



# UNIVERSIDAD DE GRANADA

## Laboratorio 4: Segmentación con ITK

TVG curso:2024/2025

Adriano García - Giralda Milena  
DNI:77944452-M [adrianoggm@correo.ugr.es](mailto:adrianoggm@correo.ugr.es) Grupo 1

17 de junio de 2025

# Índice

<b>1. Introducción</b>	<b>2</b>
<b>2. Tarea 1: Segmentación por Crecimiento de Regiones</b>	<b>3</b>
2.1. Parámetros evaluados	4
2.2. Segmentación sobre <code>saltAndPepperNoise1.jpg</code>	6
2.3. Segmentación sobre <code>gaussianNoise.jpg</code>	8
2.4. Segmentación sobre <code>coronaryAngiogram.png</code>	10
2.5. Análisis	11
<b>3. Tarea 2</b>	<b>13</b>
3.1. Segmentación Watershed sobre <code>saltAndPepperNoise1.jpg</code>	13
3.2. Segmentación Watershed sobre <code>gaussianNoise.jpg</code>	16
3.3. Segmentación Watershed sobre <code>BrainProtonDensitySlice256×256.png</code>	17
3.4. Comparativa	18

---

# 1. Introducción

En esta práctica hemos desarrollado dos programas independientes, cada uno ilustrando diferentes técnicas de segmentación espacial disponibles en ITK:

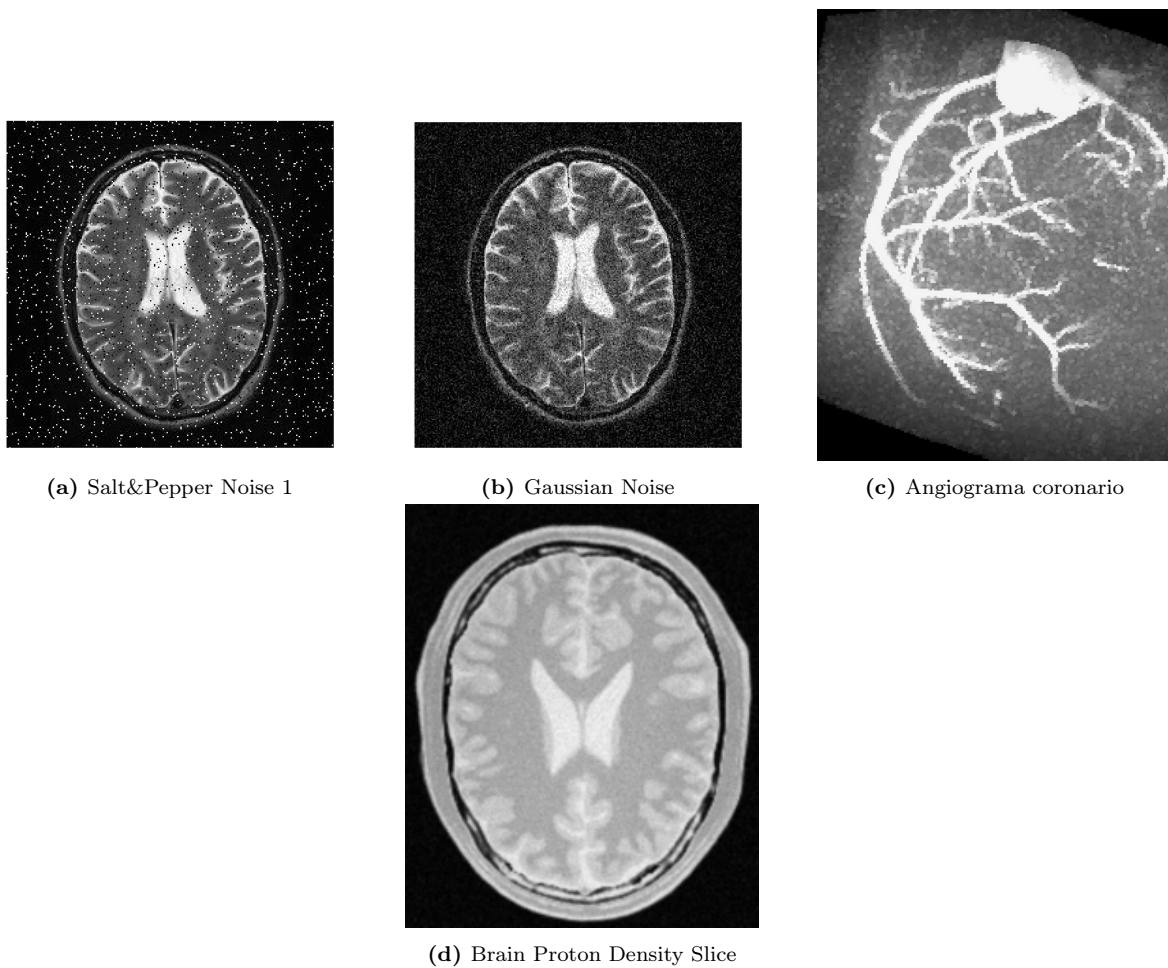
1. **Segmentación basada en Crecimiento de Regiones:** Se exploran tres variantes:

- *ConnectedThresholdImageFilter*, que crece desde una semilla en función de un rango de intensidades fijas.
- *NeighborhoodConnectedImageFilter*, que añade la condición de homogeneidad en un vecindario definido alrededor de cada píxel.
- *ConfidenceConnectedImageFilter*, que realiza un ajuste estadístico iterativo sobre la media y desviación típica local.

Cada técnica se ha probado tanto con preprocesamiento mediante *CurvatureFlowImageFilter* como sin él, y con diversas configuraciones de parámetros.

2. **Segmentación Automática mediante Watershed:** Se emplea el *WatershedImageFilter*, precedido por el cálculo del gradiente de magnitud y un suavizado opcional (difusión anisotrópica o mediana). Se han evaluado múltiples combinaciones de parámetros **Threshold** y **Level** para estudiar su impacto sobre la fragmentación de las regiones segmentadas.

Los experimentos se han realizado sobre un conjunto de imágenes que incluye **BrainProtonDensitySlice256x256.png**, **coronaryAngiogram.png**, **saltAndPepperNoise1.jpg** y **gaussianNoise.jpg**. Este conjunto representa tanto imágenes clínicas como sintéticas con distintos tipos de ruido, lo que permite observar visualmente el efecto de cada técnica, comparar sus ventajas y limitaciones, y profundizar en la configuración de parámetros en ITK.



**Figura 1:** Imágenes originales empleadas en la práctica.

## 2. Tarea 1: Segmentación por Crecimiento de Regiones

Write a program that reads an image from a file and displays the result of applying the following filters to the input image: `ConnectedThresholdImageFilter`, `NeighborhoodConnectedImageFilter`, and `ConfidenceConnectedImageFilter`. The idea is to obtain several results by combining the free parameters of each filter, and by using or not a noise reduction filter over an input image before applying each one of these filters. The idea is to learn how the different filter parameters affect the segmentation result and observe how the application, or not, of a noise reduction filter previously in the pipeline affects the segmentation result. Try this program with the input images: `BrainProton-DensitySlice256x256.png`, `coronaryAngiogram.png`, `saltAndPepperNoise1.jpg`, and `gaussianNoise.jpg`. En esta tarea hemos implementado un programa que aplica tres técnicas de segmentación por crecimiento de regiones sobre una imagen de entrada, variando sus parámetros característicos y evaluando el impacto del preprocesamiento mediante filtrado de ruido. Los filtros considerados son:

- **ConnectedThresholdImageFilter** Segmenta todos los píxeles conectados a una semilla que se encuentren dentro de un rango de intensidades fijo.
- **NeighborhoodConnectedImageFilter** Añade la condición de homogeneidad en un vecindario (de radio configurable) para extender la región.
- **ConfidenceConnectedImageFilter** Ajusta estadísticamente los valores aceptables ( $\text{media} \pm \text{multiplicador} \times \text{desviación típica}$ ) durante múltiples iteraciones.

Además, se ha incluido opcionalmente un filtrado previo mediante **CurvatureFlowImageFilter** con  $\sigma = 1,0$  (5 iteraciones), comparando sus efectos frente a la segmentación sin suavizado.

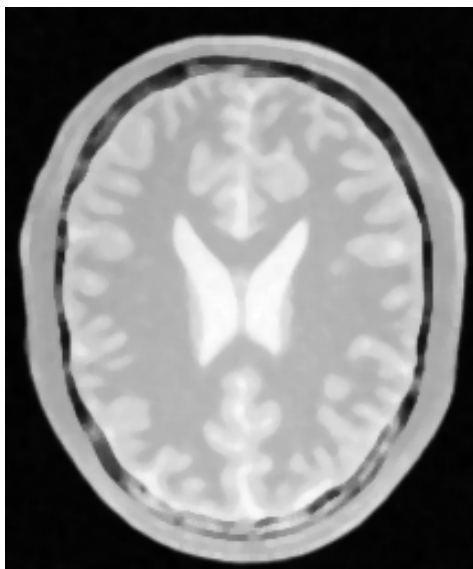
### 2.1. Parámetros evaluados

Los parámetros explorados para cada filtro han sido:

- **ConnectedThreshold:** Rango de intensidad:  $[50,150]$  y  $[80,200]$ .
- **NeighborhoodConnected:** Radios del vecindario: 1 y 3.
- **ConfidenceConnected:** Radios: 1 y 2, multiplicadores: 1.0 y 2.5, iteraciones: 1 y 3.

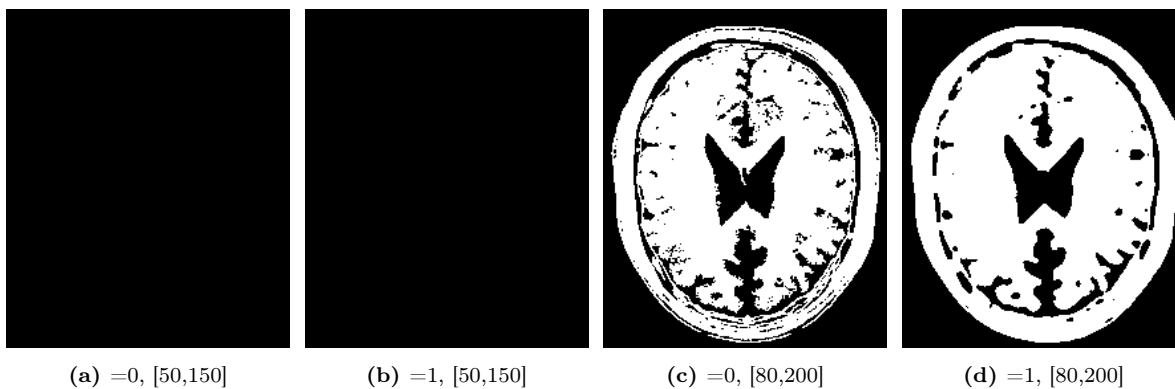
subsectionSegmentación sobre BrainProtonDensitySlice256x256.png

#### 2.1.0.1 Imagen pre-segmentación



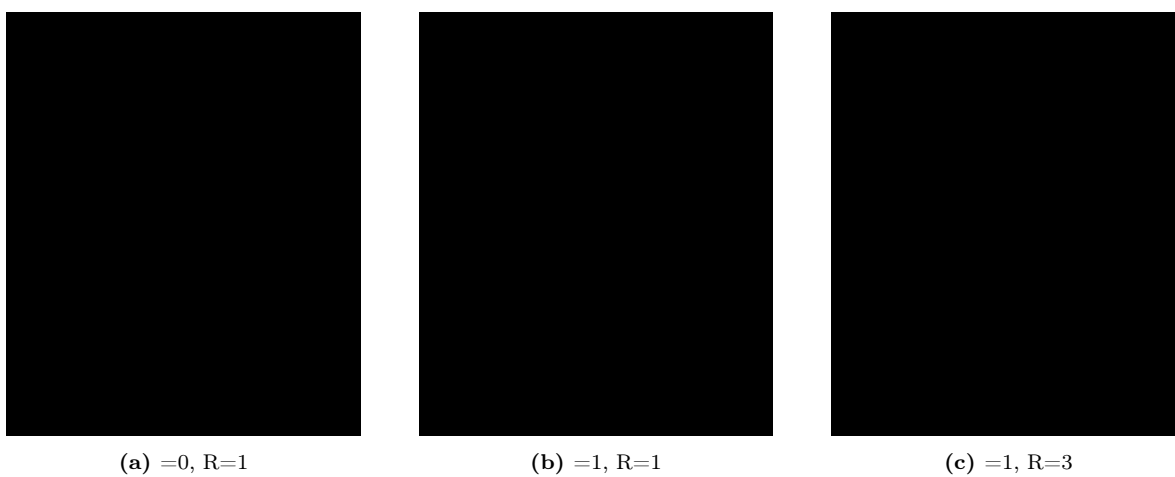
**Figura 2:** Imagen tras filtro de mediana  $5 \times 5$  ( $=1$ ).

### 2.1.0.2 ConnectedThresholdImageFilter



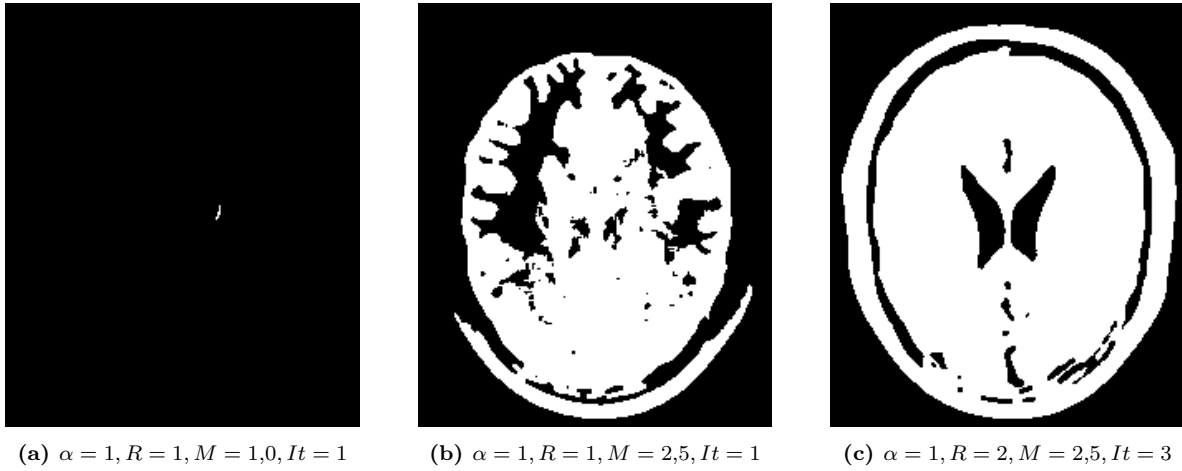
**Figura 3:** ConnectedThreshold con distintos umbrales y preprocesado ().

### 2.1.0.3 NeighborhoodConnectedImageFilter



**Figura 4:** NeighborhoodConnected variando radio y filtro previo.

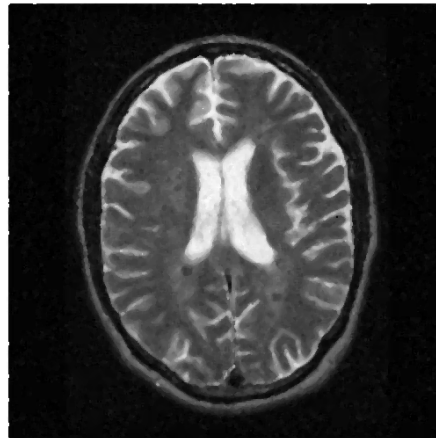
#### 2.1.0.4 ConfidenceConnectedImageFilter



**Figura 5:** Resultados de ConfidenceConnectedImageFilter con distintos parámetros y comparación sin/con suavizado.

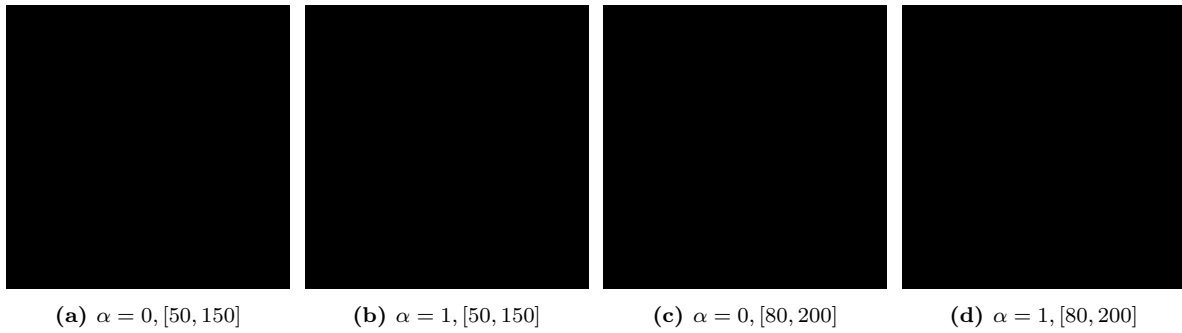
## 2.2. Segmentación sobre saltAndPepperNoise1.jpg

Además, mostramos antes la imagen pre-segmentación tras el filtrado ( $\alpha = 1$ ) :



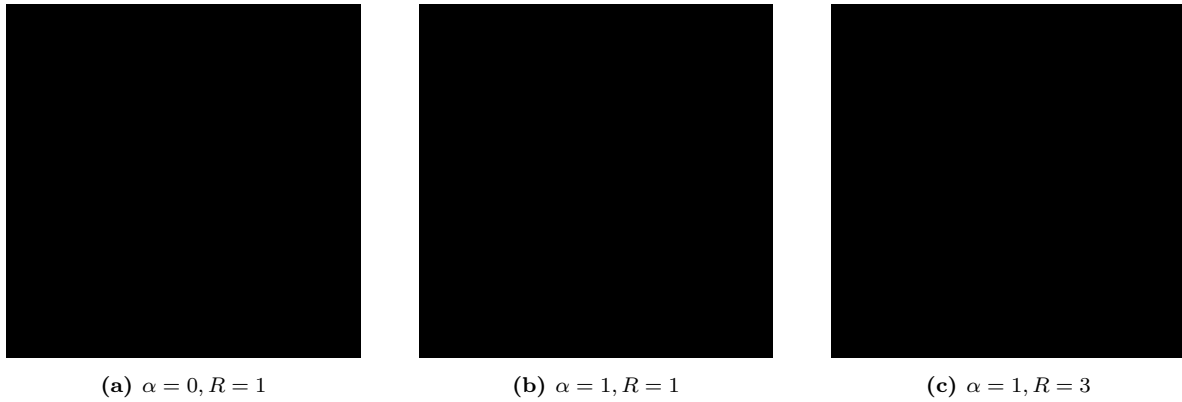
**Figura 6:** Imagen tras filtro de mediana  $5 \times 5$  ( $\alpha = 1$ ).@

### 2.2.0.1 ConnectedThresholdImageFilter



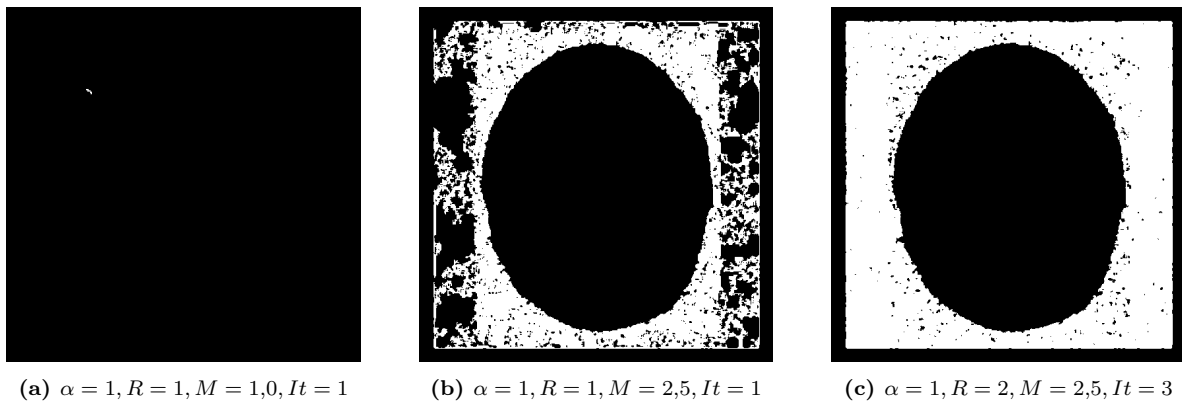
**Figura 7:** ConnectedThreshold con distintos umbrales y preprocesado ( $\alpha$ ).@

### 2.2.0.2 NeighborhoodConnectedImageFilter



**Figura 8:** NeighborhoodConnected variando radio y filtro previo.

### 2.2.0.3 ConfidenceConnectedImageFilter

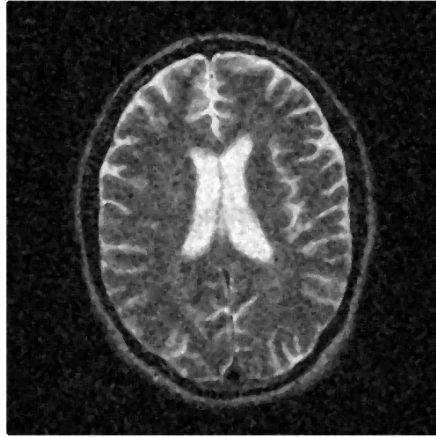


**Figura 9:** Resultados de ConfidenceConnectedImageFilter sobre saltAndPepperNoise1 con distintas configuraciones y comparación con suavizado.



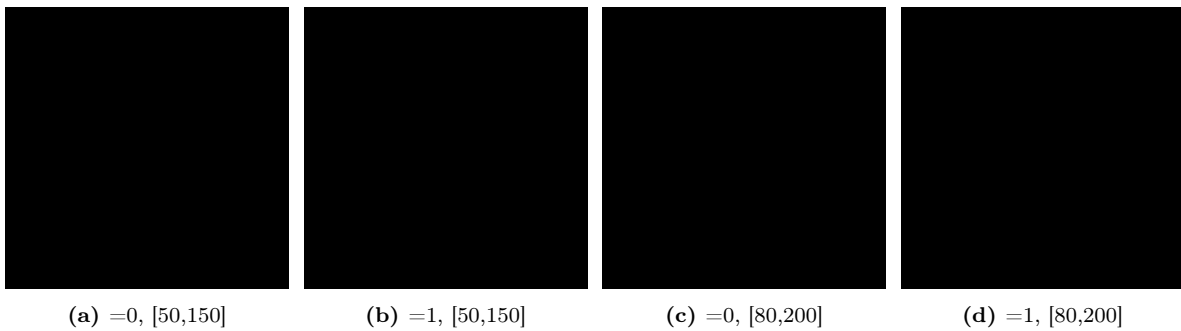
## 2.3. Segmentación sobre gaussianNoise.jpg

### 2.3.0.1 Imagen pre-segmentación



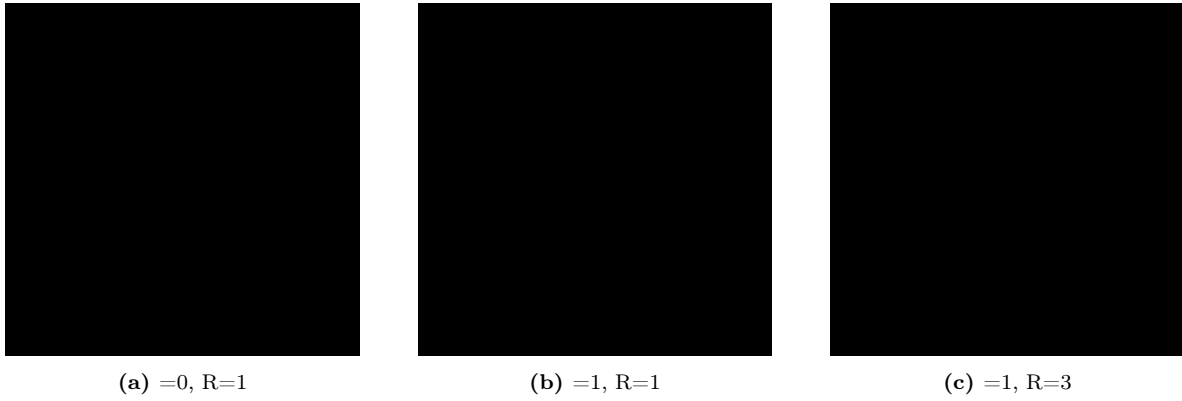
**Figura 10:** Imagen tras filtro de mediana  $5 \times 5$  ( $=1$ ).

### 2.3.0.2 ConnectedThresholdImageFilter



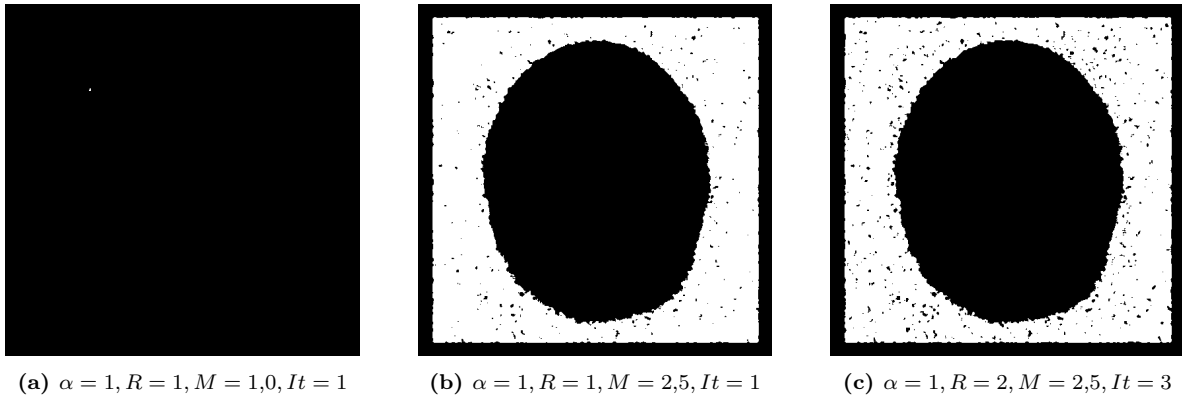
**Figura 11:** ConnectedThreshold con distintos umbrales y preprocesado ().

### 2.3.0.3 NeighborhoodConnectedImageFilter



**Figura 12:** NeighborhoodConnected variando radio y filtro previo.

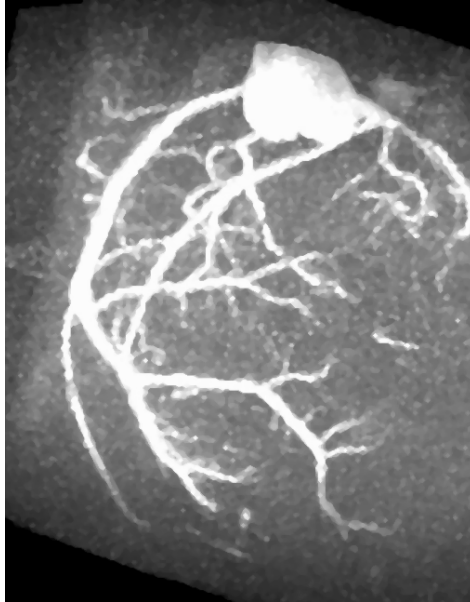
### 2.3.0.4 ConfidenceConnectedImageFilter



**Figura 13:** Resultados de ConfidenceConnectedImageFilter con distintos parámetros y comparación sin/con suavizado.

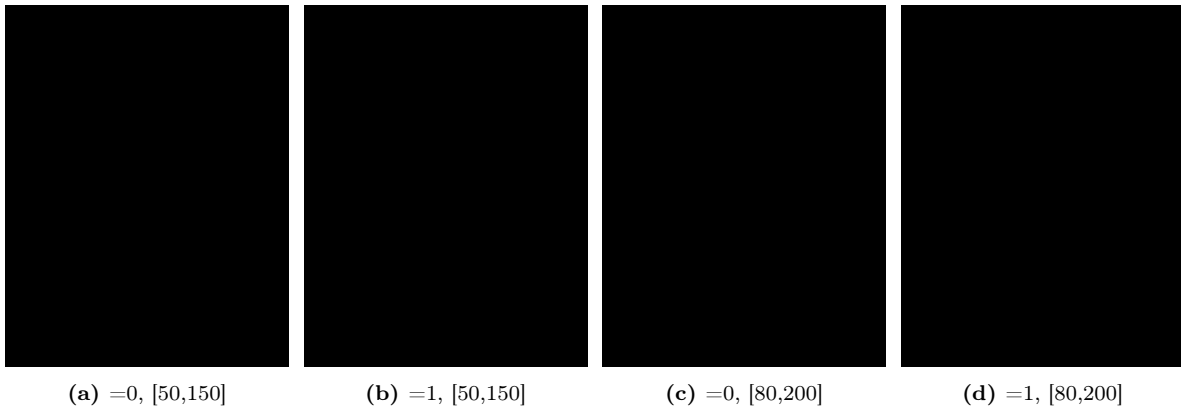
## 2.4. Segmentación sobre coronaryAngiogram.png

### 2.4.0.1 Imagen pre-segmentación



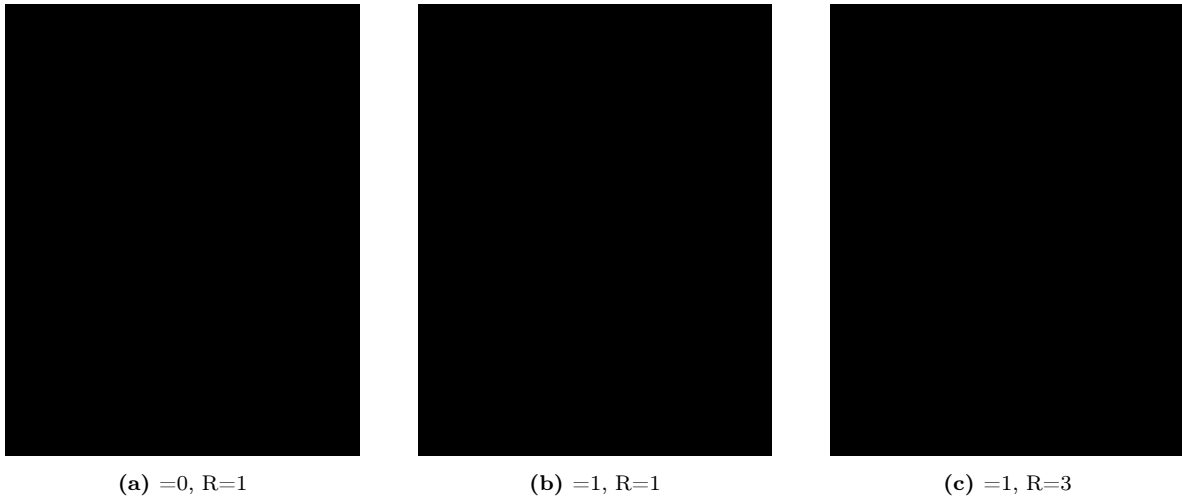
**Figura 14:** Imagen tras filtro de mediana  $5 \times 5$  ( $=1$ ).

### 2.4.0.2 ConnectedThresholdImageFilter



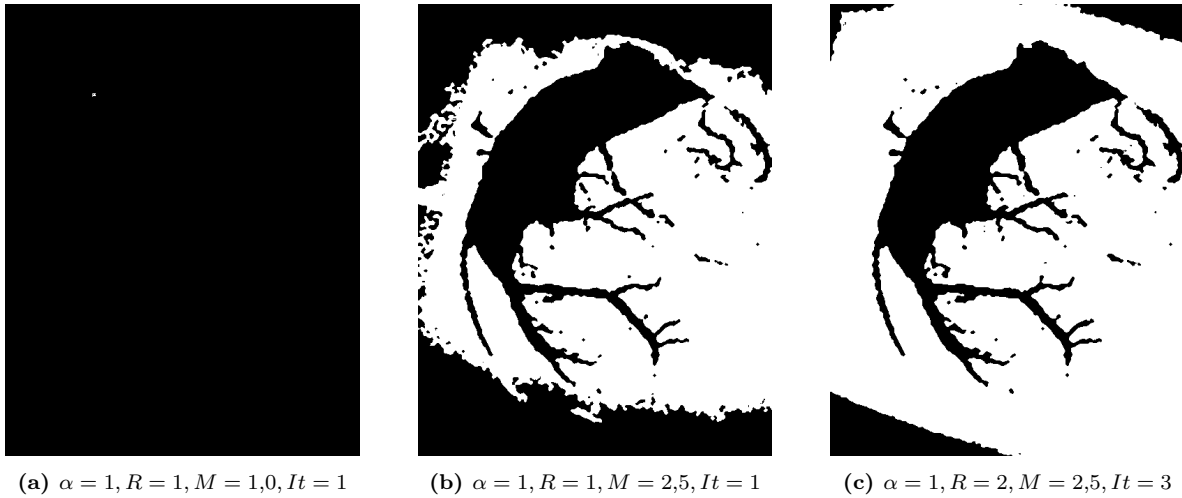
**Figura 15:** ConnectedThreshold con distintos umbrales y preprocesado ().

#### 2.4.0.3 NeighborhoodConnectedImageFilter



**Figura 16:** NeighborhoodConnected variando radio y filtro previo.

#### 2.4.0.4 ConfidenceConnectedImageFilter



**Figura 17:** Resultados de ConfidenceConnectedImageFilter con distintos parámetros y comparación sin/con suavizado.

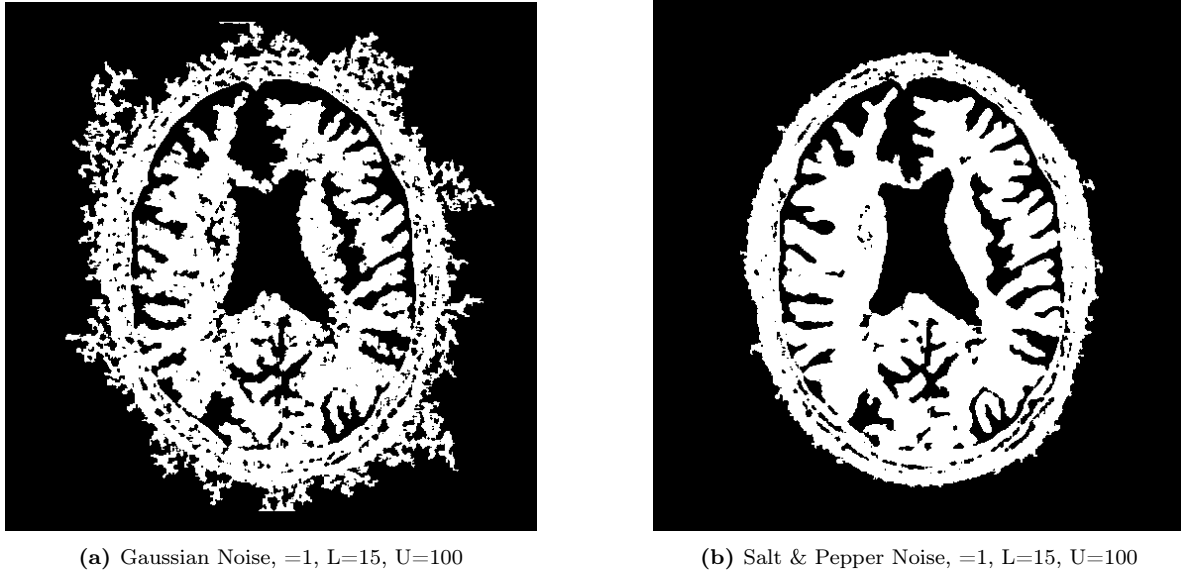
### 2.5. Análisis

En la práctica, el desempeño de los filtros de crecimiento de regiones resulta **\*\*altamente dependiente\*\*** del dominio de la imagen y del umbral elegido para la binarización. Aunque teóricamente:

- El suavizado previo mediante `CurvatureFlowImageFilter` elimina ruido aislado y estabiliza fronteras.
- `ConnectedThreshold` falla fácilmente sin preprocesamiento y es muy sensible al rango de umbrales.

- `NeighborhoodConnected` gana robustez con radios mayores, pero puede quedar demasiado restrictivo cuando el contraste local es bajo.
- `ConfidenceConnected` se adapta mejor, aunque exige ajuste fino de múltiples parámetros para converger a una máscara coherente.

En nuestros experimentos iniciales, únicamente `ConfidenceConnectedImageFilter` mostró regiones algo reconocibles. Sin embargo, **\*\*al probar rangos más bajos\*\*** (por ejemplo `[15,100]`) en `ConnectedThreshold`, obtuvimos resultados más significativos en imágenes oscuras:



**Figura 18:** Segmentación con `ConnectedThreshold` usando rango `[15,100]`: mejor desempeño en imágenes con bajo nivel de gris.

No obstante, a pesar de estos ajustes, las máscaras generadas en la mayoría de casos siguen siendo de calidad insuficiente, con regiones mal definidas o excesivamente fragmentadas. Un operador humano, guiado por la percepción visual, aún es capaz de segmentar estas estructuras con mucha mayor precisión y coherencia espacial. Adicionalmente vemos que estos filtros son muy afectados por la presencia de ruido.

### 2.5.0.1 Conclusiones prácticas

Este experimento pone de manifiesto que:

1. **No existe un “umbral universal”.** Cada conjunto de imágenes puede requerir un rango de intensidades muy distinto —incluso dentro de la misma modalidad—, por lo que la segmentación manual o semiautomática basada en inspección de histograma sigue siendo necesaria.
2. **La robustez frente al ruido es limitada.** Aunque el preprocesado por mediana o flujo de curvatura mejora ligeramente los resultados, en presencia de ruido extremo (salt & pepper, gaussianos fuertes), los algoritmos clásicos de crecimiento de regiones hilan fino y tienden a fragmentar las regiones.
3. **Se necesitan estrategias adaptativas.** Automatizar la selección de umbrales (por ejemplo, mediante Otsu o clustering en el histograma) o combinar múltiples criterios (intensidad, textura, forma) puede ofrecer un salto cualitativo en la binarización.

- 
4. **Importancia del conocimiento del dominio.** Conocer la anatomía y las características físicas de las estructuras a segmentar permite acotar mejor los parámetros y, en última instancia, decidir si un método clásico es suficiente o si es más eficaz recurrir a enfoques basados en aprendizaje profundo.

Por tanto, aunque estos filtros de crecimiento de regiones son herramientas útiles, su uso en aplicaciones reales debe ir acompañado de:

- Un paso previo de *análisis de histograma* para delimitar de forma automática las franjas de interés.
- Una etapa de *validación visual* o semiautomática, donde el usuario revise y ajuste manualmente las máscaras generadas.
- La exploración de *técnicas híbridas* (por ejemplo, segmentación inicial con region growing y refinado posterior con nivel de grises activo o redes neuronales).

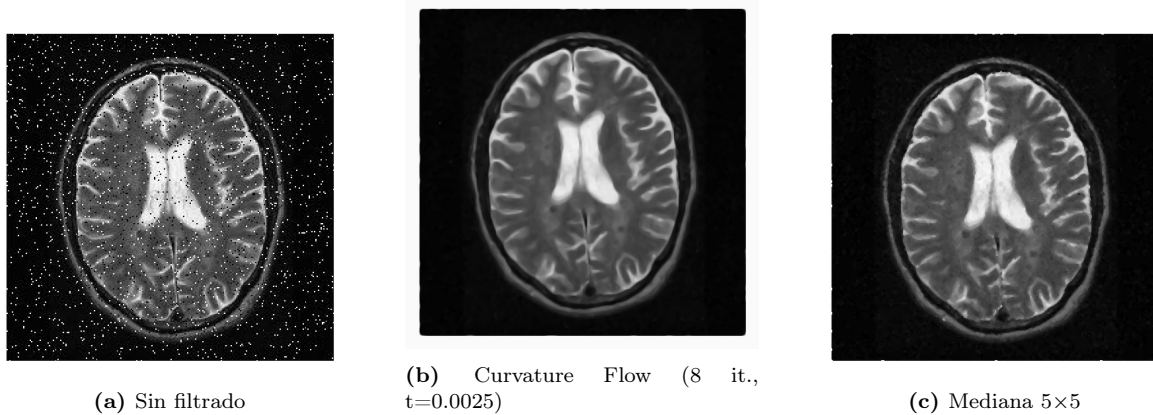
En conclusión, estos experimentos refuerzan la idea de que la segmentación basada únicamente en crecer regiones por intensidad es **insuficiente** para imágenes con variabilidad de contraste o ruido, y que la **selección de umbral** y el **conocimiento del dominio** son factores decisivos para obtener resultados aceptables.

## 3. Tarea 2

En esta tarea implementamos y comparamos watershed sobre las mismas imágenes con ruido (saltAndPepperNoise1.jpg, coronaryAngiogram.jpg y gaussianNoise.jpg) y BrainProtonDensitySlice256×256.png

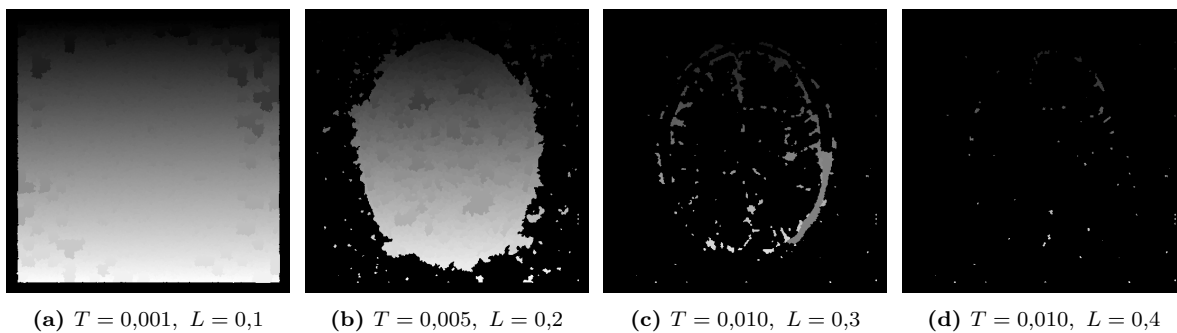
### 3.1. Segmentación Watershed sobre saltAndPepperNoise1.jpg

#### 3.1.0.1 Imágenes pre-segmentación (filtro previo)



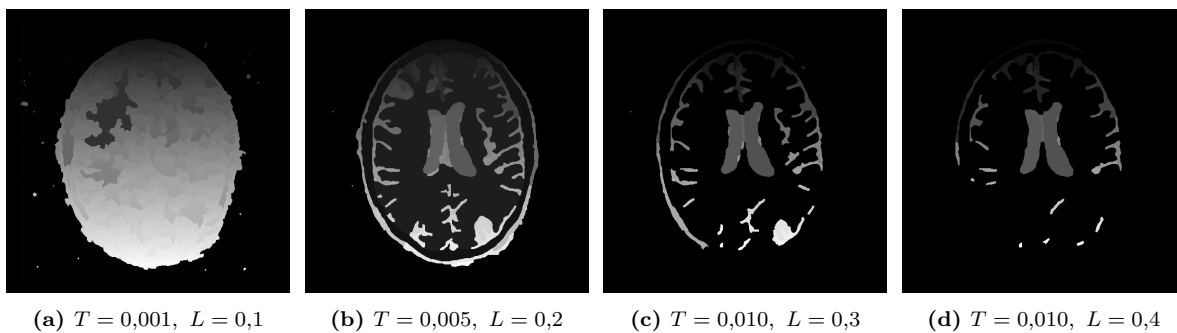
**Figura 19:** Imágenes tras los filtros previos al watershed sobre saltAndPepperNoise1.jpg.

### 3.1.0.2 Watershed sin filtrado (none)



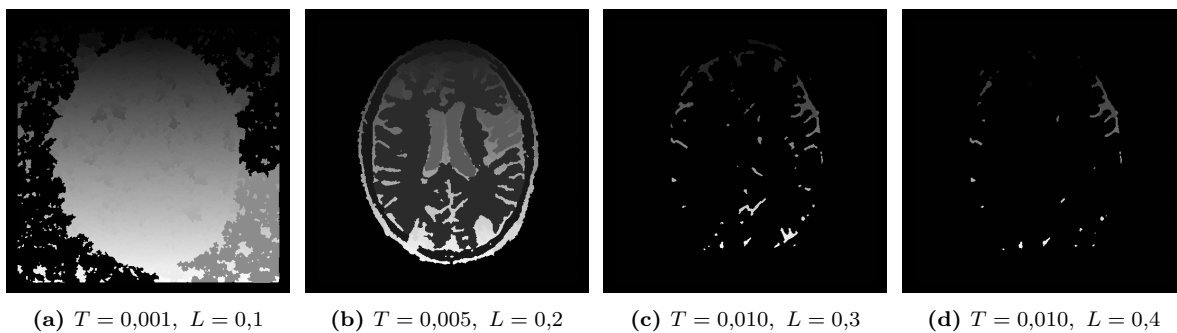
**Figura 20:** Watershed sin pre-filtrado (none) sobre saltAndPepperNoise1.jpg.

### 3.1.0.3 Watershed tras Curvature Flow (curvature)



**Figura 21:** Watershed tras Curvature Flow sobre saltAndPepperNoise1.jpg.

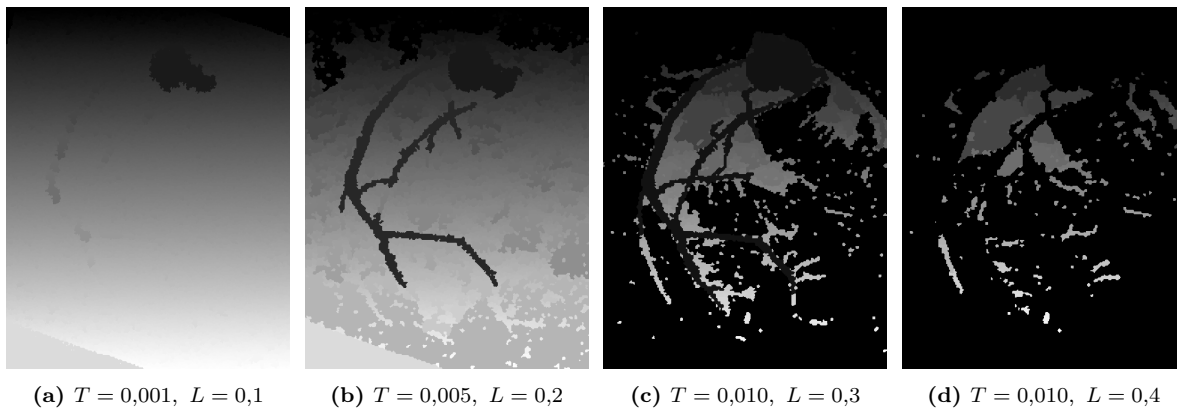
### 3.1.0.4 Watershed tras Mediana 5×5 (median)



**Figura 22:** Watershed tras Mediana 5×5 sobre saltAndPepperNoise1.jpg.

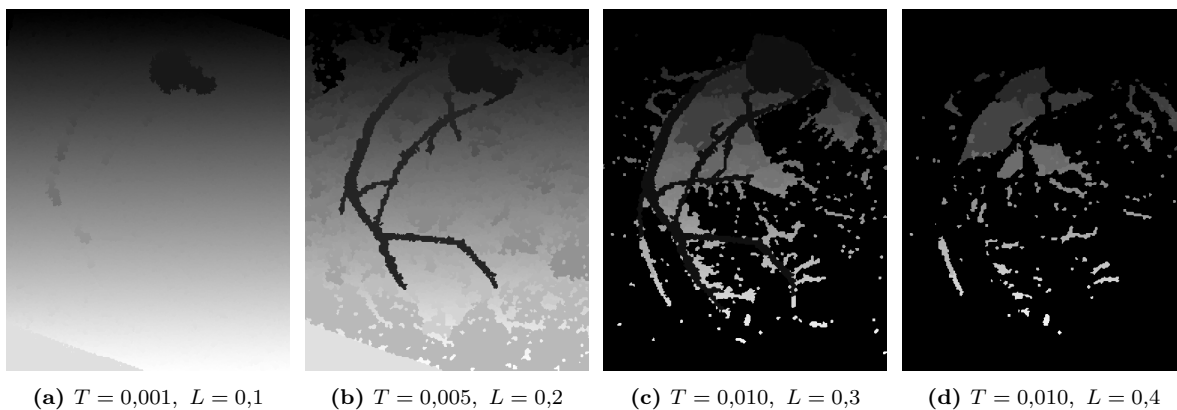
3.1.0.5 Watershed sobre coronaryAngiogram.jpg

3.1.0.6 Watershed sin filtrado (none)



**Figura 23:** Watershed sin pre-filtrado (none) sobre coronaryAngiogram.jpg.

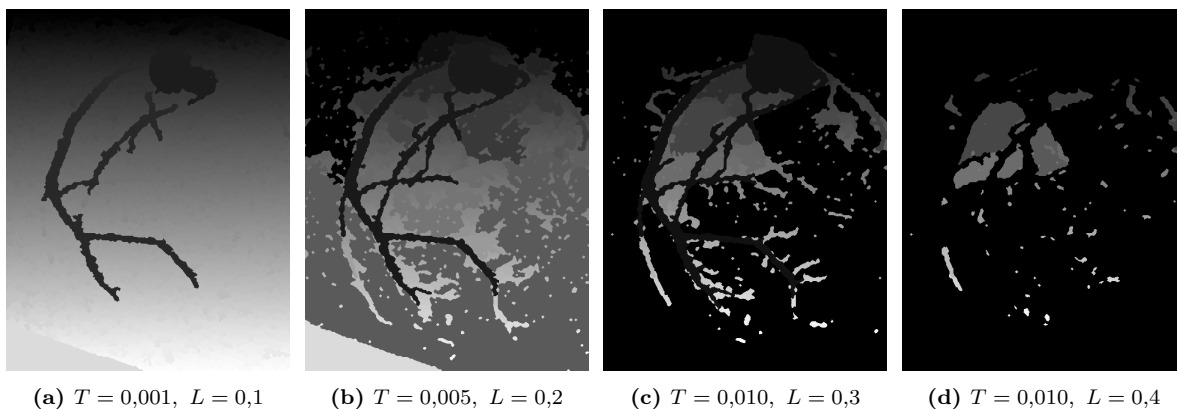
3.1.0.7 Watershed tras Curvature Flow (curvature)



**Figura 24:** Watershed tras Curvature Flow sobre coronaryAngiogram.jpg.



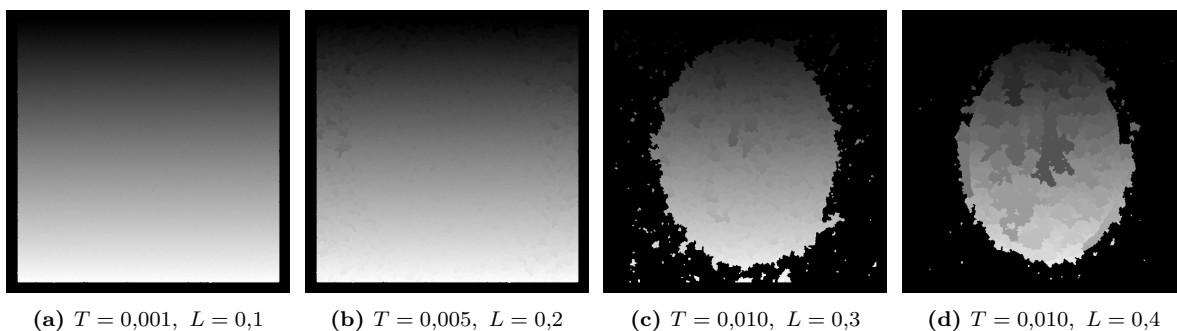
### 3.1.0.8 Watershed tras Mediana $5 \times 5$ (median)



**Figura 25:** Watershed tras Mediana  $5 \times 5$  sobre coronaryAngiogram.jpg.

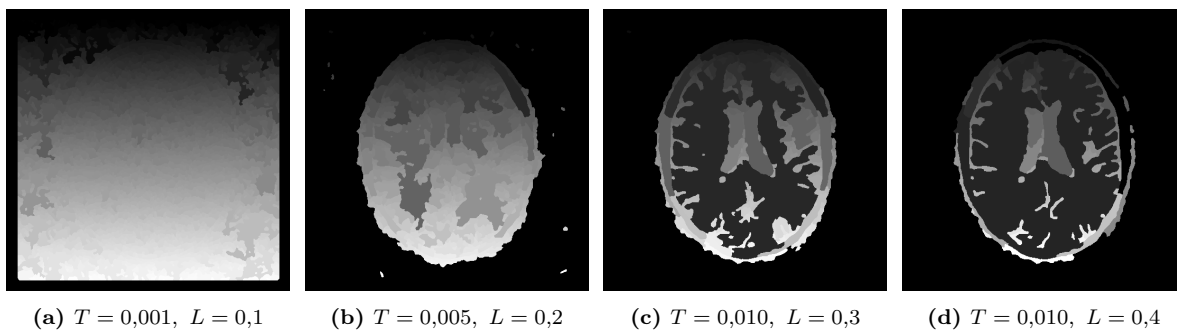
## 3.2. Segmentación Watershed sobre gaussianNoise.jpg

### 3.2.0.1 Watershed sin filtrado (none)



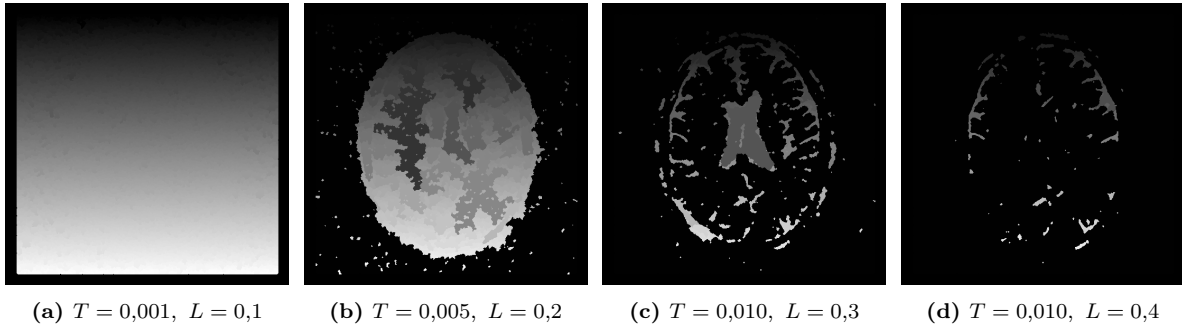
**Figura 26:** Watershed sin pre-filtrado (none) sobre gaussianNoise.jpg.

### 3.2.0.2 Watershed tras Curvature Flow (curvature)



**Figura 27:** Watershed tras Curvature Flow sobre gaussianNoise.jpg.

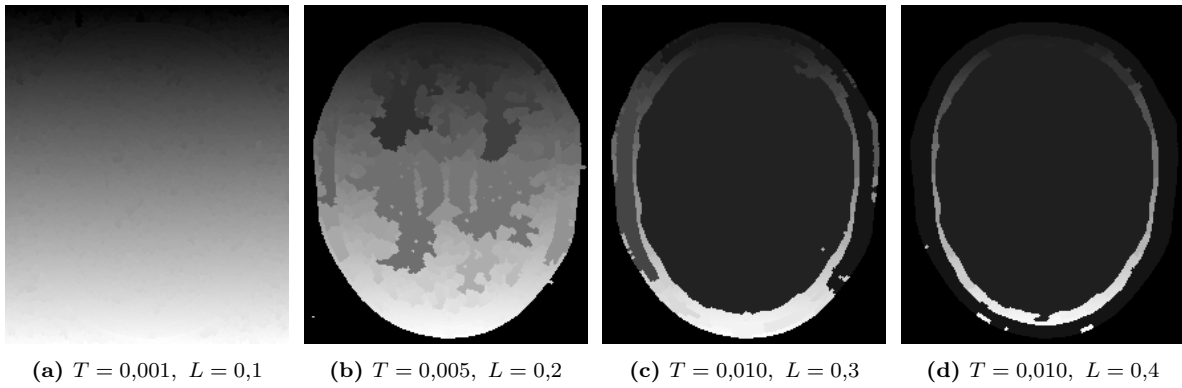
### 3.2.0.3 Watershed tras Mediana 5×5 (median)



**Figura 28:** Watershed tras Mediana 5×5 sobre gaussianNoise.jpg.

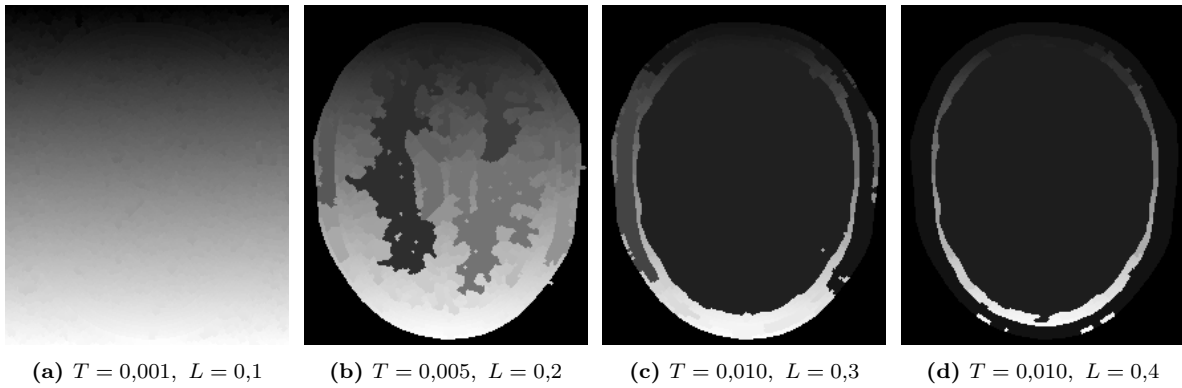
## 3.3. Segmentación Watershed sobre BrainProtonDensitySlice256×256.png

### 3.3.0.1 Watershed sin filtrado (none)



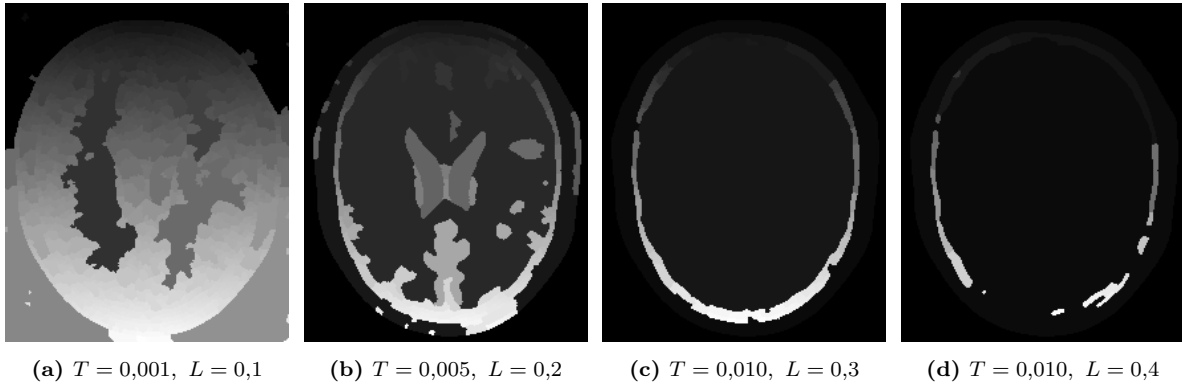
**Figura 29:** Watershed sin pre-filtrado (none) sobre BrainProtonDensitySlice256×256.png.

### 3.3.0.2 Watershed tras Curvature Flow (curvature)



**Figura 30:** Watershed tras Curvature Flow sobre BrainProtonDensitySlice256×256.png.

### 3.3.0.3 Watershed tras Mediana $5 \times 5$ (median)



**Figura 31:** Watershed tras Mediana  $5 \times 5$  sobre `BrainProtonDensitySlice256x256.png`.

## 3.4. Comparativa

En este estudio hemos comparado dos familias de métodos de segmentación:

- **Region Growing** (Task 1): `ConnectedThreshold`, `NeighborhoodConnected` y `ConfidenceConnected`.
- **Watershed** (Task 2): con distintos filtros previos (`none`, `Curvature Flow`, `Mediana`) y parámetros  $(T, L)$ .

### 3.4.0.1 Ventajas de Watershed

- Produce contornos más nítidos y continuos, especialmente en estructuras vasculares como en `coronaryAngiogram.jpg`, donde capta bien las ramas principales sin romper el trazado.
- Al ajustar el parámetro de nivel ( $L$ ), podemos regular la granularidad de los segmentos, extrayendo regiones amplias o finas según la escala de interés.
- Con un pre-filtrado ligero (`Curvature Flow`, 8 it,  $\Delta t = 0,0025$ ), se suaviza el ruido sin difuminar fronteras, mejorando la claridad de las crestas de gradiente.

### 3.4.0.2 Limitaciones de Watershed

- En imágenes con intensidades muy homogéneas y bajo contraste, como `BrainProtonDensitySlice256x256.png`, las líneas de cuenca tienden a alinearse en zonas poco significativas y generan sobre-segmentación.
- La elección de  $(T, L)$  sigue siendo crítica: un umbral demasiado bajo introduce “falsas” cuencas, mientras que uno alto puede fundir regiones contiguas.

### 3.4.0.3 Comparación frente a Region Growing

- En `saltAndPepperNoise1.jpg`, las técnicas de Region Growing (especialmente `ConfidenceConnected`) eran muy sensibles al ruido impulsivo, mostrando regiones fragmentadas o completamente negras sin un pre-filtrado agresivo. Watershed, por su parte, aunque introduce artefactos en los bordes, mantiene estructuras completas tras un suave pre-filtrado.

- Para `gaussianNoise.jpg`, Region Growing requería umbrales bajos y radios grandes para “ver” algo, a costa de incluir ruido residual. Watershed consigue delimitar crestas de gradiente más estables incluso con ruido gaussiano intenso.
- En `coronaryAngiogram.jpg`, Region Growing sólo capturaba pequeñas regiones aisladas sin reflejar la topología vascular; Watershed delineó con claridad el árbol de vasos principales.
- Solo en `BrainProtonDensitySlice256×256.png` las técnicas de Task 1 (umbral adaptativo y vecinos estadísticos) ofrecieron máscaras ligeramente más coherentes, probablemente porque la variabilidad de intensidad era alta y las regiones de interés tenían límites difusos que Watershed sobre-segmentó.

#### 3.4.0.4 Conclusión

1. **Dominio de la imagen:** No hay un método único óptimo; Watershed destaca en estructuras con bordes bien marcados y ruido suave, mientras que Region Growing (con ajuste fino de umbrales) puede funcionar mejor en imágenes con gradientes débiles pero variabilidad interna marcada.
2. **Pre-procesado:** Un ligero Curvature Flow preserva mejor los bordes que un filtrado de mediana agresivo, y facilita la extracción de crestas de gradiente estables.
3. **Parámetros críticos:**
  - Region Growing: selección de rango  $[L, U]$ , radio de vecindad y multiplicador de confianza.
  - Watershed: combinación de (Threshold, Level) y modo de suavizado previo.
4. **Recomendación práctica:**
  - Visualizar siempre histograma y mapa de gradiente antes de segmentar.
  - Automatizar búsqueda de parámetros (p.ej. barrido automático con validación cruzada o enfoque interactivo).
  - Considerar enfoques híbridos: usar Watershed para delimitación inicial y luego refinar con Region Growing o morfología matemática para cerrar huecos y eliminar ruido.