

# Procesamiento de Imágenes 2024

## Trabajo Práctico Especial

Camila Garcia Petiet

Giuliana Raffin



# Índice

Introducción	2
Método 1: Flood fill, operador morfológico de opening y threshold Gaussiano	3
Método 2: Region Growing y borde de Laplace	4
Método 3: Region Growing y borde de Prewitt	5
Pruebas descartadas	6
Medición de la efectividad de la segmentación	7
Conclusión	9
Bibliografía	10

# Introducción

En el ámbito de la radiología y el diagnóstico por imágenes, la segmentación y el análisis de las imágenes médicas es crucial para la detección, evaluación y seguimiento de diversas patologías. Este trabajo está centrado en el análisis de las radiografías de la columna vertebral, proporcionadas por la cátedra, y la detección de los bordes de cada vértebra.

Se implementaron tres métodos para llegar a una imagen objetivo. En este informe se enuncia en qué consiste cada método y cuáles fueron los pasos que se siguieron para implementarlos. Además, se especifican algunos procesamiento que, si bien no llegaron a formar parte de un método, ayudaron a la hora de aprender y tomar decisiones.

El desarrollo del trabajo requirió el uso de varias herramientas y tecnologías. Se utilizó un notebook de [Google Colab](#) donde con python se implementó el código. Además, se utilizaron las librerías Numpy, Matplotlib y OpenCV para el manejo de las imágenes, gráficos, vectores y matrices. Por último, se utilizó Pyradiomics para comparar diversos aspectos de las imágenes procesadas y evaluar la efectividad de los métodos entre sí.

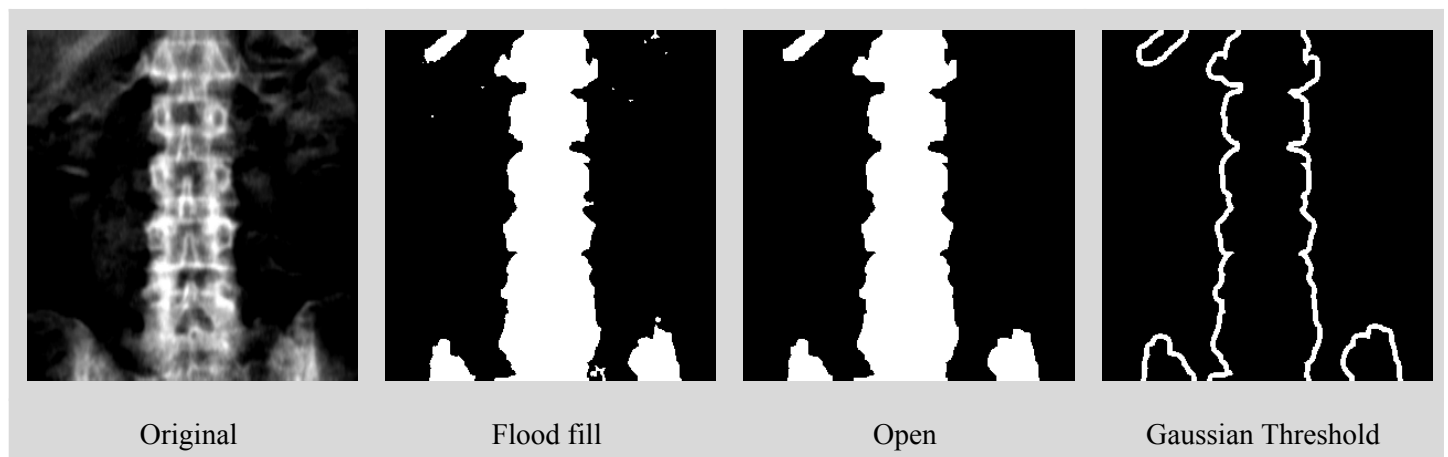
## Método 1: Flood fill, operador morfológico de opening y threshold Gaussiano

Como primer método de procesamiento de imágenes, se optó por implementar la función flood fill en conjunto con el operador morfológico open y threshold Gaussiano. La ejecución de flood fill consiste en, a partir de una imagen en formato tif, aplicar un algoritmo de relleno para separar la columna del fondo. Para ello, se aplicaron diferentes máscaras que permitieron separar el lado izquierdo de la imagen del lado derecho. Con el resultado obtenido, se separaron los huecos internos de la columna del resto de la misma y se invirtió la imagen. Hecho esto, como paso final, se superpusieron la imagen utilizada con la imagen invertida para rellenar los agujeros internos de la columna.

Seguido de esto, se implementó el operador morfológico de opening para deshacerse de información innecesaria. Esto consiste en aplicar una erosión y luego una dilatación, y así se eliminó ruido que se encontraba por fuera de la columna y se consideraba información irrelevante. Cabe destacar que durante este proceso no solo se pierde información irrelevante, sino que también provoca una cierta pérdida de información sobre la estructura que se quiere analizar.

Por último se aplicó un algoritmo de threshold adaptativo Gaussiano de la librería OpenCV. El valor threshold o umbral  $T(x,y)$  es una suma ponderada (correlación cruzada con una ventana gaussiana) de la vecindad  $\text{blockSize} \times \text{blockSize}$  de  $(x,y)$  menos  $C$ . El sigma (desviación estándar) predeterminado se utiliza para el  $\text{blockSize}$  especificado.

En las imágenes debajo se muestra paso a paso el proceso de aplicar los tres algoritmos mencionados anteriormente y su resultado final.

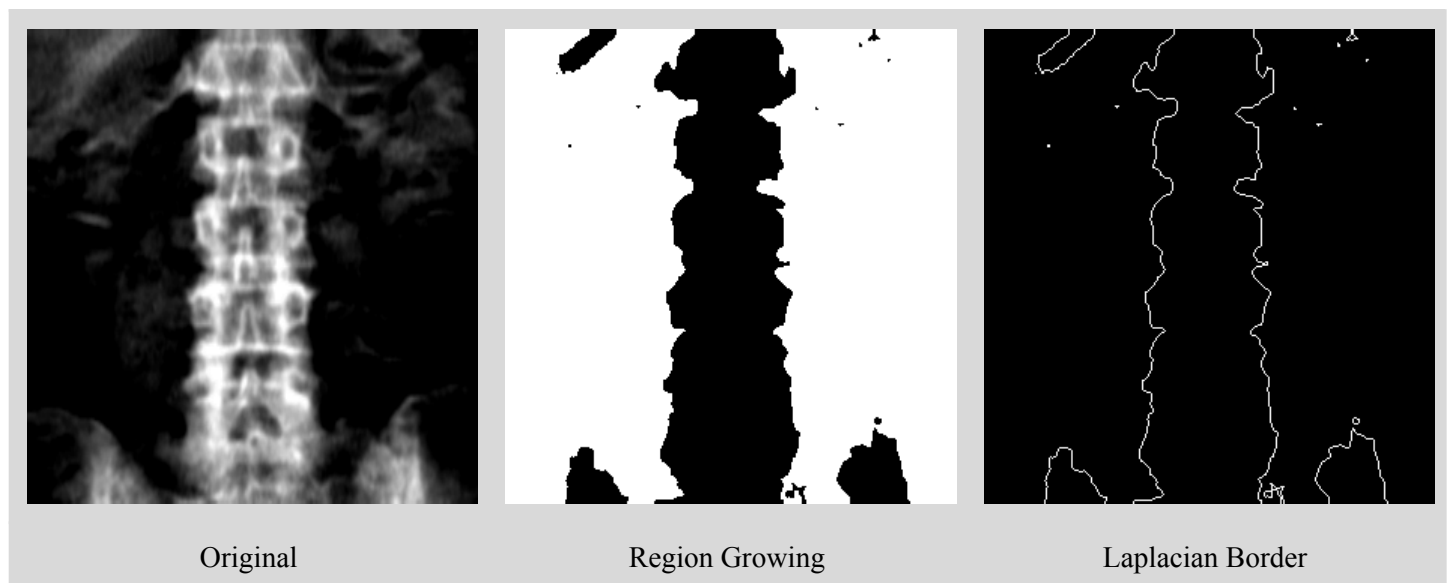


## Método 2: Region Growing y borde de Laplace

El segundo método de segmentación utilizado fue una combinación entre Region Growing y borde de Laplace. Para el Region Growing se recorrió la imagen y, de acuerdo a un umbral y una semilla elegida, se estableció que, si el valor del píxel vecino es suficientemente similar al seleccionado, son “fusionados” en una misma región.

Una vez procesada la imagen original de esta manera, al resultado se le aplicó un borde Laplaciano. El mismo consiste en, haciendo uso de la librería OpenCV, la suma de las segundas derivadas de  $x$  e  $y$ . Este cálculo se realiza cuando  $ksize > 1$ , y cuando  $ksize == 1$ , el laplaciano se calcula filtrando la imagen con una apertura de  $3 \times 3$  establecida por el algoritmo.

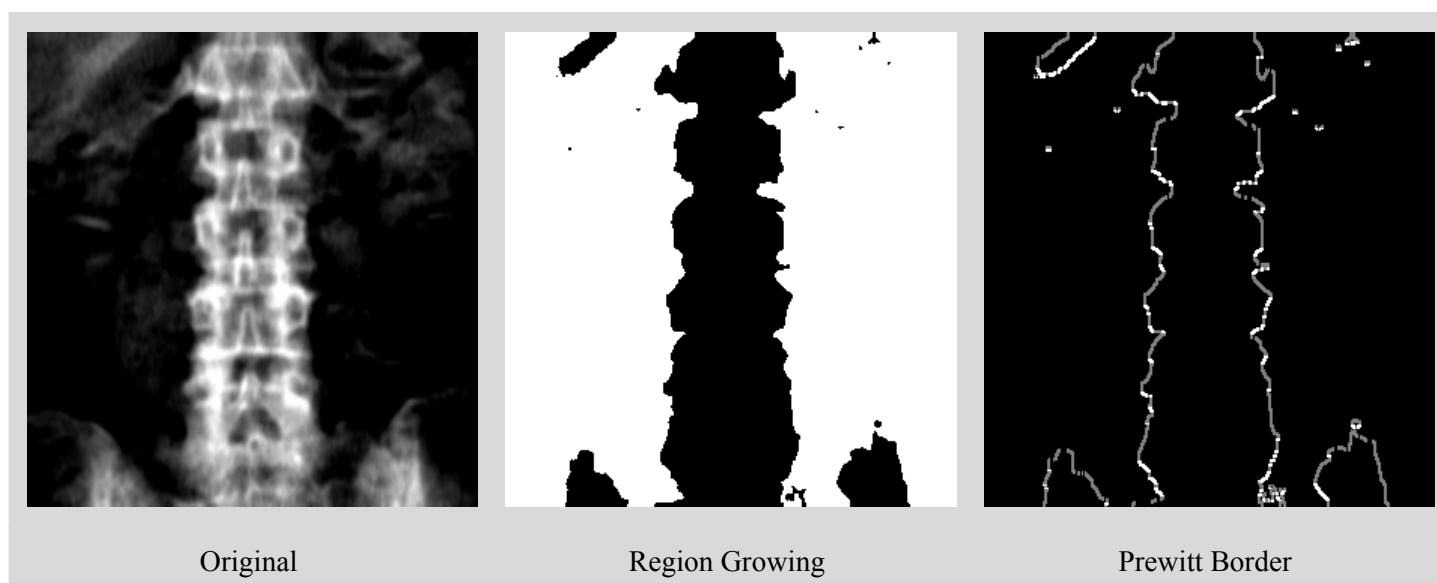
En las imágenes debajo se muestra paso a paso el proceso de aplicar los dos algoritmos mencionados anteriormente en forma secuencial.



## Método 3: Region Growing y borde de Prewitt

Por último, el tercer método utilizado combina Region Growing y la detección de bordes con Prewitt. Como se enunció en el Método 2, la técnica de segmentación Region Growing agrupa píxeles o subregiones mediante el uso de píxeles semilla. Una vez se seleccionan las semillas, se define un criterio de selección para determinar si el píxel vecino pertenece o no a la región. En caso de que cumpla con el criterio, se incluye en la región y se considera como una nueva semilla para seguir iterando.

Posteriormente, se realiza el filtrado con la técnica de Prewitt. Es un operador de detección de bordes simple y eficaz que calcula la diferencia de intensidad de píxeles vecinos para resaltar las áreas de cambio abrupto en la intensidad, que generalmente corresponden a los bordes de objetos en la imagen. Como primer paso, se definieron tres máscaras, para detectar los bordes verticales, horizontales y una similar a la horizontal pero con mayor peso en el valor de los píxeles, dando énfasis a los bordes detectados. Luego, se convoluciona cada máscara con la imagen para, seguidamente, combinarlas y obtener la imagen final.

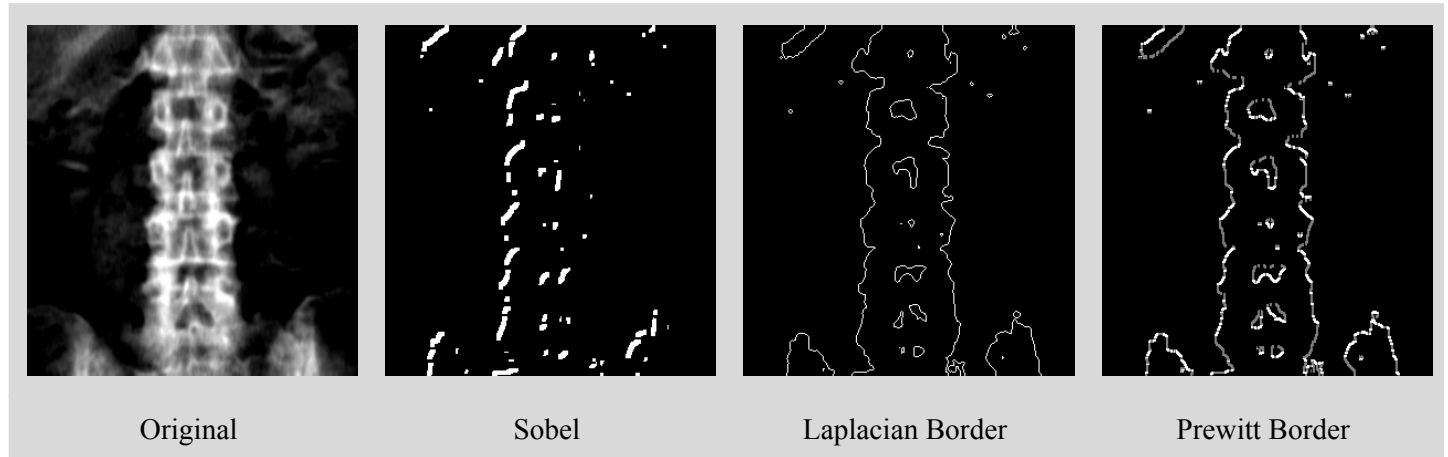


## Pruebas descartadas

Antes de seleccionar los tres métodos anteriores, se realizaron otras pruebas aplicando diferentes algoritmos. Se los mencionara a continuación a modo de ilustración para comparar los resultados con los algoritmos elegidos. El algoritmo de Sobel fue descartado ya que no fue una técnica que permitió detectar bordes de manera continua y discernible.

Luego, tanto Laplace y Prewitt por sí solos, es decir sin realizar region growing previamente, no fueron tomados en cuenta debido a que en ambos permanecía la parte interna de las vértebras. Sin embargo, cuando fueron utilizados en conjunto con la técnica de crecimiento de regiones, se obtuvieron resultados favorables.

En las imágenes debajo se muestran las imágenes resultantes de aplicar los algoritmos anteriormente mencionados.



## Medición de la efectividad de la segmentación

A la hora de medir la efectividad de las diferentes segmentaciones, se utilizó la biblioteca open source PyRadiomics. Dicha librería se utiliza para la extracción de características radiómicas de imágenes médicas que sirven para caracterizar la imagen, evaluar un tratamiento y predecir un resultado clínico. Para este trabajo, se analizaron las características de las imágenes obtenidas a partir de los tres métodos enunciados para poder evaluar y comparar las diferencias y similitudes.

En la tabla a continuación se encuentran los valores de diferentes medidas para parámetros de las imágenes resultantes. En base a dichos valores, se realizará un análisis sobre los mismos, los cuales representan una comparación entre la imagen original y la imagen procesada con el método elegido. En verde se encuentra resaltado el valor que resultó más favorable de acuerdo al parámetro elegido para realizar la comparación.

Los parámetros tenidos en cuenta fueron el contraste, la entropía, uniformidad, nivel de gris y zona de variabilidad. Un alto contraste (contrast) sugiere bordes definidos y destacados. Una alta entropía (entropy) puede indicar textura más compleja, lo que puede facilitar la identificación de bordes claros, es decir que habría mayor diferenciación entre el tejido y el hueso. Luego, valores bajos de uniformidad (run length non uniformity) puede reflejar bordes más uniformes y definidos. Un valor bajo de nivel de gris (gray level non uniformity) indica una imagen con menos ruido y bordes más definidos. Y por último, valores altos de la zona de variabilidad (zone variance) sugieren una mayor variabilidad en el tamaño de las zonas, lo que mejora la identificación de estructuras bien definidas y bordes claros.

Por ende, los parámetros seleccionados a partir de los resultados de PyRadiomics fueron:

- **original\_glm\_Contrast:** Mide la intensidad de los bordes en la imagen. Un valor alto indica una alta variación de intensidad y bordes definidos.
- **original\_firstorder\_Entropy:** Mide la aleatoriedad en la distribución de los valores de intensidad en la imagen, sin considerar la relación espacial entre píxeles. Un valor alto indica una distribución de intensidades más dispersa o impredecible, lo que sugiere una textura más compleja en la imagen.
- **original\_glrmlm\_RunLengthNonUniformityNormalized:** Mide la uniformidad de las longitudes de las secuencias de valores grises. Un valor bajo indica bordes más uniformes.
- **original\_glrmlm\_GrayLevelNonUniformityNormalized:** Mide la uniformidad de los niveles de gris. Un valor bajo indica bordes más uniformes.
- **original\_glszm\_SizeZoneNonUniformityNormalized:** Mide la variabilidad del tamaño de las zonas. Un valor bajo indica bordes más uniformes.
- **original\_glszm\_ZoneVariance:** Mide la variabilidad de las zonas de intensidad. Un valor alto puede indicar bordes más definidos.
- **original\_ngtdm\_Contrast:** Mide el contraste entre los píxeles y sus vecinos. Un valor alto indica bordes más definidos.

Si bien varios de los parámetros seleccionados comparten la característica a analizar, cada uno obtiene el valor a través de distintas formas de cálculo. Es por esto que fueron seleccionados todos ellos para asegurar el correcto análisis y comparación de resultados. A simple vista, se puede evidenciar que el Método 1 es el que contiene la gran mayoría de los resultados favorables, a excepción de la característica que analiza el contraste.

	Método 1	Método 2	Método 3
original_glm_Contrast	0.3827347998496402	0.162651052675474	0.46910894229410527
original_firstorder_Entropy	1.8723910258361094	0.615843954110711	1.3046106485630284
original_glrmlm_RunLengthNonUniformityNormalized	0.30240572010902705	0.7123620918424466	0.4216079425024142
original_glrmlm_GrayLevelNonUniformityNormalized	0.27704817491364214	0.7254170447632151	0.3828868753394065
original_glszm_SizeZoneNonUniformityNormalized	0.08264889629321116	0.22145207589441995	0.17713674663093612



original_glszm_ZoneVariance	1892.6459975010412	151.05766762351513	497.578061789791
original_ngtdm_Contrast	0.02646110551229112	0.007081040193387573	0.011554447515607604

## Conclusión

Luego de implementar diversos algoritmos de procesamiento de imágenes, se logró llegar al resultado esperado de diferentes formas. Los tres métodos elegidos han permitido definir los bordes de la columna vertebral y fue observado que un algoritmo por sí solo no es suficiente para obtener un resultado deseable, sino que mediante la combinación de ellos fueron logradas las soluciones elegidas.

Mediante el análisis de los resultados de la implementación de los diferentes métodos detallados anteriormente, fue posible llegar a diversas conclusiones. Si bien visualmente se llega a definir los bordes de la columna vertebral, se optó por un enfoque más numérico para analizar la solución brindada por cada uno.

Como fue mencionado anteriormente, el resultado de comparar la imagen procesada con la original fue más óptimo utilizando el Método 1, el cual realiza tres procesos sobre la imagen, incluido el operador morfológico opening y esto la destaca sobre el resto. Sin embargo, en ninguno de los parámetros elegidos para realizar la comparación se toma en cuenta la pérdida de información provocada por el operador morfológico open.

Si bien el primer Método es el de mejor rendimiento en cuanto a resultado numérico, el Método 3 es cercano a él en valores y además no incluye pérdida de información. Por ello, si la pérdida de información resulta demasiado sustanciosa para quien requiere procesar la imagen, el Método 3 debería ser el utilizado, y si la pérdida de información se considera despreciable, entonces utilizar el Método 3 dará los mejores resultados en general.

# Bibliografía

OpenCV. (s.f.). Biblioteca de Visión por Computadora de Código Abierto. Recuperado de <https://docs.opencv.org/>

PyRadiomics. (s.f.). Documentación de PyRadiomics. Recuperado de <https://pyradiomics.readthedocs.io/en/latest/>

Massa, J. (2024). *Procesamiento de imágenes 1 - Pattern recognition* [Filminas]