

Memoria-VA

Estevo Aldea Arias

December 2025

**5. Segmentación y conteo de núcleos celulares en histología de
anatomía patológica**

Contents

1	Introducción	1
1.1	Arquitectura y Organización del Proyecto	1
1.2	Instalación y Ejecución	1
2	Memoria del proyecto	2
2.1	Histórico de Metodologías	2
2.2	Metodología Final	4

1 Introducción

1.1 Arquitectura y Organización del Proyecto

El proyecto se estructura en torno a tres módulos secuenciales, con un **Makefile** para facilitar el flujo.

Material Celulas/ Directorio que contiene el dataset de entrada y los recursos del problema (imágenes originales y *Ground Truth*).

segmentar.py Constituye el núcleo del *pipeline*. Este módulo procesa las 30 imágenes para realizar la segmentación de núcleos. Los resultados intermedios se almacenan en el directorio **visualizaciones**, mientras que los datos numéricos se exportan a **resultados.csv**.

evaluar.py Responsable de la validación cuantitativa. Calcula métricas de segmentación, conteo y morfología comparando los resultados con el *Ground Truth*. El reporte general se muestra por pantalla y los detalles se anexan a **evaluacion.csv**.

visualizar.py Genera representaciones gráficas para el análisis cualitativo. Crea el subdirectorio **visualizaciones/RESULTADOS**, generando para cada caso una composición de cuatro imágenes:

1. Imagen original preprocesada (canal Hematoxilina).
2. Máscara binaria post-umbralización.
3. Segmentación final tras la aplicación del algoritmo *Watershed*.
4. Comparativa visual contra el *Ground Truth*, codificada por colores: Verdaderos Positivos (Verde), Falsos Negativos (Rojo) y Falsos Positivos (Azul).

Makefile Fichero de configuración.

requirements.txt Archivo de especificación de dependencias y librerías Python necesarias para el entorno.

1.2 Instalación y Ejecución

Para la puesta en marcha, se deben seguir los siguientes pasos desde la raíz del proyecto:

Configuración del entorno

```
# 1. Crear y activar entorno virtual
python3 -m venv .venv
source .venv/bin/activate

# 2. Instalar dependencias
pip install -r requirements.txt
```

Una vez configurado el entorno, el control del flujo de ejecución se realiza mediante el comando **make**, el cual ofrece las siguientes directivas:

- **make all:** Ejecuta el *pipeline* completo de forma secuencial. (*Opción recomendada*).
- **make segmentar / evaluar / visualizar:** Permite la ejecución aislada de cada módulo.
- **make limpiar:** Elimina los archivos generados y borra las carpetas de resultados.
- **make reiniciar:** Realiza una limpieza completa y ejecuta nuevamente todo el proceso desde cero.

2 Memoria del proyecto

2.1 Histórico de Metodologías

El desarrollo del sistema se ha realizado de forma iterativa, manteniendo siempre la misma estructura de *pipeline* (preprocesado, segmentación, postprocesado y evaluación), pero refinando las decisiones de diseño a partir de los resultados cuantitativos.

Fase inicial: umbral fijo y *region growing*. La primera versión partía de un enfoque clásico: conversión a escala de grises del canal de Hematoxilina, umbralización global fija (valor 110) y una etapa de morfología (closing con kernel 3×3) para rellenar huecos. La segmentación se realizaba mediante un *region growing* controlado por gradiente: las semillas se obtenían a partir de la transformada de distancia, y la expansión se detenía en bordes con gradiente elevado. La evaluación se hacía por *matching* núcleo a núcleo (IoU), calculando *Precision*, *Recall* y F1. Esta versión ya permitía una segmentación razonable, pero mostró un F1 global del 38,42 %, con *Recall* muy bajo (31,89 %) y fuerte discrepancia entre las métricas de segmentación y las de conteo, evidenciando fusiones y fragmentaciones frecuentes.

Segunda Fase: umbralización adaptativa y evaluación píxel a píxel. El primer subproblema identificado fue la sensibilidad a la variabilidad de iluminación. Para mitigarlo se incorporaron dos técnicas de umbralización complementarias: Otsu global (para captar la separación núcleo/fondo a nivel de imagen) y umbral adaptativo local (`adaptiveThreshold` de OpenCV), combinados mediante un OR lógico. En paralelo, se introdujo una evaluación a nivel de píxel (F1, IoU, *Precision*, *Recall* y *Accuracy*), más robusta. Con estos cambios, el F1 (píxel a píxel) se incrementó hasta el 65,87 %, con un *Recall* muy alto (86,73 %), a costa de un conteo poco fiable (28,04 %), lo que puso de manifiesto el segundo subproblema: la estrategia de segmentación no separaba bien núcleos tocándose.

Tercera Fase: transición a *watershed* y ajuste fino. Para abordar la separación de núcleos adyacentes se substituyó el *region growing* por un *watershed* clásico sobre la transformada de distancia. Se eliminaron las operaciones morfológicas que fusionaban núcleos y se trabajó en dos frentes: combinación de umbrales (Otsu OR local frente a umbralización secuencial) y ajuste de parámetros del *watershed* (altura mínima en la distancia y dilatación del fondo). Estas versiones consiguieron F1 en torno al 70–73 %, mejorando al mismo tiempo la precisión de área y de conteo mediante estrategias como el relleno de huecos tras el *watershed*. Sin embargo, el número de parámetros y ramas de código empezó a crecer, lo que motivó una reflexión sobre la complejidad del método.

Cuarta Fase: simplificación guiada por el *Ground Truth*. En esta fase se analizó estadísticamente el *Ground Truth* (distribuciones de área, aspecto, circularidad y solidez de 16 819 núcleos) para fijar umbrales con base empírica (por ejemplo, solidez mínima 0,78 y área mínima 50 px²). Se eliminaron heurísticas complejas (resegmentación de núcleos muy elongados, unión manual de fragmentos) y se mantuvo un *pipeline* simple: Otsu + umbral local secuencial, *watershed* sobre distancia, relleno de huecos, filtrado por área mínima y corrección de concavidades mediante *convex hull*. Esta versión alcanzó F1 $\approx 70,58\%$, con una precisión de conteo del 79,00 % y un error de área medio del 0,5 %.

Quinta Fase: reimplementación con *scikit-image*. El siguiente paso fue reescribir el núcleo de segmentación usando *scipy/scikit-image*: umbralización con `filters.threshold_otsu`, limpieza morfológica con `morphology.remove_small_objects/holes`, transformada de distancia con `ndimage.distance_transform_edt`, detección de picos con `feature.peak_local_max` y *watershed* con `segmentation.watershed`. Esta fase logró el mejor F1 de todas las versiones (73,85 %) y un *Recall* muy elevado (88,38 %), pero con una sobreestimación del área (error $\approx 110\%$) y una precisión de conteo más modesta (60,46 %), debido a fusiones de núcleos cercanos.

Fase Final: postprocesado de forma y versión final. Las últimas iteraciones se centraron en afinar la forma de los núcleos y reconciliar las métricas de segmentación con las de conteo y área. Se introdujeron tres ideas principales: (i) pre-suavizado de la distancia o de la máscara para obtener marcadores más estables, (ii) selección del umbral en función de la estructura del histograma (Otsu vs Multi-Otsu según el número de modas) y (iii) postprocesado específico tras el *watershed*: unión selectiva de fragmentos que comparten una porción significativa de frontera y relleno por contorno para eliminar huecos internos sin fusionar núcleos distintos. La elección de Otsu/MultiOtsu frente a una umbralización local. La versión actual ofrece un compromiso razonado, frente a versiones anteriores maximiza el equilibrio entre detección, conteo, realismo morfológico y complejidad de implementación.

2.2 Metodología Final

1. **Preprocesado y selección de canal:** se parte del canal de Hematoxilina, donde el contraste núcleo/fondo es máximo, y se normaliza la imagen para reducir variaciones de intensidad.
2. **Obtención de máscara binaria robusta:** se analiza el histograma suavizado del canal H para decidir entre umbralización de Otsu (dos clases) o Multi-Otsu (tres clases). En el segundo caso se selecciona la clase más oscura como núcleos. La máscara resultante se limpia con operaciones morfológicas (*remove small objects/holes* y erosión ligera) para eliminar ruido y pequeños artefactos.
3. **Segmentación por *watershed* sobre distancia:** se calcula la transformada de distancia euclídea sobre la máscara binaria y se suaviza. Los máximos locales del mapa de distancia (con una distancia mínima entre picos ajustada empíricamente) definen los marcadores interiores; la máscara actúa como dominio. Sobre $-$ distancia se aplica *watershed*, separando núcleos en contacto.
4. **Postprocesado a nivel de instancia:** a partir de las etiquetas del *watershed*, se fusionan de forma controlada fragmentos que comparten una fracción de frontera por encima de un umbral relativo, y se rellenan huecos internos mediante el contorno externo de cada núcleo. Finalmente se filtran componentes por área mínima para eliminar ruido residual.
5. **Evaluación cuantitativa y cualitativa:** las máscaras finales se convierten en imágenes coloreadas y se comparan con el *Ground Truth* tanto a nivel de píxel (F1/Dice, IoU, *Precision*, *Recall*, *Accuracy*) como a nivel de conteo y área media. Además, se generan visualizaciones (contornos superpuestos, mapas de TP/FP/FN y rejillas de etapas) para una inspección cualitativa caso a caso.

Para evaluar la calidad de la segmentación se han utilizado métricas a nivel de píxel y de objeto. El **F1-Score** resume el equilibrio entre *precision* y *recall*. La **IoU** (Intersection over Union) mide el solapamiento relativo entre la máscara predicha y el *Ground Truth*. La **Precision** indica qué proporción de los píxeles etiquetados como núcleo son realmente correctos, mientras que el **Recall** indica qué fracción de los píxeles de núcleo reales se han recuperado. La **Accuracy** refleja el porcentaje total de píxeles correctamente clasificados (núcleo y fondo).

Además, se analiza el **conteo de núcleos**, comparando el número medio de núcleos reales (GT) con el número medio de núcleos predichos y su precisión de conteo. Finalmente, se estudia el **área media** de los núcleos (en píxeles cuadrados) y la diferencia relativa entre el área media de la predicción y la del *Ground Truth*, como indicador de realismo morfológico.

Table 1: Resumen de métricas globales para la versión final.

Tipo	Métrica	Valor
Segmentación	F1-Score	71.42%
Segmentación	IoU	56.41%
Segmentación	Precision	72.74%
Segmentación	Recall	74.40%
Segmentación	Accuracy	85.70%
Conteo	Núcleos GT (media)	723.8
Conteo	Núcleos predichos (media)	561.3
Conteo	Precision de conteo	71.13%
Área	Área media GT	463.47 px ²
Área	Área media predicha	523.91 px ²
Área	Diferencia absoluta	60.44 px ²
Área	Diferencia relativa	13.0%

Soluciones adoptadas, evaluación y mejoras pendientes

A pesar de estos avances, se identifican posibles líneas de mejora: ajuste automático de parámetros por imagen (por ejemplo, `min_distance` o umbrales relativos de contacto), incorporación de modelos de aprendizaje profundo (*U-Net*, etc.) como etapa inicial de segmentación sobre la que aplicar después el *watershed*, o una evaluación adicional basada en *matching* de instancias que combine información geométrica (centroides, área, forma) para caracterizar mejor fusiones y fragmentaciones. Estas extensiones, sin embargo, se consideraron fuera del alcance de la práctica y se dejan como trabajo futuro.