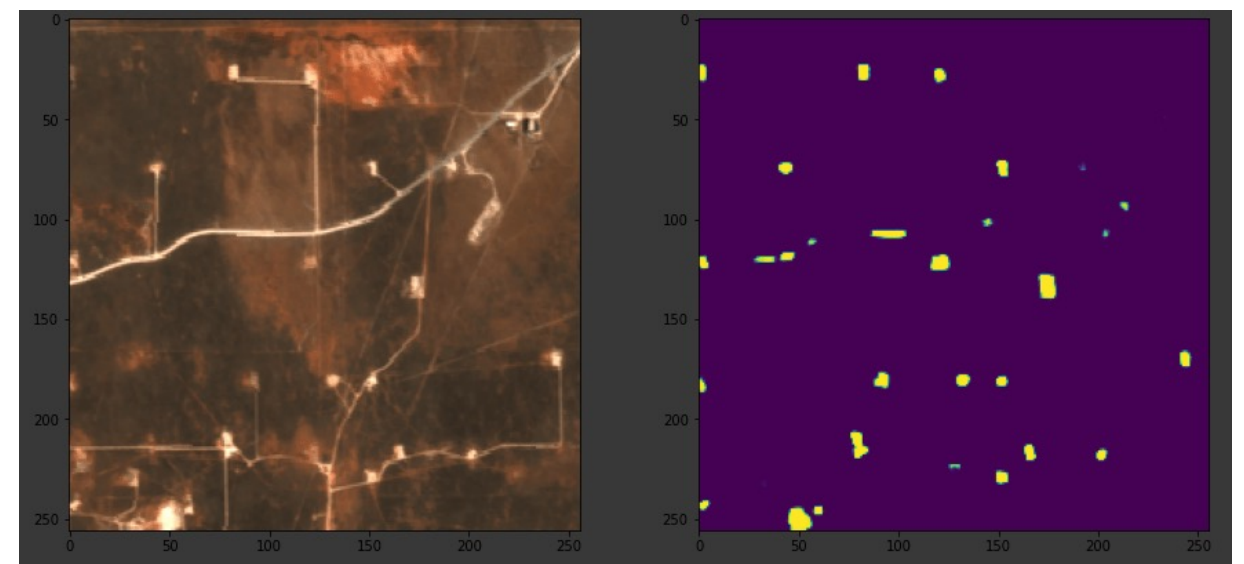
Validación de la segmentación

Validación

- Rendimiento del algoritmo:
 - Tiempo necesario para ejecutar el algoritmo.
 - Tasa de acierto (cuántas veces el algoritmo devuelve una decisión correcta).
- Problemas para medir el rendimiento del algoritmo:
 - Conjuntos de datos pequeños.
 - Conjuntos de datos no representativos del problema.
 - Métricas no adecuadas.
 - Metodología de test no adecuada.

Validación de la segmentación

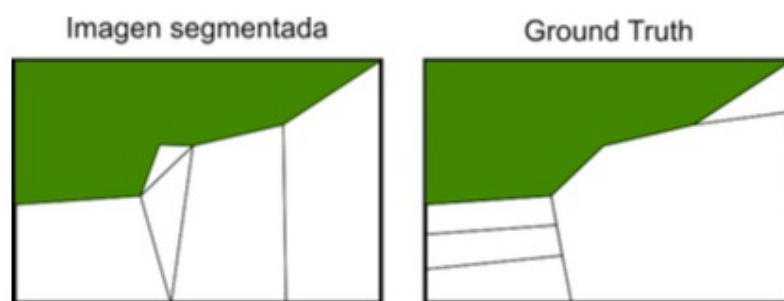
- Tipos habituales de problemas:
 - Extraer todas las regiones básicas de la imagen.
 - Identificar únicamente las instancias de un objeto (o anomalía) en la imagen.



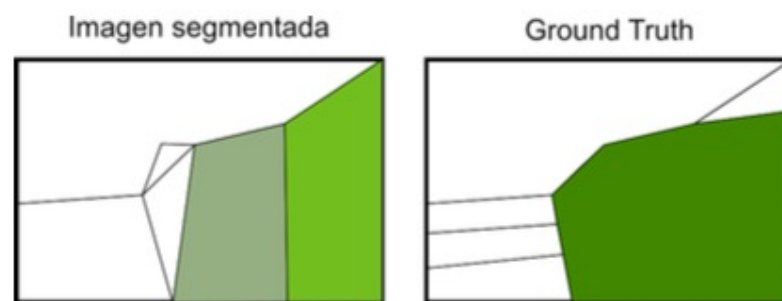
Identificación de zonas de perforaciones para extracción de gas o petróleo (imagen: Brendan Jarrell)

Métricas

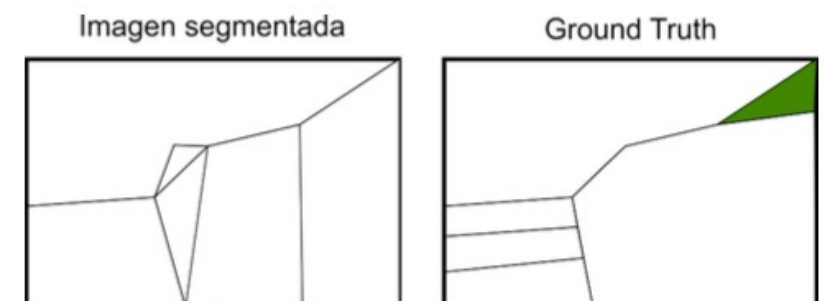
- Extraer todas las regiones:
 - Porcentaje de solapamiento entre regiones segmentadas y de *ground truth*.
 - Distancia entre los bordes de las regiones.
 - Error cuadrático medio entre los bordes de las regiones.



$\text{Área}(\text{RegSeg} \cap \text{RegGT}) > k \cdot \text{Área}(\text{RegGT})$



Sobre-segmentación



Región perdida

Métricas

- Identificar todas las instancias de un objeto:
 - FROC (Free-Response ROC)
 - ROC: proporción de verdaderos frente a la proporción de falsos positivos (sensibilidad frente a 1-especificidad).
 - Un algoritmo de detección sobre imagen puede obtener un número arbitrario de falsas detecciones (número máximo de falsos positivos desconocido).
 - FROC: en el eje X se representa la media de falsos positivos por imagen.

