UNIVERSITAT POLITÈCNICA DE VALÈNCIA

ESCUELA TÉCNICA SUPERIOR INGENIEROS INDUSTRIALES VALENCIA

Curso Académico:

AGRADECIMIENTOS

A Valery, por permitirme la posibilidad de trabajar con ella,
por enseñarme y motivarme, por su amistad
y por su confianza depositada en mí.

A Fran y a todo el grupo de CVBLab, por su apoyo, por sus consejos
y por su disposición a ayudarme en todo momento.

A mi familia, por debérselo todo, pues soy quien soy
y estoy donde estoy gracias a ella.

A mis amigos, por enseñarme a disfrutar de los pequeños placeres
y por tener la cura de todos los problemas.

A Zaida, por creer siempre en mí y conseguir que
todo sea más fácil a su lado.

A mis compañeros del equipo de fútbol, por encontrar en ellos
la ambición y la competitividad, además de
una vía de escape para el estudio.

RESUMEN

El presente TFG pretende abordar el desarrollo de un sistema de segmentación automática de glándulas en imágenes histológicas de tejido prostático sano, como una primera fase del proyecto nacional SICAP, cuyo objetivo final es proporcionar una herramienta que aporte nuevo conocimiento para la mejora, a corto y medio plazo, del diagnóstico precoz del cáncer de próstata. Para encarar ese objetivo final, se implementa una metodología basada en una clasificación no supervisada que posibilita la agrupación de los distintos tejidos presentes en la imagen en su *cluster* correspondiente. Por otra parte, se lleva a cabo una clasificación supervisada, con un etiquetado y una extracción de características de las muestras, a fin de poder discriminar los elementos de la imagen que forman parte de glándulas prostáticas y los que no. De esta forma, se emprende la tarea de segmentación a partir de dichos objetos, utilizando una técnica conocida como "*Locally Constrained Watershed Transform",* definida como una variante de la tradicional *"watershed* con marcadores".

Con la intención de encarar líneas venideras de investigación, en este proyecto, además de detallar las diversas problemáticas encontradas y las posibles soluciones para la mejora de los algoritmos, se exponen también unos resultados que permiten evaluar el sistema propuesto. Estos resultados radican en: (i) la elección del mejor clasificador para entrenar las características extraídas de las muestras, (ii) la predicción realizada para la clasificación de nuevas imágenes histológicas, y (iii) la segmentación automática implementada como finalidad última del trabajo. Los resultados obtenidos aportarán nueva información para contribuir al cumplimiento del objetivo final, el cual reside en el desarrollo de un sistema de ayuda al diagnóstico precoz del cáncer de próstata.

Palabras clave: cáncer de próstata, imágenes histológicas, *clustering*, clasificación supervisada, segmentación automática, *constrained watershed.*

RESUM

El present TFG pretén abordar el desenvolupament d'un sistema de segmentació automàtica de glàndules en imatges histològiques de teixit prostàtic sà, com una primera fase del projecte nacional SICAP, l'objectiu final del qual és proporcionar una ferramenta que aporte nou coneixement per la millora, a curt i mig termini, del diagnòstic precoç del càncer de pròstata. Per encarar aquest objectiu final, s'implementa una metodologia basada en una classificació no supervisada que possibilita l'agrupació de diversos teixits presents a la imatge en el *clúster* corresponent. D'altra banda, es realitza una classificació supervisada, amb un etiquetatge i una extracció de característiques de les mostres, a fi de poder discriminar els elements de la imatge que formen part de les glàndules prostàtiques i els que no. D'aquesta manera, s'emprén la tasca de segmentació a partir de aquests objectes, utilitzant una tècnica coneguda com "*Locally Constrained Watershed Transform*", definida com una variant de la tradicional "*Watershed* amb marcadors".

Amb la intenció d'encarar línies futures de la investigació, en aquest projecte, a més de detallar les diverses problemàtiques trobades i les possibles solucions per la millora dels algoritmes, s'exposen també uns resultats que permeten avaluar el sistema proposat. Aquests resultats radiquen en: (i) l'elecció del millor classificador per a entrenar les característiques extretes de les mostres, (ii) la predicció realitzada per la classificació de noves imatges histològiques, i (iii) la segmentació automàtica implementada com a última finalitat del treball. Els resultats obtinguts aportaran nova informació per contribuir a l'acompliment de l'objectiu final, el qual resideix en el desenvolupament d'un sistema d'ajuda al diagnòstic precoç de càncer de pròstata.

Paraules clau: càncer de pròstata, imatges histològiques, *clustering*, classificació supervisada, segmentació automàtica, *constrained watershed.*

ABSTRACT

This TFG aims to approach the development of an automatic glands segmentation system on histological images of prostatic healthy tissue, as a first step for the national SICAP project, of which the final objective is to develop a tool that provides knowledge for the improvement, in the short and medium term, of early diagnosis of prostatic cancer. In order to face this final objective, a methodology based on non-supervised classification that allows grouping different tissues of the image in their corresponding cluster is implemented. Additionally, supervised classification is performed, with labelling and extraction of sample characteristics, with the objective of being able to discriminate between elements of the images that are part of prostatic glands and those that are not. This way, the task of segmentation is started from these objects, using a technique known as "Locally Constrained Watershed Transform", defined as a variant of the classic "Watershed With Markers".

With the intention of facing future lines of research, in this project, aside from detailing different problems found and possible solutions for the improvement of the algorithms, results are presented that allow for evaluation of the proposed system. These results consist of: (i) the choice of the best classifier to train the features extracted from the samples, (ii) the prediction given for the classification of new histological images, and (iii) automatic segmentation implemented as the final objective of this work. The obtained results will give new information that contributes to the accomplishment of the final objective of SICAP, which consists on the development of an early diagnosis guidance system for prostatic cancer.

Keywords: prostatic cancer, histological images, clustering, supervised classification, automatic segmentation, constrained watershed.

ÍNDICE GENERAL

I. Memoria
II. Presupuesto

I. MEMORIA

ÍNDICE DE LA MEMORIA

1. Introducción ... 1

1.1 Motivación y descripción del problema ... 2

1.1.1 Cáncer ... 2

1.1.2 Cáncer de próstata ... 2

1.1.3 Imágenes histopatológicas ... 6

1.1.4 Proyecto SICAP ... 10

1.2 Objetivos ... 10

1.3 Guía del trabajo ... 12

2. Material y Métodos ... 13

2.1 Introducción ... 14

2.2 Material ... 15

2.2.1 Base de datos de imágenes histológicas ... 15

2.2.2 Software utilizado ... 17

2.2.3 Hardware utilizado ... 17

2.3 Métodos ... 18

2.3.1 Diagrama del sistema y lista de funciones ... 18

2.3.2 Creación de una base de datos de imágenes de alta resolución ... 20

2.3.3 Clasificación de tejidos ... 23

2.3.4 Clasificación de lúmenes ... 30

2.3.5 Segmentación automática de glándulas ... 53

2.3.6 Segmentación manual ... 60

3. Resultados y discusión ... 61

3.1 Clasificación ... 62

3.1.1 Resultados del mejor clasificador ... 62

3.1.2 Resultados de la clasificación ... 64

3.2 Segmentación ... 67

3.2.1 Resultados cuantitativos ... 67

3.2.2 Visualización de resultados ... 69

4. Conclusiones y líneas futuras ... 71

4.1 Conclusiones ... 72

4.2 Líneas futuras ... 74

CAPÍTULO 1

1. Introducción

Índice de contenidos

1.1 Motivación y descripción del problema 2

1.1.1 Cáncer 2

1.1.2 Cáncer de próstata 2

1.1.3 Imágenes histopatológicas 6

1.1.4 Proyecto SICAP 10

1.2 Objetivos 10

1.3 Guía de la Memoria 12

Este Trabajo Fin de Grado (TFG) se ha llevado a cabo para colaborar en uno de los proyectos en los que actualmente está trabajando el grupo CVBLab (Computer Vision and Behavoir Analysis Lab), perteneciente al I3B (Instituto de Investigación e Innovación en Bioingeniería) de la Universitat Politècnica de València.

En concreto, este trabajo de investigación es parte del proyecto nacional SICAP (Sistema de interpretación de imágenes histopatológicas para la detección del cáncer de próstata) subvencionado por el Ministerio de Economía, Industria y Competitividad, en el que el grupo CVBLab es colaborador junto con la Universidad de Granada y el Servicio de Anatomía Patológica del Hospital Clínico Universitario de Valencia.

1.1 Motivación y descripción del problema

1.1.1 Cáncer

Según la Organización Mundial de la Salud (OMS), el "cáncer" es un proceso de crecimiento de células incontrolado que puede afectar prácticamente a cualquier parte del organismo [1]. Es una de las principales causas de morbilidad y mortalidad en el mundo, y se prevé que el número de incidencias aumente aproximadamente en un 70% durante los próximos 20 años [2]. Uno de los consejos más recomendables para prevenir el cáncer es evitar la exposición a factores de riesgo como pueden ser: la falta de actividad física, el consumo de tabaco y el consumo de alcohol. Lógicamente, esto no asegura el hecho de no padecer cáncer, ya que también hay que tener en cuenta factores no controlables como el genético. No obstante, un porcentaje importante de los enfermos que tienen cáncer puede curarse mediante cirugía, radioterapia o quimioterapia, sobre todo si se detecta en fase temprana. El problema es que la detección del cáncer en una fase avanzada y la falta de diagnóstico y tratamiento son, desafortunadamente, demasiado frecuentes. Por ello, lo que resulta interesante en este trabajo es esa parte de prevención, detección temprana y diagnóstico precoz que permita la elaboración de un sistema de apoyo al diagnóstico con el que, a la larga, se haga factible la reducción de la mortalidad.

A continuación se explica en mayor profundidad el cáncer de próstata atendiendo a diferentes estadísticas relacionadas con la incidencia, el número de casos y de muertes estimadas para 2017 y otros factores como los tratamientos o los análisis para el diagnóstico.

1.1.2 Cáncer de próstata

El cáncer de próstata es uno de los tipos de cáncer más comunes en el ser humano junto con el cáncer de mama y el de pulmón en EEUU, tal y como revela el Instituto Nacional de Salud de los EEUU (NIH) en sus estudios más recientes (publicados en 2017) [3]. Esto se puede ver en las tablas que se muestran a continuación, donde aparecen los números estimados de casos nuevos y muertes para los diez tipos de cáncer más frecuentes.

Tipos de cáncer	Número de casos estimados
Mama	252710
Pulmón	222500
Próstata	**161360**
Colon y recto	135430
Melanoma	87110
Vejiga	79030
Linfoma No-Hodgkin	72240
Riñón	63900
Leucemia (todos los tipos)	62130
Páncreas	53670

Tabla 1: Número estimado de casos nuevos en 2017 en EEUU para los diez tipos de cáncer [3] .

La tabla 2, procede de la misma fuente. En ella se observa el número de muertes estimadas para los mismos tipos de cáncer ordenados por orden decreciente, con respecto al parámetro de estudio, igual que en la tabla anterior.

Tipos de cáncer	Número de muertes estimadas
Pulmón	155870
Colon y recto	50260
Páncreas	43090
Mama	40610
Próstata	**26730**
Leucemia (todos los tipos)	24500
Linfoma No-Hodgkin	20140
Vejiga	16870
Riñón	14400
Melanoma	9730

Tabla 2: Número de muertes estimadas en 2017 en EEUU para los diez tipos de cáncer.

Según los datos estadísticos más recientes del programa *Surveillance, Epidemiology and End Results del NCI* la edad media de un diagnóstico de cáncer es de 66 años, tal y como se muestra en la figura 1.1, donde se recoge el porcentaje de casos nuevos de todos los tipos de cáncer, teniendo en cuenta todas las razas y ambos sexos, para la población de U.S.

Concretamente, el cáncer de próstata es la causa más común de muerte por cáncer en hombres mayores de 70 años de edad, pues rara vez se encuentra en hombres menores de 40. Además, la edad media al momento del diagnóstico es también de 66 años.

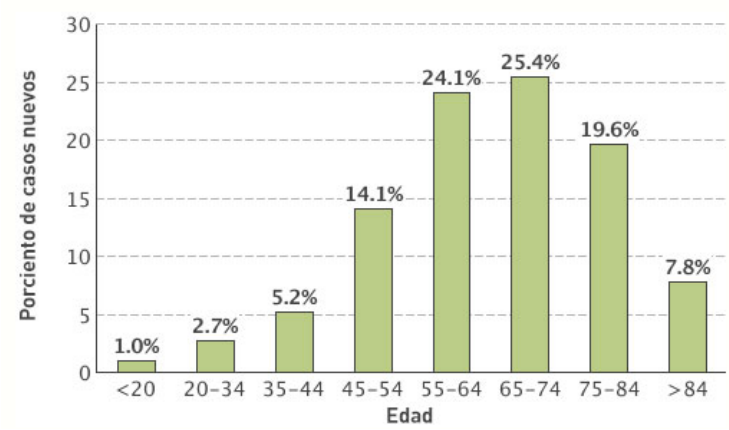


Figura 1.1: Porcentaje de casos nuevos de cáncer por edad para la población de U.S. [4].

Hoy en día, la tasa de supervivencia para los hombres con cáncer de próstata, tras 5 años, es del 99%. El 98% también vive después de 10 años, y aproximadamente el 95% vive, al menos, 15 años. Otra estadística llamativa es que, por razones todavía desconocidas, el riesgo de cáncer de próstata es un 70% mayor en las personas de raza negra que en las personas de raza blanca no hispanas. La mayoría de los casos de cáncer de próstata (92%) se detectan cuando la enfermedad está ubicada en la glándula prostática o en los órganos adyacentes. Esto se conoce como el estadio local o enfermedad localmente avanzada [5].

En la figura 1.2 se puede apreciar la diferencia entre una próstata sana y una próstata enferma. Los rasgos distintivos hacen referencia principalmente al aumento del tamaño del órgano en el caso del cáncer, lo cual conlleva una oclusión en el conducto urinario que provoca a su vez una retención de la orina en la vejiga.

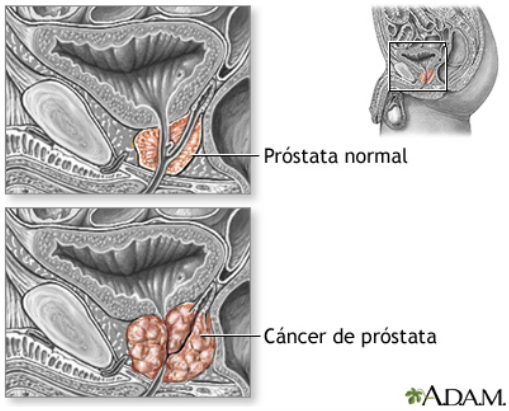


Figura 1.2: Diferencia entre próstata sana y próstata con cáncer [6].

Los métodos de tratamiento prescritos para el cáncer de próstata incluyen: cirugía, radioterapia, vigilancia activa, criocirugía, tratamiento de vacunas, terapia hormonal y quimioterapia [7]. Normalmente, estos tratamientos son afrontados de manera individual, pero en ocasiones se combinan para facilitar el diagnóstico. En este diagnóstico están involucrados los procedimientos de la examinación rectal, el análisis sanguíneo, la biopsia[1] del tejido y el test de imagen. Además, gracias a la disponibilidad de diferentes tecnologías y dispositivos como la tomografía computarizada (CT) y la imagen por resonancia magnética (MRI) es posible generar una gran cantidad de imágenes. En la actualidad, estas tecnologías están empezando a cobrar importancia en el diagnóstico, pero las técnicas que realmente son importantes para diagnosticar el cáncer son el análisis de la PSA[2] y especialmente la biopsia. Concretamente, a partir de la biopsia, y tras un proceso de preparación de las muestras, se hace posible el diseño de sistemas automáticos, basados en procesado de imagen, reconocimiento de patrones y algoritmos de *machine learning*[3], como el desarrollado en este TFG.

Por otra parte, debido a la indefinición de los síntomas de esta enfermedad, el diagnóstico para el cáncer de próstata es realmente complicado, llevando a los patólogos en muchas ocasiones a diferir en sus conclusiones. Además, requiere de múltiples procedimientos. Entre ellos, el más importante es el análisis del tejido prostático, obtenido de la biopsia, para detectar la presencia de las regiones cancerígenas en el tejido y asignar un tipo de puntuación conocida como *Gleason Score* (figura 1.7) que determina la gravedad del cáncer en esa muestra.

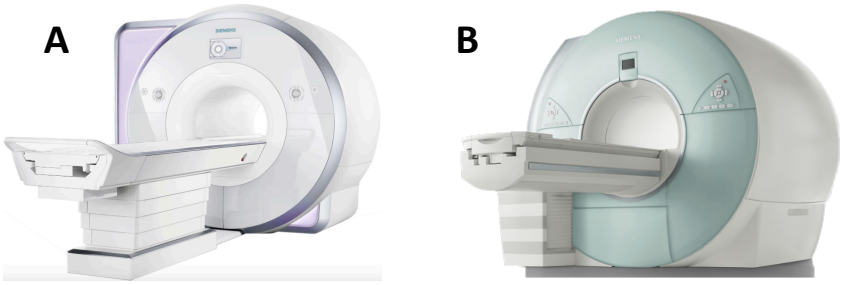


Figura 1.3: A) Equipo de CT. B) Equipo de MRI [8], [9].

Este TFG, se ha enfocado básicamente desde el punto de vista de los sistemas CAD (Computed-Aided Diagnosis) que analizan las imágenes histopatológicas de la próstata, es decir, las imágenes digitalizadas del tejido prostático obtenidas a partir de la biopsia. En la figura 1.5 se puede observar un ejemplo de lo expuesto, donde se hace referencia también al tratamiento que recibe la imagen antes de ser escaneada para su posterior análisis.

[1] Biopsia. Examen microscópico de un trozo de tejido o una parte de líquido orgánico que se extrae de un ser vivo con el objetivo de analizar diversos componentes presentes en la muestra.

[2] PSA son las iniciales de *Prostate Specific Antigen*. El análisis de PSA se utiliza para medir la concentración de este antígeno en la sangre, de tal forma que una alta concentración suele indicar que hay cáncer de próstata.

[3] *Machine learning* se traduce como “aprendizaje automático” y es el sub-campo de una rama de la inteligencia artificial cuyo objetivo es desarrollar programas capaces de generalizar comportamientos que permitan a los ordenadores aprender a partir de ejemplos.

1.1.3 Imágenes histopatológicas

La histología es la parte biológica que estudia la composición, la estructura y las características de los tejidos orgánicos de los seres vivos. Por ello, con las imágenes histopatológicas lo que se busca es un diagnóstico mediante el análisis microscópico de biopsias o muestras del tejido.

Para la preparación del tejido y su digitalización, la biopsia es procesada con un fijador para prevenir que se descomponga. Después, se embebe la muestra en parafina para darle rigidez y poder realizar un corte adecuado. A continuación, el tejido es seccionado en cortes usando un micrótomo (una máquina que puede crear cortes muy delgados) y se coloca en los portaobjetos antes de ser teñido con el pigmento correspondiente para revelar los componentes del tejido.

En el tipo de imágenes con las que se trabaja en este proyecto (cortes histopatológicos de muestras de próstata) se utiliza un escáner para captar la información celular disponible en el tejido, pero antes del escaneado se usa el pigmento de hematoxilina y eosina (H&E) que permite distinguir los componentes tisulares individualmente.

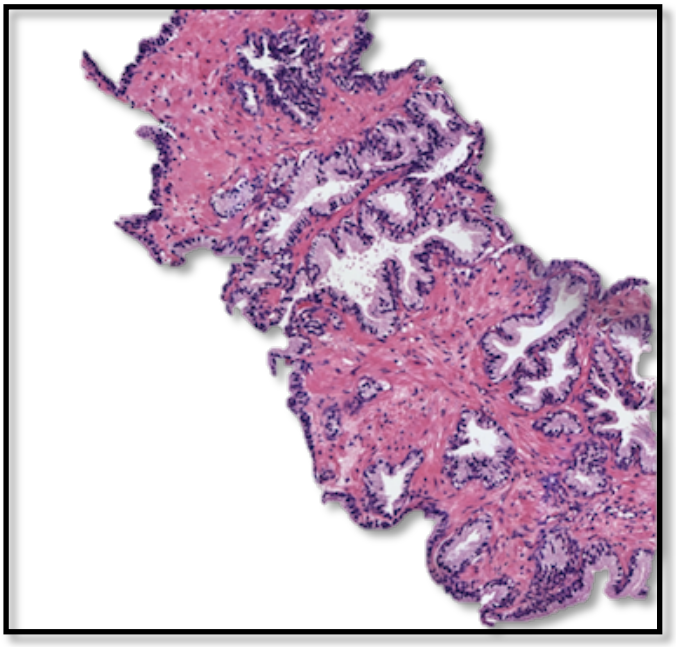


Figura 1.4: Imagen histopatológica de una muestra de tejido prostático.

Este tejido se ha teñido utilizando la conocida técnica de tinción de H&E. Las áreas rosas corresponden a las regiones de un tejido conjuntivo extracelular llamado estroma, cuya función es servir de sostén a las células. Las zonas azules (más oscuras) y las púrpuras (más claras) corresponden a los núcleos y al citoplasma, respectivamente. Por otra parte, la región de color blanca rodeada por los núcleos y el citoplasma se conoce como lumen o luz. En la figura 1.5 se muestran claramente los cuatro componentes principales de este tipo de imágenes.

El tamaño original de la imagen que se visualiza en la figura 1.4 es de 4096x4096x3 píxeles, correspondientes a una resolución espacial de 0,2 micrómetros por píxel y a un objetivo de 40x.

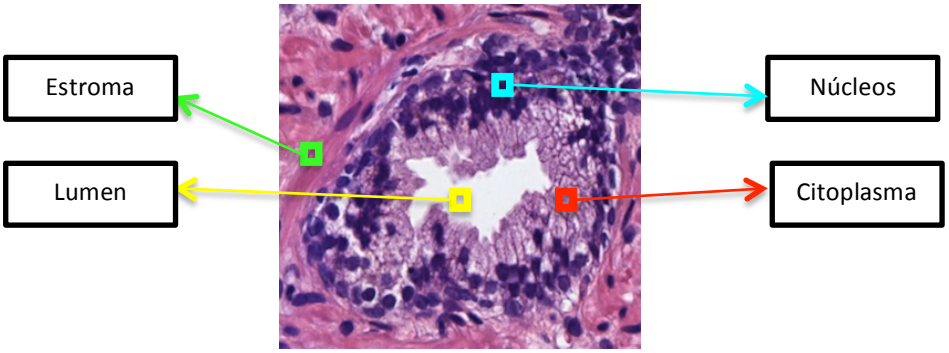


Figura 1.5: Componentes diferenciables en una imagen histopatológica de próstata, donde se considera como glándula aquella parte del tejido en la que aparecen: lumen, citoplasma y núcleos.

El procedimiento en la preparación del corte de tejido para la observación al microscopio se muestra en el siguiente esquema:

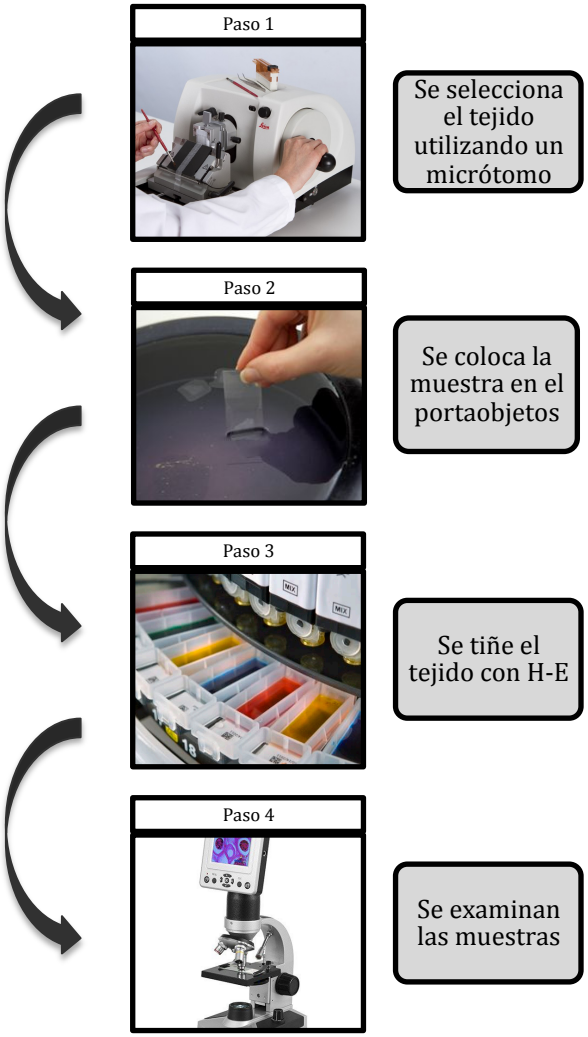


Figura 1.6: Procedimiento para la preparación del corte histológico [10]–[13] *. La muestra es examinada directamente con un microscopio o digitalizada y procesada en un ordenador.*

La detección y los procesos de clasificación están basados tanto en las estructuras glandulares como en las propiedades citológicas del tejido. En los análisis rutinarios, los patólogos[4] tienen que buscar en el tejido biopsiado utilizando un microscopio y recorriendo la muestra con el fin de encontrar alguna zona que presente información relevante. Esto conlleva un tiempo muy valioso que ralentiza el proceso y lo hace ineficiente. Por esta razón, se está avanzando mucho en el desarrollo de la patología digital, la cual permite obtener imágenes de tejidos a partir de la muestra física. Estas imágenes pueden ser visualizadas en un monitor y anotadas mediante herramientas software por los especialistas que analizan y clasifican las muestras.

El objetivo de esta clasificación histopatológica es, como ya se ha comentado antes, determinar la gravedad del desarrollo de la enfermedad en el tejido y para ello se lleva a cabo un proceso que incluye tres pasos fundamentales: (i) detectar las regiones del tejido que presentan cáncer, (ii) asignar una puntuación de Gleason a cada región y (iii) atribuir una suma de Gleason al conjunto tisular.

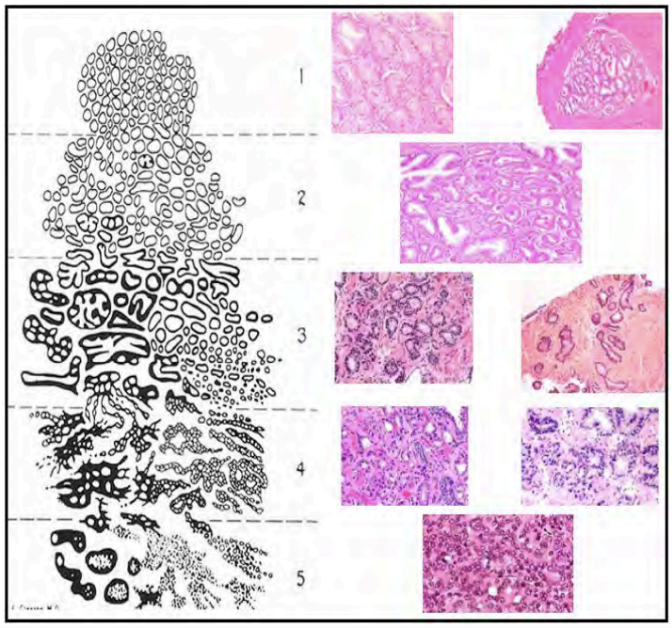


Figura 1.7: Clasificación de Gleason para el cáncer de próstata, desde el menos agresivo (grado 1 o normal) hasta el más agresivo (grado 5) [14].

Los tres pasos mencionados se describen a continuación:

i) Cuando el patólogo examina la muestra, primero comienza con un aumento pequeño (1x o 2x) para localizar las regiones de interés (regiones con glándulas anormales) en la imagen. Después incrementa el zoom, a 5x o 10x, para analizar la estructura de las glándulas y su distribución. Posteriormente, si el patólogo no observa una clara evidencia de lo que busca, debe incrementar de nuevo el objetivo, a 20x o 40x, para examinar la información citológica en el tejido.

[4] Patólogo. Doctor que tiene una formación especial para identificar las enfermedades mediante el estudio de las células o los tejidos con un microscopio.

ii) El patólogo asigna una puntuación de Gleason a cada región de interés detectada en el primer paso. El método de *Gleason grading* define cinco grados diferentes correspondientes a cinco patrones distintos, como se muestra en la figura 1.7. Los grados 1 y 2 hacen referencia a tejidos que siguen patrones muy similares al del tejido sano. Por otra parte, el grado 3 todavía sigue un patrón que, en ciertos casos se parece al tejido normal, pero ya hay indicios de rotura de glándulas. Sin embargo, en las imágenes correspondientes al grado 4, y sobre todo al 5, las glándulas tienden a fusionarse con las vecinas y comienza a haber una acumulación de núcleos, apreciándose cada vez menos estructuras como el lumen y, en grados muy avanzados, el citoplasma.

iii) Cabe destacar que normalmente los cánceres de próstata contienen partes de tejido que presentan grados diferentes, por lo que la puntuación de Gleason se determina añadiendo los dos grados mayoritarios [15], [16]. Por ejemplo, si el grado más común de una muestra de tejido es grado 3, seguido del grado 4, la puntuación de Gleason para esa muestra será de 3+4 = 7. No obstante, conviene señalar que se puede obtener la misma puntuación final, (el mismo *Gleason sum* [5]) pero como resultado de la suma inversa, es decir, 4+3 = 7 en caso de que el grado más común sea el 4 seguido del 3. Esta diferencia es reseñable porque para el primer ejemplo se hablaría de un cáncer menos severo que en el segundo y el tratamiento prescrito podría ser totalmente diferente. Por este motivo es más específico identificar el grado del cáncer con dos números en vez de con uno.

En resumen, con esta forma de calificación, actualmente la puntuación más baja es de grado 6, por lo que las puntuaciones por debajo de ese valor corresponderán a muestras de tejido normales o casi normales. Como extremo superior, las de grado 9 y 10 corresponden a muestras con un cáncer muy avanzado.

Aunque la biopsia solo representa una porción de la próstata, la suma de Gleason sigue siendo uno de los factores más importantes en el pronóstico, ya que puede ayudar a los patólogos a determinar el tratamiento más apropiado. También está recomendado que la biopsia sea examinada por más de un patólogo para mejorar la precisión en el diagnóstico y tener en consideración el error humano. Esta parte cobra una importante relevancia, pues el procedimiento de análisis de muestras es un trabajo tedioso que supone un gran desafío para los patólogos a la hora de aportar un diagnóstico de forma precisa y eficiente. Por tanto, se hace visible en este punto la necesidad de desarrollar mecanismos que permitan un procesado automático de imágenes de tejidos de próstata que puedan ayudar a los patólogos a la toma de decisiones y a mejorar su rendimiento.

En este proyecto nacional, lo que se pretende conseguir es un método que facilite a los patólogos los pasos (i) y (ii) mencionados anteriormente, con el objetivo de que después, dichos especialistas puedan determinar mediante esta herramienta cuál es la puntuación final de Gleason para el tejido. Concretamente en este TFG, como una de las tareas principales de la primera fase del proyecto, se va a analizar un conjunto de imágenes no malignas (calificadas con una puntuación de Gleason menor de 6) con la finalidad de detectar y segmentar las glándulas que aparezcan en cada una de esas imágenes, aplicando técnicas de clasificación y segmentación.

[5] "*Gleason sum"* también es conocido como "*Gleason Score*" en algunos estudios.

1.1.4 Proyecto SICAP

SICAP es el acrónimo de "Sistema de interpretación de imágenes histopatológicas para la detección del cáncer de próstata". Lo que se pretende con SICAP principalmente es, haciendo uso de una base de datos privada adquirida por los investigadores del Hospital Clínico Universitario de Valencia, trabajar en la segmentación automática y en la extracción de rasgos basados tanto en la estructura de los tejidos (ayudándose de la experiencia de los médicos) como en las técnicas de filtrado (sin necesidad de apoyo médico). Estas características extraídas serán utilizadas como entrada de clasificadores que permitirán la detección del cáncer y su valoración atendiendo a la escala Gleason, explicada anteriormente. Por otra parte, también se implementarán técnicas de aprendizaje profundo basadas en *Deep Learning (DP)*[6] para la tarea de clasificación. Cabe destacar que en este proyecto se tendrá que lidiar con ciertos problemas que presenta la tecnología *Whole-Slide Image*[7] (WSI) relacionados con las dimensiones de las imágenes, ya que se trabaja con unas dimensiones de tamaño considerablemente grandes.

En conclusión, SICAP es un proyecto compuesto por un equipo coordinado y multidisciplinar en el que colaboran médicos, ingenieros y científicos, procedentes de distintas universidades y centros de investigación con la finalidad de proporcionar un software (con vistas a la integración en la empresa) para el diagnóstico del cáncer. Además, se busca mejorar la interpretación de las tecnologías WSI y conseguir diseñar un sistema de ayuda al diagnóstico.

Concretamente, en este TFG se pretende llevar a cabo una de las principales tareas de la primera fase correspondiente a la segmentación de las imágenes histopatológicas del tejido prostático con carácter de no malignidad.

1.2 Objetivos

El fin último del presente trabajo es la segmentación de glándulas en las imágenes procedentes de muestras histopatológicas de las biopsias de la próstata. La idea es poder diferenciar, en una primera instancia, los rasgos característicos de un tejido sano que van a permitir, posteriormente, discriminar regiones en función de si presentan cáncer, o no.

Actualmente, el proyecto SICAP, que comenzó en enero de 2017, se encuentra en sus primeras fases, ya que todavía se están valorando las diferentes maneras de encarar el problema de la detección y la segmentación de imágenes con diferentes grados de cáncer. Por tanto, en esta etapa previa, únicamente se llevará a cabo el proceso completo en las imágenes que han sido etiquetadas por los especialistas con una puntuación de Gleason no mayor de 6.

---

[6] *Deep Learning* es un conjunto de algoritmos de "*machine learning*" (aprendizaje automático) que permite utilizar modelos computacionales que están compuestos de múltiples capas de procesamiento para aprender representaciones de datos con múltiples niveles de abstracción [17].

[7] *Whole-Slide Image* es una tecnología que ha permitido que las imágenes digitales de alta resolución de muestras completas de biopsias de tejidos se hayan convertido en una práctica clínica cada vez más habitual en el diagnóstico del cáncer de próstata.

Cabe destacar que la forma de proceder con las imágenes que presentan cáncer es en general muy similar, sobre todo en los primeros pasos implementados en este proyecto. Sin embargo, la extracción de características deberá enfocarse de otra manera, pues los rasgos que presentan las imágenes de glándulas prostáticas con cáncer es muy diferente a la que presentan las de tejido sano.

Los objetivos que se plantean en este trabajo de investigación son los siguientes:

1. Revisar la literatura científica actual para conocer el estado del arte en la segmentación de las imágenes histopatológicas teñidas con hematoxilina y eosina (H&E). Principalmente ha de considerarse lo referente a la próstata, pero también puede ser útil la información relacionada con imágenes histológicas de otros tipos de cáncer como el de colon por ejemplo, para incorporar ideas aplicables a la programación de código en MATLAB.
2. Conocer y comprender el método de obtención de las imágenes con distintos valores en cuanto a resolución espacial y tamaño de imagen se refiere, e implementar un nuevo código que permita combinar dichos factores sin comprometer ninguno de ellos al optimizar el otro. Esto es de vital importancia, ya que en función de ello se definirán todos los parámetros que conformarán la base del resultado final del trabajo.
3. Estudiar de manera pormenorizada distintas técnicas de *clustering* (clasificación no supervisada) que permitirá obtener una buena clasificación, y con ello un buen análisis, de los distintos componentes que se revelan en las imágenes histopatológicas del tejido de la próstata al utilizar la tinción (H&E). Estos componentes son el lumen, el citoplasma, los núcleos y el estroma.
4. Detección de los objetos que corresponden a estructuras glandulares en las imágenes de H&E con distintos clasificadores para obtener los mejores resultados posibles. Este apartado incluye:
   4.1. La creación de dos bases de datos de imágenes correspondientes a:
      4.1.1. Las imágenes que servirán para entrenar el modelo (*Training*).
      4.1.2. Las imágenes con las que se va a testear el modelo (*Test*).
   4.2. Además, se deberá generar una tercera base de datos de imágenes donde las glándulas (tanto de las imágenes de *training* como de las de *test)* hayan sido detectadas manual o semi-manualmente para poder establecer correspondencias en las características, a partir de las muestras etiquetadas, y evaluar los resultados obtenidos del algoritmo de detección.
5. Abordar la segmentación de las criptas presentes en las muestras histopatológicas valorando el mejor método para conseguirlo. Además, la segmentación incluirá en otro color aquellos objetos cuya etiqueta sea más compleja de asignar, incluso en la valoración humana, con la finalidad de dotar al programa de la mayor precisión y detalle posible y que el especialista pueda determinar, en base a sus conocimientos, la clase correcta a la que pertenecen dichos objetos.
6. Preparar las imágenes en el formato necesario para poder realizar una segmentación manual y obtener así un *ground truth*[8] que permita evaluar y comparar los resultados conseguidos tras la ejecución del programa con los valores obtenidos de la segmentación manual.
7. Identificar y describir los problemas y limitaciones encontrados en el estudio.
8. Proponer posibles mejoras para investigaciones futuras abordando los problemas detectados y teniendo en mente los próximos objetivos del proyecto SICAP, del cual forma parte este TFG.

[8] *Ground truth* hace referencia a las imágenes que han sido segmentadas manualmente por un experto. Estas imágenes son las que se utilizan para comparar, procurando que los resultados proporcionados por el programa se parezcan lo máximo posible a ellas.

1.3 Guía de la Memoria

En el capítulo 2 se exponen los conceptos teóricos que hacen referencia al material utilizado y a la metodología implementada para conseguir la clasificación de los objetos, y a partir de los cuales realizar la segmentación automática de las glándulas de próstata sana como fin último del proyecto. Además, se resume la información más importante relacionada con el estado del arte, la cual se ha obtenido mediante revisión bibliográfica. Finalmente, se presentan los datos de mayor relevancia sobre las muestras de imágenes disponibles.

En el capítulo 3 se exponen y se discuten los resultados obtenidos con las distintas técnicas propuestas para la selección del mejor clasificador, la clasificación supervisada de posibles lúmenes y la segmentación automática de las estructuras glandulares.

En el capítulo 4 se recopilan las conclusiones más importantes que se han extraído de los estudios realizados y se detallan algunas propuestas como posibles líneas futuras de investigación.

CAPÍTULO 2

2. Material y Métodos

Índice de contenidos

2.1 Introducción 14

2.2 Material 15

2.2.1 Base de datos de imágenes histológicas 15

2.2.2 Software utilizado 17

2.2.3 Hardware utilizado 17

2.3 Métodos 18

2.3.1 Diagrama del sistema y lista de funciones 18

2.3.2 Creación de una base de datos de imágenes de alta resolución 20

2.3.3 Clasificación de tejidos 23

2.3.3.1 *Clustering* 24

2.3.4 Clasificación de lúmenes 30

2.3.4.1 Creación del *ground truth* para la clasificación 32

2.3.4.2 Clasificación supervisada 35

2.3.4.3 Clasificación y predicción 50

2.3.5 Segmentación automática de glándulas 53

2.3.5.1 Watershed 53

2.3.5.2 Watershed con marcadores 54

2.3.5.3 Constrained Watershed 55

2.3.6 Segmentación manual de glándulas 60

2.1 Introducción

En primer lugar, es necesario mencionar el trabajo que realizan los patólogos para poder comprender la dinámica del funcionamiento del proyecto. Al inicio del proceso, los patólogos parten de una determinada cantidad de imágenes histológicas, procedentes de las biopsias, que están disponibles en una base de datos privada. La privacidad de la base de datos tiene la finalidad de asegurar en todo momento la confidencialidad de los pacientes. De esta forma se preserva tanto la protección de los derechos, la seguridad y el bienestar de los sujetos que participan en el proyecto, como la garantía pública. Una vez se tienen al alcance todas las imágenes de las biopsias, los patólogos utilizan una aplicación llamada "*MicroDraw*" con la que pueden anotar las muestras histológicas. Cabe destacar que *MicroDraw* es una aplicación desarrollada sobre una librería de código abierto que ha sido modificada por los ingenieros del CVBLab, conjuntamente con los patólogos, con la finalidad de que estos puedan anotar los grados de patología en las imágenes y, de esa forma, el sistema automático pueda aprender a partir de ejemplos.

En cuanto al proceso de adquisición de las imágenes, estas son escaneadas y capturadas mediante un microscopio que permite exportarlas en dos formatos: **.bif* y **.tiff*. Concretamente, el microscopio empleado para la adquisición de las imágenes es el disponible en el Hospital Clínico de Valencia, entidad colaboradora en el proyecto SICAP. Este microscopio permite escanear diferentes laminillas de tejido al mismo tiempo a 20 y 40 aumentos (20x y 40x). Una laminilla puede contener varias imágenes completas. (Ver Figura 2.1.A).

Para el proyecto se trabajará con imágenes adquiridas a 40x utilizando varios puntos de enfoque para evitar que algunas zonas se vean borrosas al no estar bien enfocadas. Una imagen de esas características puede tener un tamaño del orden de [50000, 90000] x [90000, 190000] píxeles y puede ocupar desde 300 Mb hasta 1,5 Gb, dependiendo del tamaño exacto. Esto es uno de los objetivos, ya mencionados, en los que se deberá trabajar para afrontar los problemas relacionados con la resolución espacial y el tamaño de las imágenes. Para exportarlas, se utilizará el formato **.tiff*, ya que al visualizar las imágenes **.bif* aparece un "error de cosido" que causa un desplazamiento vertical en algunos tramos de la imagen. (Ver Figura 2.1.B).

Una vez conocida la forma de adquisición de las imágenes, los patólogos pueden hacer uso de la aplicación *MicroDraw* que consiste en una plataforma web especializada en la visualización y el etiquetado de las mismas. Se establece así una comunicación bidireccional entre los expertos en el campo de la medicina y de la ingeniería, pretendiendo conseguir un mismo objetivo.

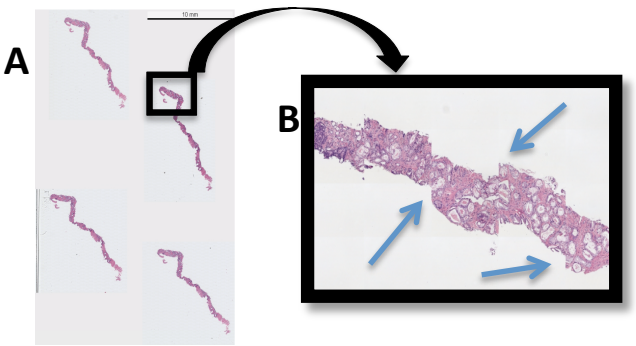


*Figura 2.1: A) Cuatro muestras de tejido prostático contenidas en una misma laminilla. B) Error de cosido en las imágenes *.bif.*

2.2 Material

2.2.1 Base de datos de imágenes histológicas

Una vez los patólogos llevan a cabo las anotaciones pertinentes sobre las imágenes procedentes de las biopsias, el siguiente paso es generar una base de datos privada donde las imágenes estén etiquetadas en función de si son sanas o enfermas. En el caso de las enfermas, se profundiza también en el grado de cáncer que presenta cada muestra, atendiendo a la clasificación de Gleason.

Por tanto, hasta aquí se dispone de una determinada cantidad de imágenes histopatológicas en las que se diferencian las muestras de tejido prostático sano y las que tienen cáncer de grado 3, 4 y 5. Partiendo de esta base de datos de imágenes se hace uso de una librería llamada *OpenSlide* [18] que proporciona una interfaz sencilla para leer las imágenes *Whole Slide.* Estas se caracterizan por ser de alta resolución y considerablemente grandes, lo cual supone un inconveniente a la hora de cargarlas con librerías estándares, ya que estas están diseñadas para leer imágenes que puedan ser fácilmente descomprimidas en la RAM. Sin embargo, las imágenes *Whole Slide* normalmente ocupan un tamaño del orden de los 10 Gb al descomprimirlas. Además, no existe un formato universal para trabajar con ellas, por lo que cada vendedor implementa su propio formato.

En conclusión, las imágenes disponibles estarán, para este caso concreto, en formato **.tiff* y con un aumento de 1x. Se puede ver un ejemplo de ello en las muestras de la figura que se muestran a continuación, las cuales han sido directamente extraídas de la base datos proporcionada por los especialistas.

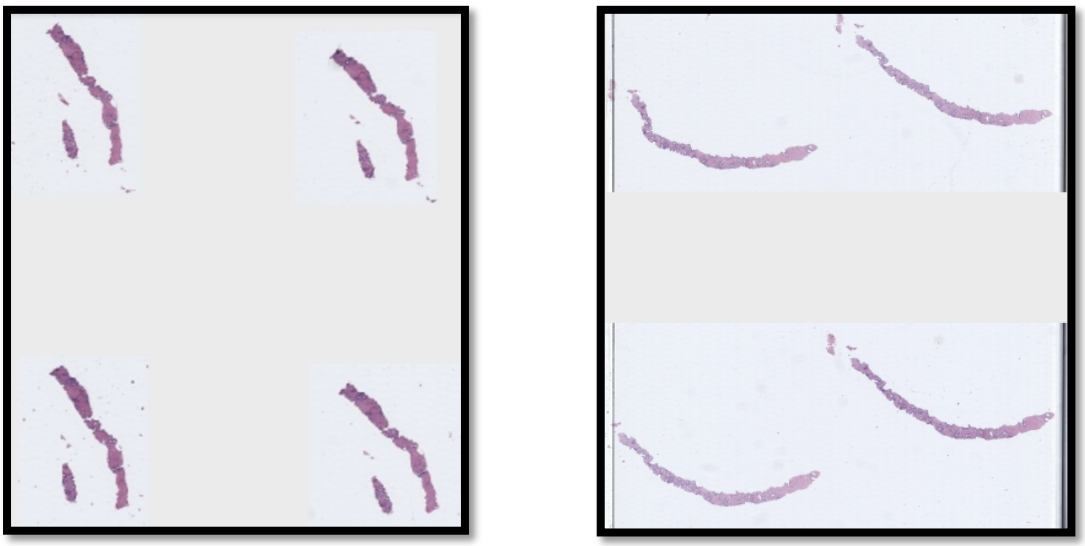


*Figura 2.2: Ejemplos de imágenes histológicas completas de tejido prostático sano en formato *.tiff y con un aumento de 1x.*

Cabe destacar que, en el caso de las muestras con cáncer, cada imagen es de un tamaño diferente, ya que los patólogos seleccionan regiones de cada ejemplar para identificar las zonas de interés, es decir, las zonas que presentan regiones con cáncer de grado 3, 4 y 5.

A partir de estas muestras, haciendo uso de una librería de procesado de imagen llamada *"vips",* se puede pasar de las imágenes completas en formato **.tiff* a imágenes más pequeñas en formato **.dzi*. Es decir, las imágenes completas pueden dividirse ("trocearse") en fragmentos más pequeños sin afectar a la resolución espacial ni a la calidad de la imagen (ver figuras 2.3 y 2.4). De esta forma, se logra disponer de imágenes ampliadas para analizar únicamente las regiones que interesen de cada muestra. Cuanto mayor sea la ampliación de las nuevas imágenes, menor tamaño ocuparán, pues se habrán tomado menos píxeles para formarlas. Por tanto, el software podrá cargar más fácilmente aquellas imágenes que sean más pequeñas, es decir, las de mayor ampliación.

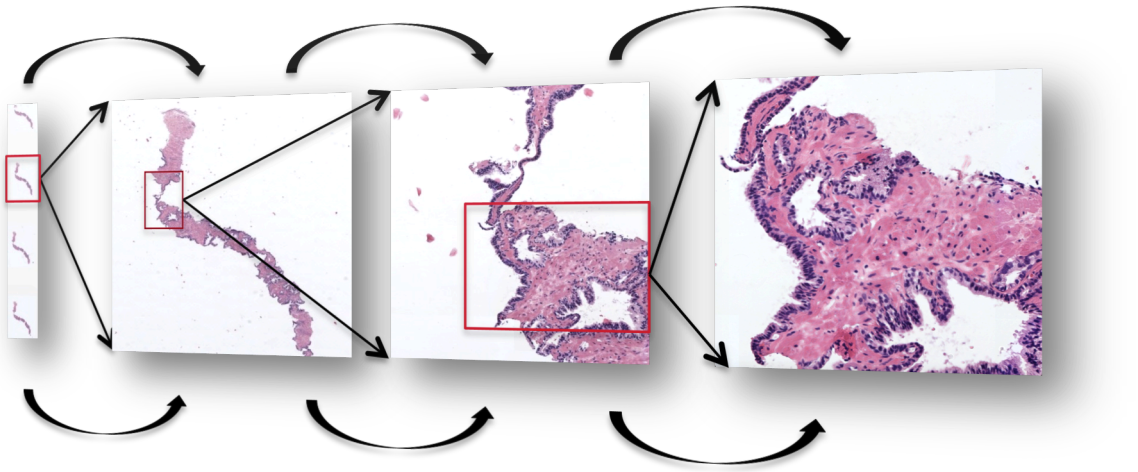


Figura 2.3: Imágenes ampliadas utilizando distintos niveles de zoom óptico.

Para este proyecto de fin de grado se ha decidido trabajar con las imágenes que corresponden al mayor número de aumentos (40x), con la idea de tener en cuenta la componente de más alta resolución. En un principio se pensó que se debería llegar a una solución de compromiso, puesto que por un lado, las características de alta resolución son interesantes para diferenciar bien los distintos componentes de la imagen (principalmente en lo que a núcleos se refiere), pero por otro lado presenta importantes inconvenientes relacionados con el número de glándulas que aparecen en cada imagen para ese tamaño. En conclusión, si se decide trabajar con las imágenes de alta resolución surgiría el problema de encontrar muy pocas glándulas o ninguna, mientras que si se trabaja con imágenes de menos resolución, ese inconveniente quedaría solventado, pero se daría entonces una pérdida del detalle que proporcionan los componentes presentes en la alta frecuencia de la imagen.

Por tanto, teniendo en cuenta estas limitaciones, el primer método llevado a cabo en este proyecto ha consistido en realizar una concatenación de las imágenes de más alta resolución. Con esto se han podido tener en cuenta los detalles correspondientes a las altas frecuencias y, por otra parte, también se ha logrado tener, para cada imagen, un número de glándulas suficientemente grande como para poder evaluar los distintos métodos de clasificación, detección y segmentación.

En resumen, el material inicial del que se parte son las imágenes histopatológicas del tejido prostático no maligno (teñidas con el pigmento de hematoxilina y eosina) utilizando un objetivo de 40x para trabajar con la resolución espacial más alta posible. El método empleado para encarar este apartado se explica con más detalle en el capítulo 2, apartado 2.3.1.

2.2.2 Software utilizado

Para la implementación de los distintos algoritmos utilizados y el cálculo de todos los resultados de este TFG se ha empleado el programa MATLAB® v.R2017a, de The MathWorks, Inc. (Natick, Massachusetts, Estados Unidos). El software MATLAB®, cuyo nombre proviene de *matrix* y *laboratory,* es una plataforma optimizada para resolver problemas de ingeniería y científicos. Consiste en un lenguaje de alto nivel que combina tareas de programación, cálculo y visualización, expresando los resultados en notación matemática. Este software permite el desarrollo de algoritmos, la adquisición de datos y otras tareas relacionadas con el modelado, la simulación y la construcción de interfaces gráficas de usuario. Además, con respecto a las versiones anteriores destacan principalmente los siguientes aspectos:

- *Data analytics* → Consiste en una app utilizada para regresión, clasificación y más algoritmos de *big data* empleando técnicas de *machine learning.*
- *Deep learning* → Permite utilizar modelos pre-entrenados y entrenar sirviéndose de varias GPUs e instancias en la nube.
- Conducción autónoma → Consiste en una nueva *toolbox*[9] que posibilita el diseño, la simulación y el testeo de sistemas de conducción autónoma.
- Simulaciones paralelas → Permite la ejecución de múltiples simulaciones en paralelo.
- Librería 5G → Hace referencia a funciones de MATLAB para simular nuevas tecnologías radio 3GPP 5G. [19]

2.2.3 Hardware utilizado

El proyecto ha sido desarrollado y ejecutado en el ordenador portátil MacBook Air, que cuenta con un procesador Intel Core i5, cuya velocidad es de 1,3 GHz. En cuanto a la memoria RAM, la capacidad es de 4 GB (1600 MHz DDR3), y los gráficos corresponden al modelo Intel HD Graphics 5000 1546 MB.

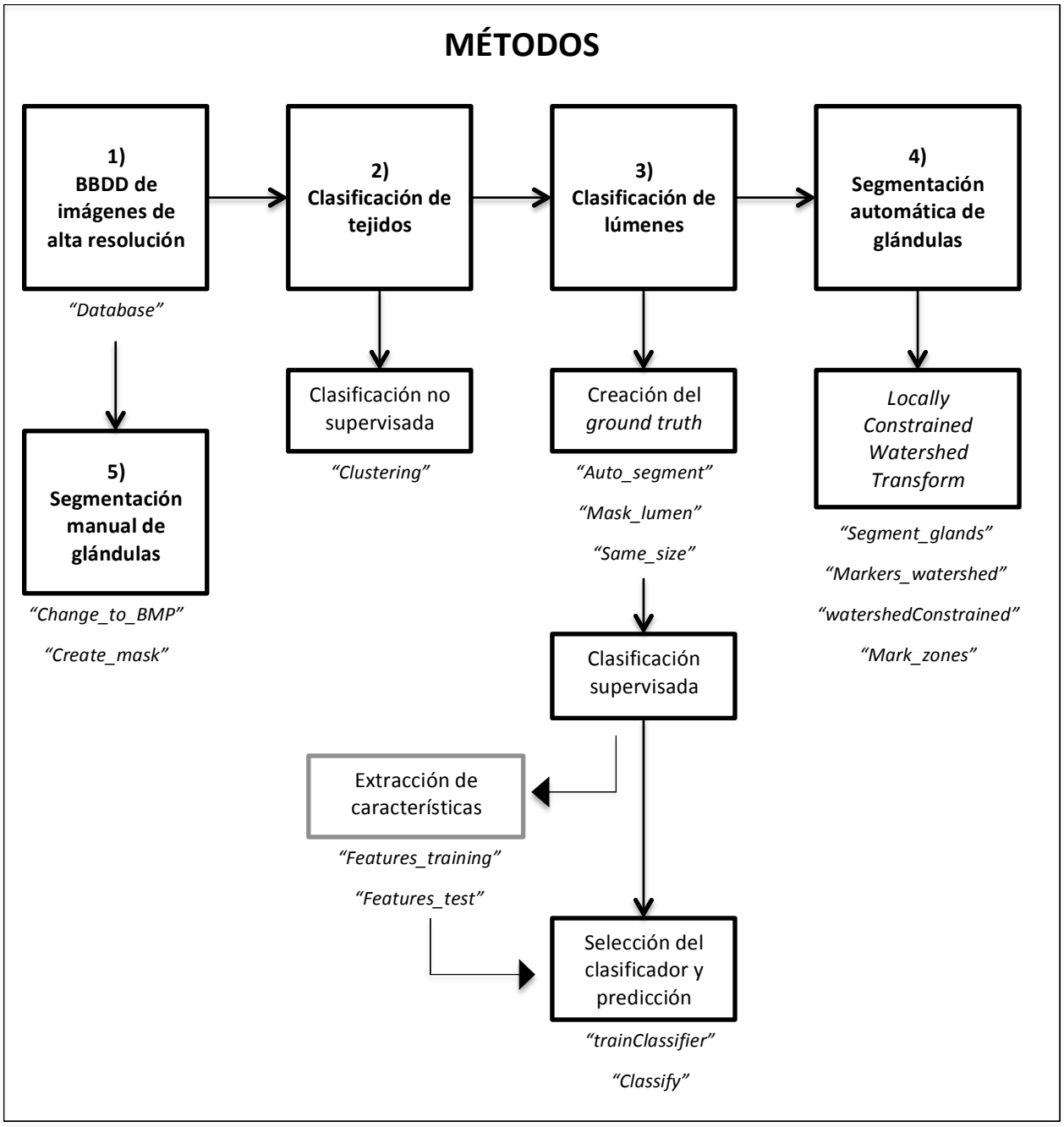
Es importante conocer las características internas del hardware empleado para evaluar el funcionamiento y el rendimiento que se puede alcanzar a la hora de la ejecución de los algoritmos y los tiempos de obtención de resultados. Esto nos permitirá establecer comparaciones futuras y predecir dichos factores cuando se implementen los programas en un *hardware* diferente.

[9] *Toolbox*. Su traducción al castellano es "caja de herramientas". Se define como un entorno de desarrollo que permite introducir la programación en materias sin competencias informáticas.

2.3 Métodos

2.3.1 Diagrama del sistema y lista de funciones

Diagrama de bloques:



Lista de funciones para los métodos:

A continuación se muestran todas las funciones con sus respectivos *inputs* y *outputs* que se han programado a lo largo de este TFG y se han implementado en los diferentes métodos que se exponen en el diagrama anterior. Conforme se avance en la lectura de esta memoria se irán explicando, a medida que se haga necesario, las variables de entrada y de salida de cada función con el fin de entender qué se necesita, qué se hace y qué se consigue para cada uno de los métodos.

1. [img] = **Database**(num_rows, num_cols, read_directory, save_directory)
2. [manual_mask] = **Auto_segment**(read_directory, save_directory)
3. [img, file] = **Same_size**(read_directory, image)
4. [mask_lumen, mask_black] = **Mask_lumen**(img)
5. [cluster_img] = **Clustering**(img, num_clusters)
6. [c_svm, app_svm, mask_svm, mask_app, pos_lum] = **Classify**(folder_img, image)
7. **Features_Training**(docum)
8. [Data, img, mask_black, nuclei_img] = **Features_Test**(folder_img, image)
9. [trainedClassifier, validationAccuracy] = **trainClassifier**(trainingData)
10. **Segment_glands**
11. [img, mask_nuclei, mark_int, mark_ext, file, pos_lum] = **Markers_watershed**(folder_img, image)
12. [imout] = **watershedConstrained**(imin, fg, bg, tamFg, tamBg)
13. [immarc] = **Mark_zones**(img, limits)
14. **Change_to_BMP;**
15. [final_mask] = **Create_mask**(read_directory, save_directory)

Lista de funciones para los resultados:

De la misma forma, se expone la lista de funciones creadas para la obtención de los resultados correspondientes a la clasificación de los lúmenes que indican la presencia de glándulas y a la segmentación de las propias glándulas.

16. [lumen_results, indicators] = **Result_lumen**(read_directory, folder_man_seg)
17. [Coefficients] = **Result_segmentation**(folder_my_result, folder_man_seg)
18. [Dice, Jaccard] = **Evaluate_Coefficients**(auto_mask, man_mask)

2.3.2 Creación de una base de datos de imágenes de alta resolución

El objetivo en este punto es generar una nueva base de datos de imágenes histológicas partiendo de las obtenidas al escanear a la máxima resolución las muestras procedentes de las biopsias. Para ello se buscará la concatenación de las imágenes agrupando tantas como se desee. Esto permitirá eliminar el compromiso explicado en el apartado anterior, y observado en la figura 2.4, ya que se podría trabajar con imágenes de alta resolución sin dejar de tener varias estructuras glandulares que poder detectar y segmentar en una misma imagen.

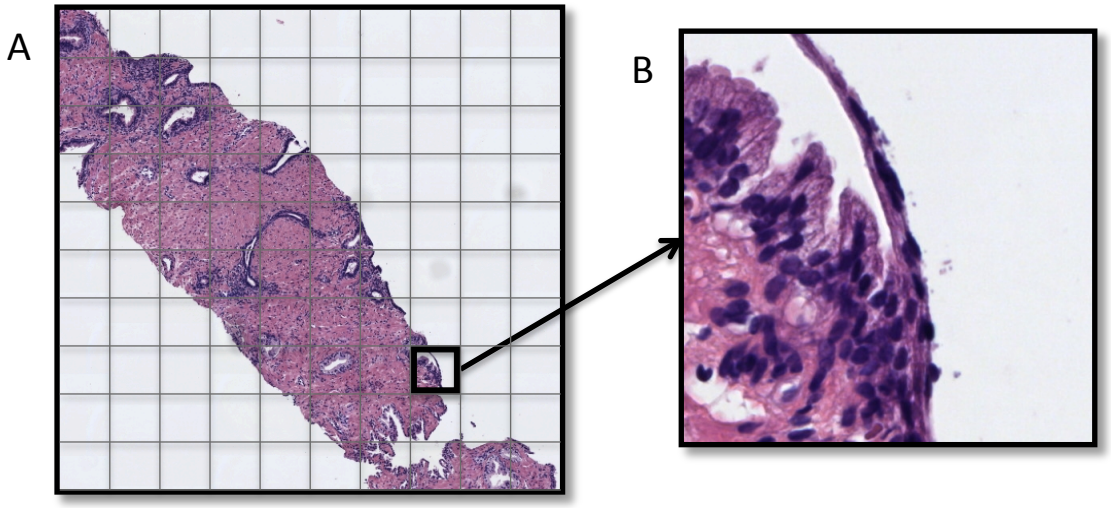


Figura 2.4 : A) Muestra histológica con un zoom óptico aceptable desde el punto de vista de la cantidad de criptas a segmentar, pero con una escasa resolución espacial que dificulta la extracción de características. B) "Trozo" de imagen adquirido de la misma muestra, donde el zoom óptico es demasiado elevado para que exista la posibilidad de segmentar varias glándulas, pero cuya resolución espacial es mucho más elevada, permitiendo así el estudio de otras características.

Como se ha comentado anteriormente, para adoptar una solución al compromiso que se muestra en la figura 2.4 se ha desarrollado una función llamada "*Database*" (ver lista de funciones), que permite la unión de varias imágenes de alta resolución (modelo B de la figura). Con esto será posible evaluar los detalles más pequeños al mismo tiempo que trabajar en la detección y la segmentación de varias glándulas en una misma imagen.

Para el funcionamiento del código es necesario especificar cuatro parámetros que actuarán como *inputs*, a partir de los cuales, el programa devolverá la imagen final de alta resolución, que será el resultado del proceso de concatenación. Para entender bien lo que se pretende con el código, se presenta la figura 2.5 a modo de ejemplo reducido, donde se parte de 4 imágenes de alta resolución (1, 2, 3 y 4) de dimensiones MxN = 512x512 píxeles, y se consigue una imagen final, también de alta resolución, de dimensiones 1024x1024 píxeles.

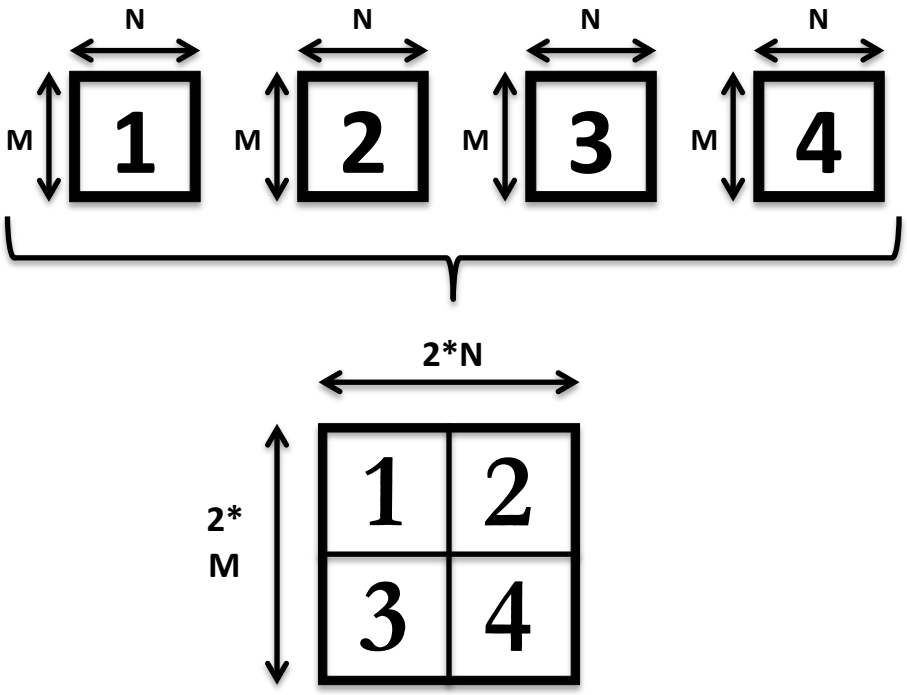


Figura 2.5: Ejemplo aclaratorio del funcionamiento del método de concatenación de imágenes implementado en la función "Database".

Observando el modelo de la figura 2.5 se puede determinar que la nueva imagen está formada por una matriz cuyas entradas son sub-imágenes. Es decir, el resultado se compone de 4 sub-imágenes que se han generado al añadir 2 columnas y 2 filas de imágenes. Como cada sub-imagen ocupa 512x512x3 píxeles, la matriz final en este caso ocupará 1024x1024x3 píxeles.

Los cuatro parámetros de entrada (*inputs)* son: a) el número de imágenes que se quieren concatenar por filas, b) el número de imágenes que se quieren concatenar por columnas, c) el directorio donde están todas las imágenes de alta resolución y d) el directorio donde se desea guardar las nuevas imágenes creadas. De esta forma, finalmente se obtendrá una serie de imágenes, procedentes de una carpeta origen, que habrán sido concatenadas en tantas filas y columnas como se haya especificado y que se habrán guardado en una carpeta destino. A continuación, en la figura 2.6, se visualiza un ejemplo real con diferentes concatenaciones de imágenes para demostrar la escalabilidad del programa diseñado. Además, cabe destacar que, puesto que las imágenes de alta resolución han sido adquiridas "troceando" las imágenes originales, la mayoría son del mismo tamaño: 512x512x3 píxeles, excepto aquellas que limitan con los bordes de la imagen original, que tendrán dimensiones menores. Este detalle está previsto en el desarrollo del código y es importante tenerlo en cuenta porque los parámetros pueden verse alterados al trabajar con imágenes de diferentes tamaños. Con esta información, puede verse que el tamaño de las nuevas imágenes generadas va a depender del número de muestras que se quieran concatenar en filas y columnas. Es decir, si se concatenan 4 imágenes por columna y otras 4 por fila, lo que se obtendría finalmente sería una matriz de 4*4 = 16 sub-imágenes cuyas dimensiones, en lugar de 512x512x3, constarían de 2048x2048x3 píxeles.

Es importante resaltar que, por comodidad, para este trabajo solo se han realizado pruebas generando imágenes cuadradas, es decir, de tamaño MxN, siendo M igual a N. No obstante, el código está diseñado para que las imágenes finales puedan tener dimensiones donde M y N no precisen ser necesariamente iguales.

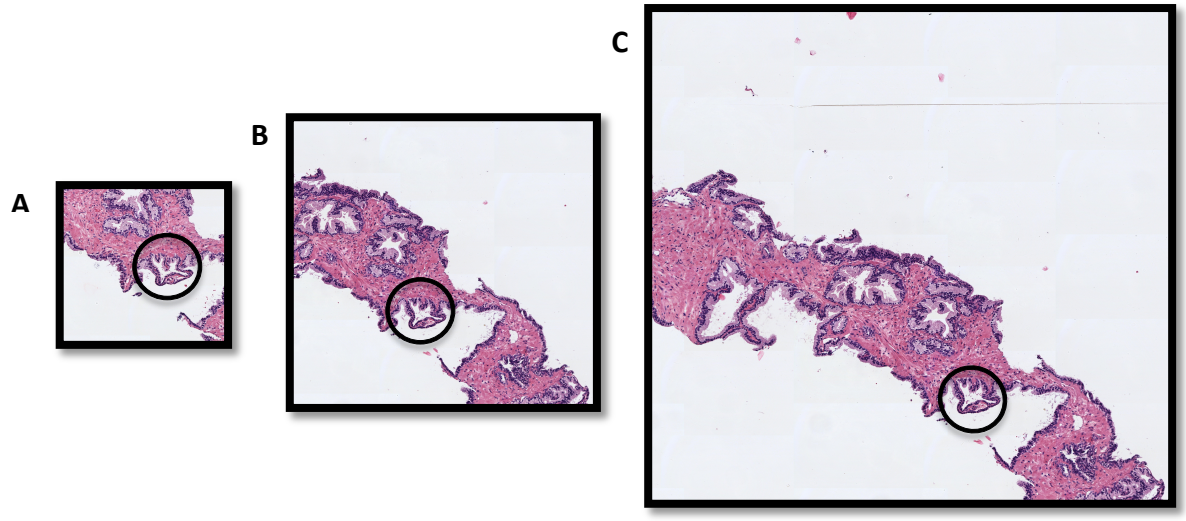


Figura 2.6: A) Muestra histológica compuesta por la concatenación de 5 imágenes por fila y 5 imágenes por columna. Tamaño: 2560x2560x3 píxeles.
B) Muestra histológica compuesta por la concatenación de 10 imágenes por fila y 10 imágenes por columna. Tamaño: 5120x5120x3 píxeles.
C) Muestra histológica compuesta por la concatenación de 15 imágenes por fila y 15 imágenes por columna. Tamaño: 7680x7680x3 píxeles.

Las conclusiones que se pueden extraer al analizar la figura 2.6 son que para la misma resolución (la máxima) se han construido varias imágenes cuya diferencia principal reside en la cantidad de tejido y, por tanto, en el número de glándulas presentes en cada modelo. No obstante, se hace hincapié en que el tamaño de las glándulas es el mismo para los tres casos (analizar el círculo). Por tanto, con este procedimiento surge un nuevo compromiso, pues se tiene en cuenta también el tamaño de la imagen, el cual puede resultar determinante si es demasiado grande. En definitiva, este método permite mejorar la eficiencia en cuanto a resolución espacial y número de criptas a segmentar, pero tiene limitaciones de tamaño cuando se concatenan demasiadas sub-imágenes. Concretamente, en el caso explicado, el modelo de concatenación de 15x15 imágenes no sería idóneo para trabajar porque aunque podría llevarse a cabo la detección y la segmentación de muchas criptas, se estaría hablando de una matriz formada por 225 sub-imágenes de 512x512x3 píxeles cada una, dando como resultado un total de 176.947.200 píxeles, lo cual es excesivamente grande. Esto se traduce en un tiempo computacional demasiado elevado y, por tanto, no sería una práctica eficiente. Emplear el método siguiendo el modelo A, el de la concatenación de 5x5 imágenes, tampoco sería buena idea, pues aunque los tiempos de cómputo disminuirían (por ser el número de píxeles totales: 19.660.800) el número de criptas que se podrían analizar sería demasiado pequeño, lo cual tampoco interesa. Por tanto, los valores idóneos para M y N estarían entre 6 y 10 imágenes, posibilitando llegar así a una buena solución de compromiso.

Por otra parte, es necesario señalar que en muchas ocasiones, sobre todo cuanto más pequeño es el número de M y N, se adquieren más imágenes en blanco o de tejido donde no hay presencia de glándulas. Se observan algunos ejemplos de esto en la figura 2.7, donde los únicos componentes que pueden encontrarse son núcleos y estroma.

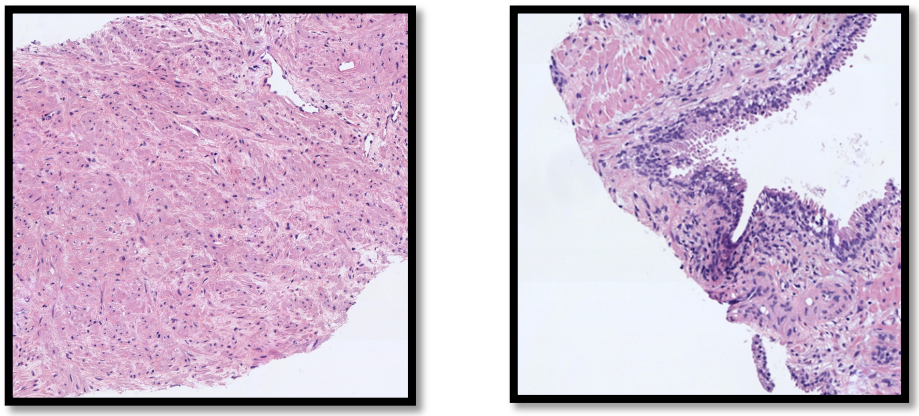


Figura 2.7: Muestras de fragmentos de imágenes de H&E donde no se observa ninguna glándula; únicamente se distinguen zonas rosas correspondientes al estroma y zonas azules que hacen referencia a los núcleos.

Por último y para concluir esta sección, cabe destacar que las imágenes con las que se trabajará de aquí en adelante estarán regidas por los parámetros M=N=8. Es decir, las nuevas imágenes generadas serán el resultado de la concatenación de 64 sub-imágenes, obteniendo finalmente unas dimensiones de 4096x4096x3 píxeles, lo que hace un total de 50.331.648 píxeles. Este valor intermedio permite ajustarse al compromiso existente entre la resolución espacial, el número de glándulas presentes en una misma imagen y el tamaño de dicha imagen que repercute directamente en la facilidad de su lectura en el software y en el tiempo de cómputo que tardan los algoritmos en ejecutarse. Un ejemplo de este tipo de imágenes se refleja en el capítulo 1, en la figura 1.4.

2.3.3 Clasificación de tejidos

Una vez se dispone de las imágenes con las dimensiones deseadas, lo que se pretende a continuación es llevar a cabo una clasificación que permita agrupar, en diferentes clases, los distintos componentes de interés que aparecen en la imagen. Estos componentes de interés son los núcleos, el estroma, el citoplasma y los lúmenes, tal y como se indica en el capítulo 1, en el apartado de imágenes histopatológicas. (Ver figura 1.5).

Para llevar a cabo la agrupación de los datos en diferentes clases es necesario realizar una clasificación no supervisada, ya que solo se dispone de un conjunto de observaciones y no hay información sobre las etiquetas de cada muestra. El objetivo principal es deducir relaciones entre las muestras o las variables que existen detrás de las observaciones. De esta forma, partiendo de un grupo homogéneo de datos, se consigue llegar a clasificar dichos datos en diferentes grupos basándose en la información proporcionada por la tinción de hematoxilina y eosina de las muestras.

2.3.3.1 Clustering

El *clustering* es un tipo de clasificación no supervisada de patrones (observaciones, datos o vectores de características) en grupos (*clusters).* Con este método se puede distinguir entre ítems y *features*, donde cada ítem puede tener varias *features*. Es importante en este punto tener en cuenta el tipo de imagen con la que se va a trabajar, ya que si la imagen se presenta en escala de grises, sólo se observa una característica por cada ítem, el nivel de gris. Sin embargo, partiendo de una imagen de tipo RGB, cada ítem viene representado por tres características, el nivel de rojo (Red), el de verde (Green) y el de azul (Blue).

Por tanto, el *clustering* es una familia de clasificadores no supervisados que realizan la clasificación de los datos en base a la propia distribución de los mismos según la característica o características que se tengan en cuenta. Su objetivo es, a partir de un conjunto de datos, agrupar en diferentes clases aquellos cuyas características sean más parecidas entre sí. Cabe destacar que la medida de similitud entre dos observaciones se especifica en términos de distancia entre vectores de características, siendo la más utilizada la distancia euclídea. Esta funciona muy bien cuando se trata de grupos compactos y aislados. Matemáticamente, siendo *x* y *c* dos vectores de características (ítems, en este caso píxeles, distintos) se expresa la distancia euclídea como:

$$d(x,c) = (x-c)(x-c)^T \qquad (2.1)$$

También se pueden utilizar otras distancias como las que se definen en la siguiente tabla:

Cityblock o Manhattan o norma l_1	$d(x,c) = \sum_{j=1}^{P} \lvert x_j - c_j \rvert$
Coseno	$d(x,c) = 1 - \frac{xc^T}{\sqrt{(xx^T)}\sqrt{(cc^T)}}$
Correlación	$d(x,c) = 1 - \frac{(x-\overline{x})(c-\overline{c})}{\sqrt{(x-\overline{x})(x-\overline{x})^T}\sqrt{(c-\overline{c})(c-\overline{c})^T}}$
Mahalanobis	$d(x,c) = (x-c)\sum^{-1}(x-c)^T$

Tabla 3: Medidas de similitud en términos de distancias [20].

Dentro de las técnicas de *clustering* se pueden distinguir los métodos jerárquicos y los no jerárquicos. Concretamente en este TFG, para realizar el *clustering*, sólo se ha utilizado la técnica conocida como *kmeans,* que pertenece a la familia de los métodos no jerárquicos de la clasificación no supervisada. Otras técnicas conocidas correspondientes a este tipo de métodos son el *fuzzy c-means* como una versión difusa del *kmeans* que permite asignar a cada observación un grado difuso de pertenencia a cada grupo*,* el *kmedoids* y el *mean-shift,* que se proponen como líneas futuras para intentar optimizar el método y, por tanto, el resultado final.

La técnica del *kmeans* consiste en un método de agrupamiento que permite clasificar cada observación en el *cluster* que se encuentra más próximo en términos del centroide (media). Su objetivo, por tanto, es agrupar un conjunto de *n* observaciones en *k* grupos, de tal forma que cada observación pertenecerá al grupo cuyo valor medio sea más cercano. En términos formales, la técnica del *kmeans* puede ser definida de la siguiente forma: dado un conjunto de observaciones $(x_1, x_2, \ldots, x_n)$ donde cada observación es un vector real de *d* dimensiones, *kmeans* crea una partición de las observaciones en *k* conjuntos, donde $k \leq n$, con el objetivo de minimizar la suma de los cuadrados dentro de cada grupo (WCSS): $S = \{S_1, S_2, \ldots, S_k\}$ [21].

Matemáticamente se expresa como:

$$\arg min \sum_{i=1}^{k} \sum_{x_j \in S_i} ||x_j - \mu_i||^2 \qquad (2.2)$$

El algoritmo de la técnica *kmeans* se basa en un procedimiento iterativo en el que se diferencian dos pasos principalmente:

1. Paso de asignación. En primer lugar, se define el número de *clusters* iniciales *(k)* y se calculan sus centroides. A continuación, para el conjunto de observaciones disponibles se calcula la distancia (euclídea normalmente) de cada elemento a cada centroide de los *k clusters* y se asigna la etiqueta del *cluster* cuyo centroide es más próximo.
2. Paso de actualización. Se calculan los nuevos centroides de los *k clusters* como los centroides de las observaciones en el grupo y se repite el procedimiento hasta que el algoritmo converja. Esto ocurre cuando ya no se produce ninguna reasignación porque los elementos se han estabilizado en alguno de los grupos.

Siguiendo los pasos 1 y 2 se muestra el siguiente ejemplo para la comprensión del algoritmo:

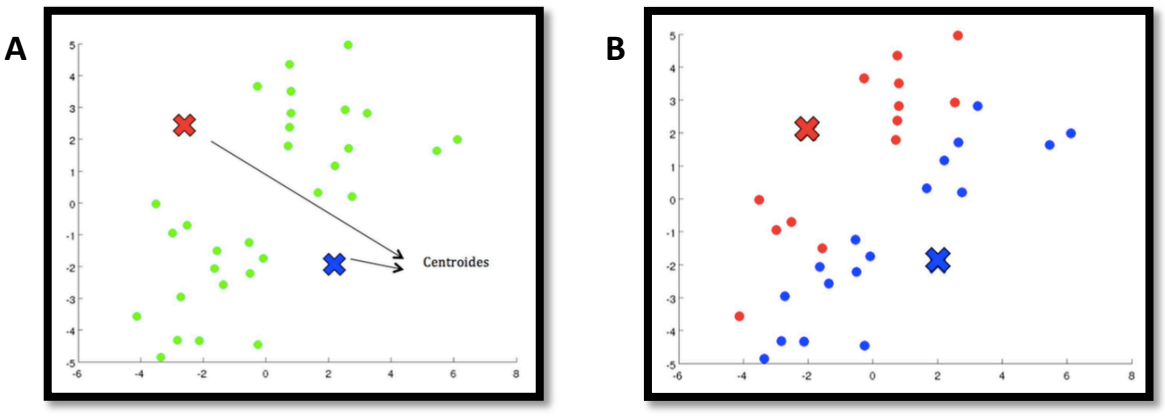


Figura 2.8: A) Paso de asignación en el que se define el número de clusters iniciales, a partir de un grupo homogéneo de datos, y se calculan los centroides. B) Imagen que muestra cómo a cada observación se le asigna una etiqueta en función del centroide del cluster cuya distancia euclídea es más próxima.

A continuación, en el paso 2, se recalculan los centroides como la media de las observaciones que hay en cada grupo y de nuevo se asigna a cada observación la etiqueta del *cluster* cuyo centroide es más cercano, de forma que se repetiría el proceso hasta lograr la convergencia.

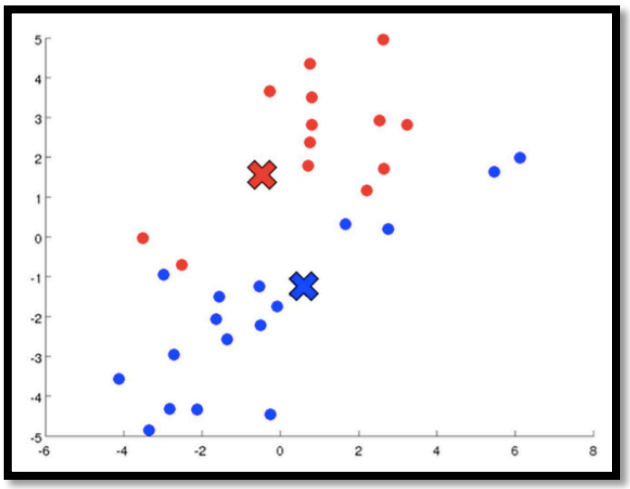


Figura 2.9: Paso 2 del algoritmo kmeans en el que se recalculan los centroides y se repite el procedimiento anterior hasta que converge el algoritmo, es decir, hasta que las etiquetas de cada observación no varían.

Por último, se puede determinar en base a lo expuesto que la técnica del *kmeans* presenta ventajas como la simplicidad y la rapidez, pero también inconvenientes como la sensibilidad a los *outliers* y la necesidad de definir inicialmente el número de *clusters* [20]. Se dice que el método no es robusto a *outliers* porque tiene en cuenta todas las observaciones, incluso aquellas que están muy alejadas de los centroides. De esta forma, a la hora de calcular los centroides como la media de las observaciones de cada grupo, se consideran también los valores atípicos, desvirtuando con ello la exactitud del método.

En cuanto al inconveniente de la necesidad de definir el número de *clusters*, en lo que a este TFG respecta, podría considerarse incluso como una ventaja, ya que se conoce el pigmento de hematoxilina y eosina en las imágenes histológicas de próstata y se sabe que aparecen cuatro colores diferentes (rosa, azul, púrpura y blanco) correspondientes a los cuatro elementos que interesa analizar. De esta forma, se ha elaborado una función llamada "*Clustering"* (ver lista de funciones) en la que se ha implementado el método de clasificación no supervisada: *kmeans.* Este paso es fundamental, pues en función de cómo se clasifiquen los distintos objetos de la imagen se podrán obtener las máscaras binarias[10] en las que visualizar los distintos componentes por separado.

La función "*Clustering*" requiere como *inputs*: (i) *img*, que corresponde a la imagen con la que se va a trabajar y (ii) *num_clusters,* que se refiere al número de clases en las que se agruparán los datos. Este parámetro se definirá como *k=4* para obtener las máscaras de los cuatro componentes de interés. Como *output*, el programa devolverá, tras su ejecución, una imagen llamada "*cluster_img*" de dimensiones MxN donde todos los píxeles tendrán valores asignados (1, 2, 3 ó 4) correspondientes a las etiquetas de cada componente de la imagen. (Ver figuras 2.11 y 2.12). En el funcionamiento interno del código, en primer lugar, se reorganizan los datos presentes en la imagen, ya que como se menciona en el apartado 2.3.1, se está trabajando con imágenes de 4096x4096x3

[10] Máscara binaria es una imagen en la que los píxeles solo tienen valores de 0s y 1s, de tal forma que aquellos objetos cuyo nivel de intensidad sea 1 aparecerán de color blanco en la imagen, y aquellos con un valor igual a 0 aparecerán en negro.

píxeles, lo cual es un tamaño considerablemente grande. Para ello, primero se utiliza un re-escalado implementando un función llamada "*Same_Size*" (ver lista de funciones) cuyos *inputs* son (i) *read_directory,* correspondiente al directorio donde se encuentra la imagen que se quiere re-escalar y (ii) *image,* que es la imagen en sí a modificar. De esta forma se consigue finalmente, como *output "img", que es* una imagen de dimensiones 1024x1024x3 píxeles, que mejora el coste computacional, aunque pierde necesariamente en resolución espacial. A continuación, se genera una nueva matriz donde el número de filas es 1.048.576 (resultado de multiplicar 1024 por 1024) y el número de columnas, 3. Es decir, se crea una matriz bidimensional para poder utilizar el método de clasificación *kmeans.* Resaltar que antes de eso se hace un cribado de datos en el que solo se toma uno de cada cincuenta píxeles. Con esto se consigue disponer de una cantidad de información lo suficientemente grande como para que sea representativa del modelo y, también, mejorar la eficiencia computacional. Posteriormente se obtienen los centroides, se ordenan y se calcula la distancia euclídea de cada observación al centroide de cada *cluster,* de modo que a cada observación se le asigna la etiqueta del *cluster* más cercano en términos de centroide. Por último, se vuelven a reorganizar los datos para tener una matriz tridimensional como la original, pero donde los píxeles en este caso tienen valores de 1, 2, 3 ó 4 según el *cluster* en el que hayan sido clasificados.

En la figura 2.10 se puede ver, por colores, cómo se distribuyen los datos para la imagen A y cómo han sido clasificados en 4 *clusters* teniendo en cuenta 1 de cada 50 píxeles.

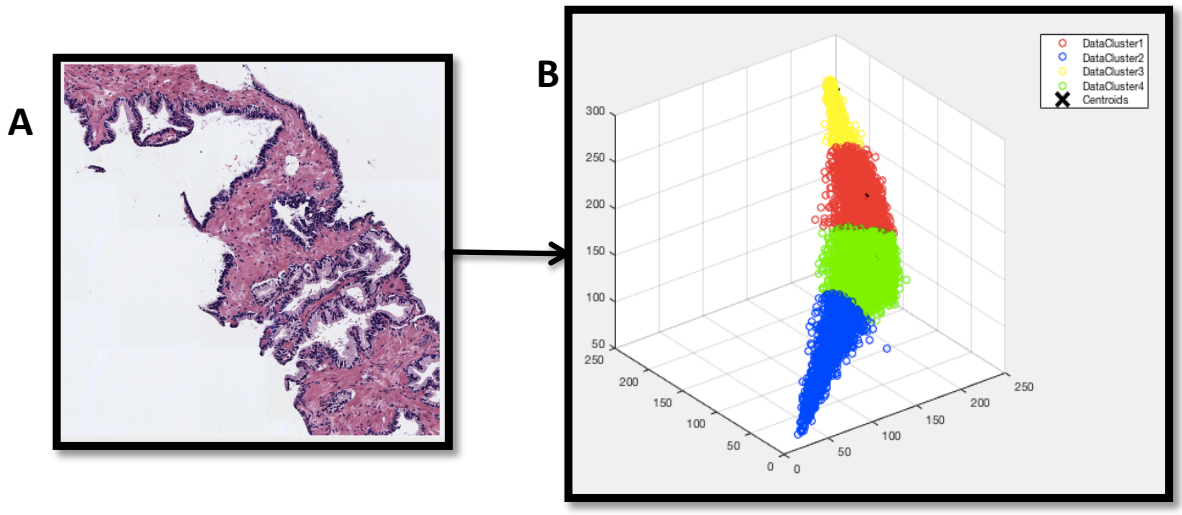
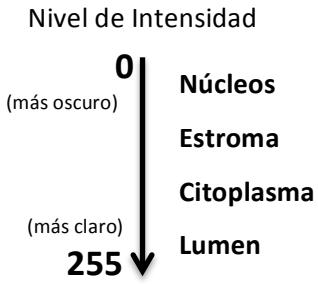


Figura 2.10: A) Imagen a partir de la cual se obtiene la información de los píxeles. B) Distribución de los datos en los diferentes clusters cogiendo 1 de cada 50 píxeles. No se alcanza a visualizar la localización del centroide debido a la gran densidad de datos que se representan.

Como se puede observar en la figura 2.10.B, hay cuatro pseudocolores diferentes correspondientes a los cuatro componentes que se diferencian en una imagen de hematoxilina y eosina, tal y como se explica en el capítulo 1, figura 1.5 (mismos pseudocolores). Partiendo de la imagen B de la figura, y puesto que se han asignado las etiquetas 1, 2, 3 y 4 a los componentes de interés, se puede generar una imagen de grises donde se visualice cada etiqueta con un tono de gris diferente, quedando establecidas de esa forma las siguientes equivalencias:

- *Cluster* azul → Se corresponde con los núcleos.
- *Cluster* verde → Se corresponde con el estroma.
- *Cluster* rojo → Se corresponde con el citoplasma.
- *Cluster* amarillo → Se corresponde con el lumen.



En la figura 2.11, se muestra la imagen "*cluster_img*" que contiene los 4 componentes marcados con distintos tonos de gris, según la etiqueta del *cluster* al que pertenecen. De esta forma se puede observar claramente que a los núcleos se les asigna la etiqueta 1, al estroma la 2, al citoplasma la 3 y, por último, al lumen la 4.

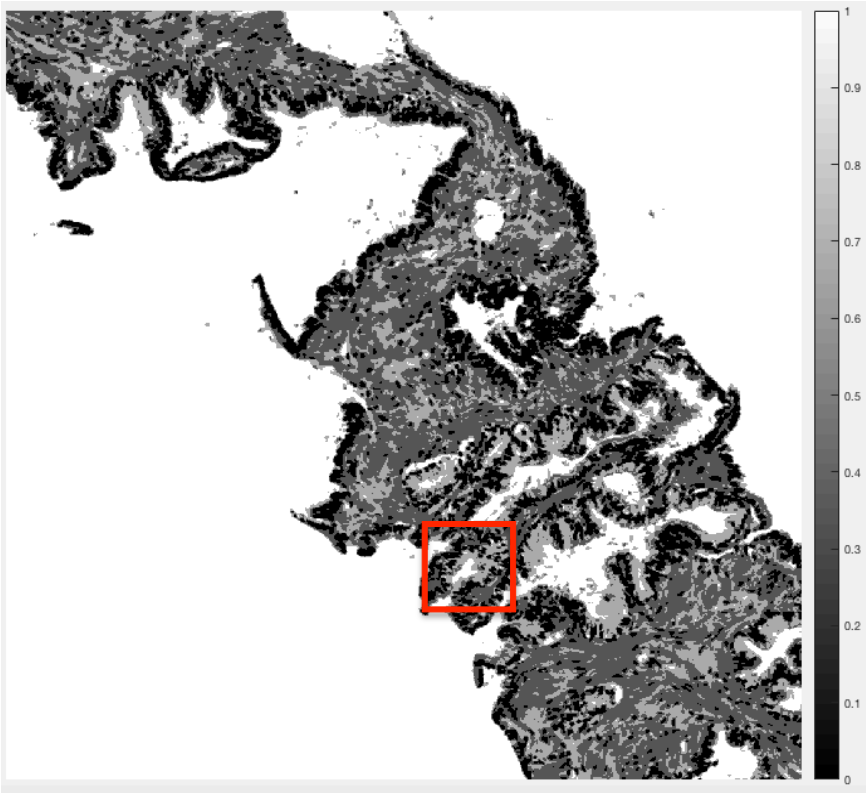


Figura 2.11: Imagen generada como output de la función Clustering, donde se observan los cuatro componentes en distintos niveles de gris, siendo los más oscuros los núcleos, seguidos del estroma, del citoplasma y en último lugar del lumen, que se observa en blanco coincidiendo con el fondo de la imagen.

Una vez se dispone de la imagen de la figura 2.11, se pueden obtener las diferentes máscaras binarias para conseguir analizar cada uno de los componentes por separado. (Ver figura 2.12). Haciendo zoom en la zona marcada en rojo de la figura, se puede observar el procedimiento llevado a cabo hasta la obtención de dichas máscaras.

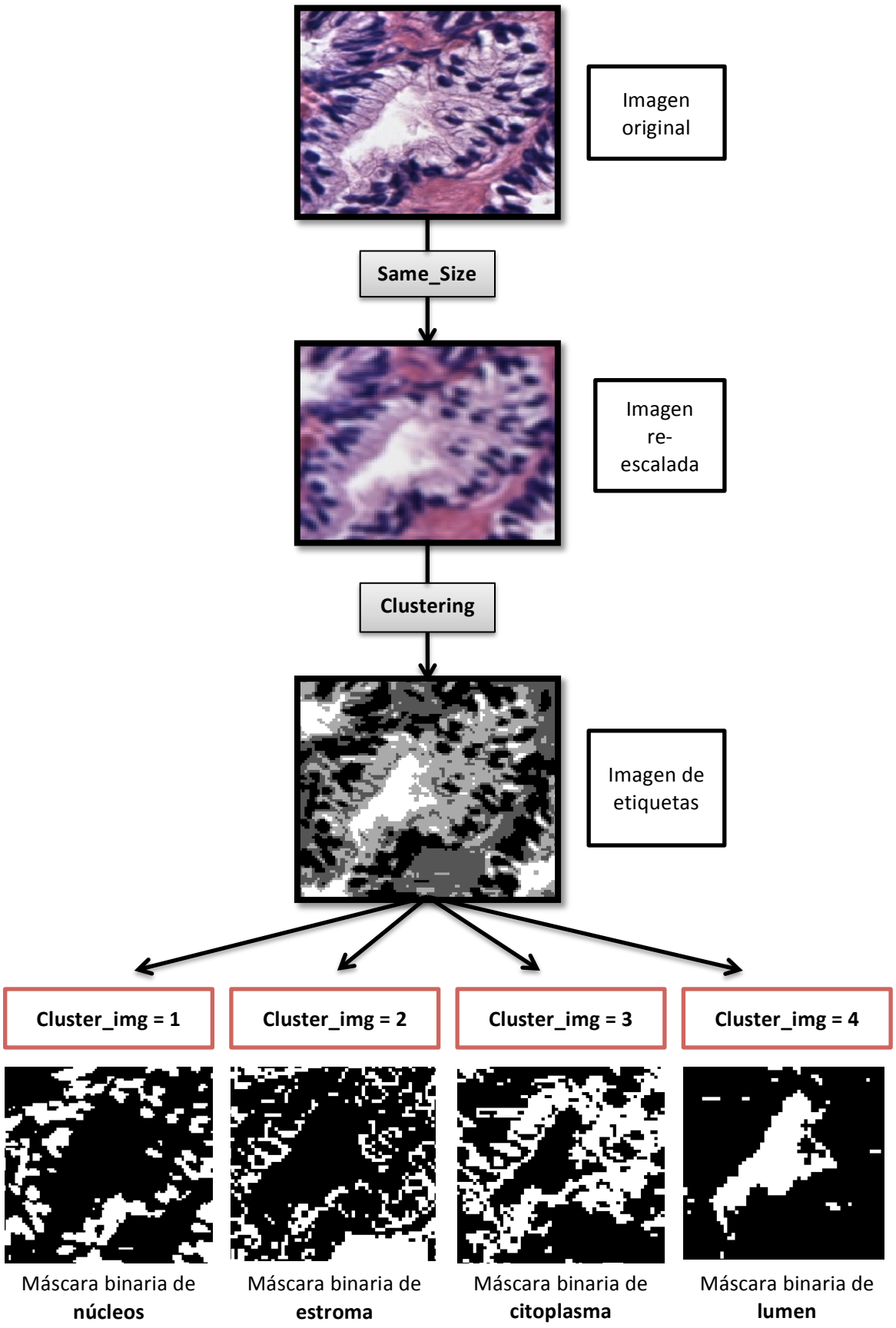


Figura 2.12: Procedimiento implementado para la obtención de las máscaras binarias con el método de clasificación no supervisada basado en la técnica "kmeans".

2.3.4 Clasificación de lúmenes

Como ya se ha comentado en el capítulo 1, el objetivo final de este TFG es la segmentación de glándulas en imágenes histológicas de próstata. Para ello, el primer paso imprescindible consiste en detectar dónde hay glándulas a lo largo del tejido para después poder aplicar las técnicas de segmentación. Por tanto, para el caso que se trata en este proyecto, donde las imágenes histológicas corresponden a muestras de biopsias de próstata que no presentan cáncer o que presentan un grado de cáncer muy bajo (Gleason<3), se ha decidido encarar el problema de la detección desde el punto de vista de los lúmenes. Esto es, cuando las muestras corresponden a tejido sano, todas las glándulas suelen presentar un lumen con citoplasma alrededor y núcleos bordeando al citoplasma. (Ver figura 1.5). Conociendo esta información, se ha decidido estudiar las imágenes en busca de lúmenes, que son los que indican la presencia de glándulas. Además se sabe que, para este caso, todas las glándulas que haya en el tejido van a contener un elemento que se reconocerá como lumen. El inconveniente en este punto es que con la máscara binaria de lúmenes, que se obtiene tras el proceso del *clustering*, la mayoría de los objetos detectados como lumen no se corresponden con los lúmenes que formarían parte de una glándula. Es decir, la máscara binaria de lúmenes contiene todos aquellos lúmenes que efectivamente pertenecen a una glándula, pero también aquellos objetos que han sido clasificados como lumen cuando realmente no lo son (es decir, falsos positivos). Esto ocurre porque en la imagen "*cluster_img*" (figura 2.11) hay objetos, como el fondo (sustrato sin tejido) cuyos píxeles tienen un nivel de intensidad igual o muy parecido al que tienen los lúmenes y por eso son clasificados con la misma etiqueta. Para entender mejor el problema, se muestran las figuras 2.13 y 2.14.

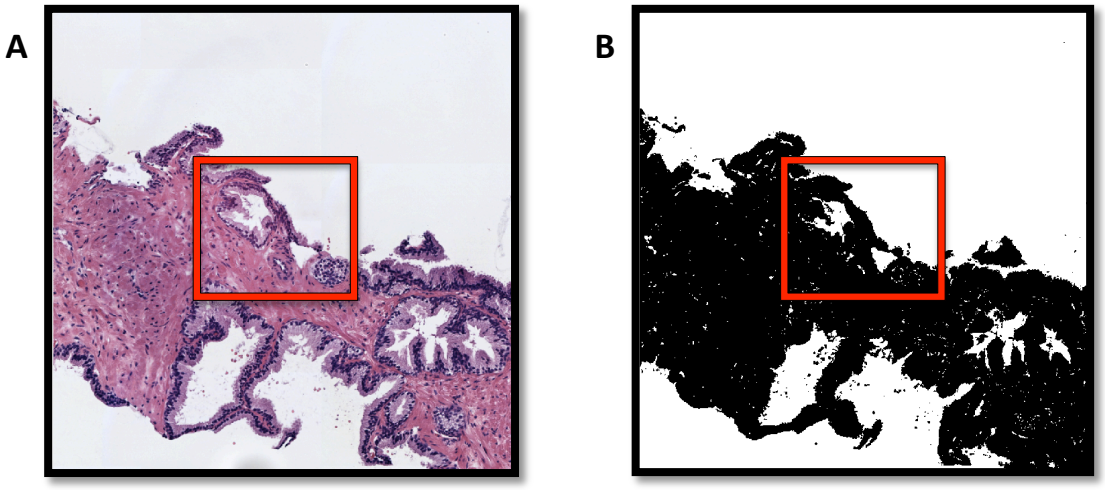


Figura 2.13: A) Imagen original de tejido prostático sano. B) Máscara binaria de lúmenes adquirida con la función "Clustering".

En el figura 2.13.A, lo que interesaría tener sería una imagen donde todo aquello que no fuera lumen se presentara de color negro, y solo se vieran en blanco aquellos elementos que realmente son lúmenes. Como se puede observar, hay una gran cantidad de componentes en la imagen que presentan el mismo nivel de intensidad que el lumen y que, obviamente, no corresponden a tal objeto. Un ejemplo de esto es el fondo de la imagen sin ir más lejos. No obstante, además del fondo, aparecen también otros elementos que son como pequeños puntitos blancos correspondientes a la

luz que presenta el tejido a lo largo del estroma. Al ser de color blanco en la imagen original de hematoxilina y eosina, el método del *clustering* clasifica esos pequeños componentes como lúmenes, lo cual es incorrecto. Por tanto, hasta ahora en el problema de clasificación no supervisada se tienen dos inconvenientes: (i) objetos demasiado grandes que hacen referencia al fondo de la imagen y (ii) objetos muy pequeños que se corresponden con parte del estroma. Y además, aparece un tercer inconveniente relacionado con aquellos objetos que tienen un tamaño muy similar a los elementos que realmente son lúmenes, pero que no lo son porque, por sus características, se consideran roturas de tejido. Haciendo zoom en las zonas marcadas en rojo de la figura 2.13, se pueden distinguir los cuatro tipos de objetos que se encuentran en las máscaras de lúmenes.

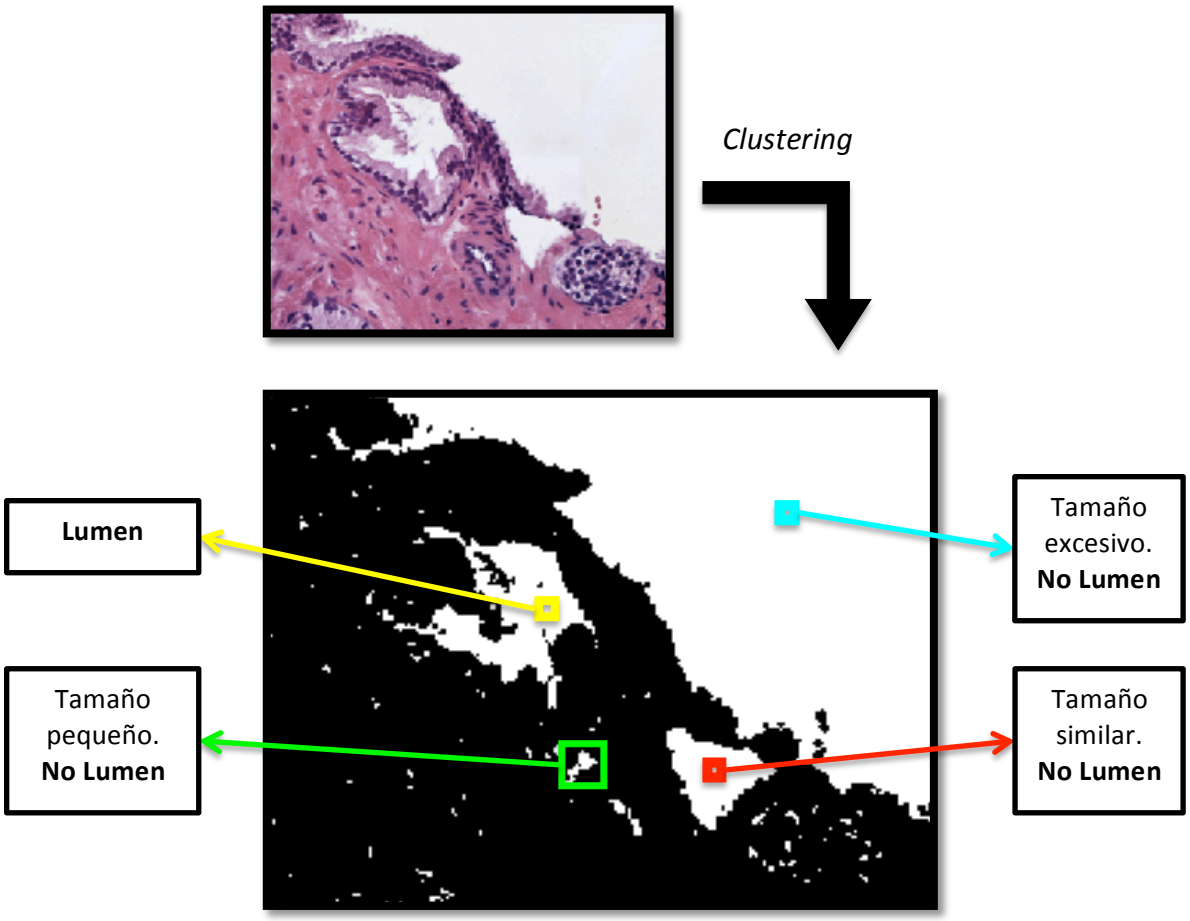


Figura 2.14: Ejemplo aclaratorio de los tipos de elementos identificados como lumen tras el proceso del clustering donde solo uno de esos cuatro tipos se corresponde con un lumen auténtico.

Tras exponer el problema de la máscara binaria extraída directamente del proceso de clasificación no supervisada, una de las conclusiones a las que se puede llegar es que las imágenes resultantes presentan lúmenes para todas las glándulas que aparecen en el tejido, pero no todos los objetos identificados como lumen se corresponden necesariamente con una glándula, por lo que esos objetos no son lúmenes, sino falsos positivos. Por tanto, se hace indispensable un sistema manual que permita clasificar correctamente aquellos objetos que sí son lumen, a partir de la máscara obtenida del *clustering*, creando de esta forma el *ground truth.*

2.3.4.1 Creación del *ground truth* para la clasificación

En primer lugar, se diseña la función "*Mask_lumen"* (ver lista de funciones) que tiene como *input* la variable "*image"* correspondiente a la máscara binaria extraída directamente del *clustering* y como *outputs* "*mask_lumen" y "mask_black",* que son dos máscaras de lúmenes procesadas como las que se observan en la figura 2.15.B y 2.15.C. De esta forma se hace un procesado de la imagen de entrada eliminando principalmente dos de los cuatro tipos de elementos que se encontraban en la figura 2.14, concretamente los representados por los colores cian y verde. Es decir, el fondo de la imagen y los elementos más pequeños que corresponden a partes del estroma, respectivamente. Esto es posible gracias a la definición de dos umbrales en los que se especifica un tamaño de píxeles límite. Para eliminar los objetos más pequeños se utiliza el comando *bwareaopen* que permite fijar un umbral mínimo y eliminar todos aquellos objetos cuyo número de píxeles conectados sea inferior a dicho umbral. Por otra parte, para eliminar los objetos pertenecientes al fondo de la imagen se desarrolla un pequeño código donde se fija, en este caso, un umbral máximo. De esta forma, todos los objetos que tengan un número de píxeles conectados por encima de dicho umbral se etiquetarán con un cero para que ya no sean identificados como lumen. Los valores de dichos umbrales han sido fijados como 150 y 30000 píxeles respectivamente, para que todos los objetos más pequeños y más grandes sean eliminados siendo máxima la sensibilidad en un subconjunto de imágenes de la base de datos (el conjunto de entrenamiento). Lo que se pretende por tanto con esta función es generar una nueva máscara que permita una detección de los lúmenes más precisa. Es necesario hacer hincapié en la importancia de la fijación de los umbrales para que no se elimine ningún objeto que sea lumen auténtico. De esta forma, la nueva imagen tendrá una sensibilidad de 1, pues todas las glándulas tendrán su lumen correspondiente en la máscara, y además se gana en especificidad, al haber menos falsos positivos.

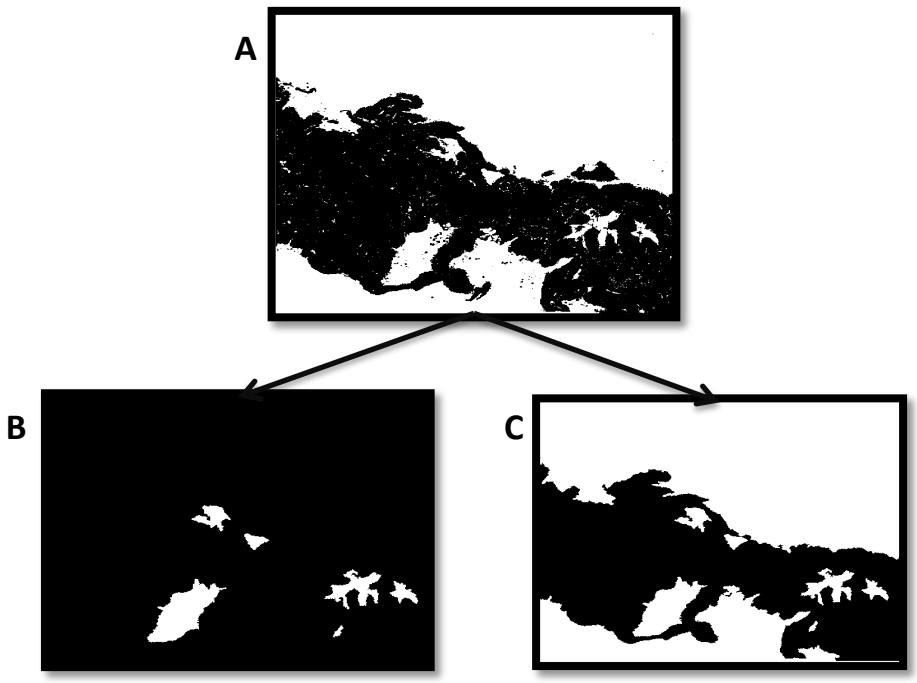


Figura 2.15: A) Máscara binaria de lúmenes extraída directamente del clustering. B) Máscara de lúmenes procesada donde solo están en blanco los objetos que tienen un número de píxeles conectados dentro de los límites fijados por ambos umbrales. C) Máscara igual que la B pero poniendo en blanco también los objetos correspondientes al fondo de la imagen, es decir, teniendo en cuenta solo el umbral inferior, no el superior.

La máscara realmente importante es la que se observa en la figura 2.15.B, no obstante se genera también la *C* para que sirva de guía en el proceso de la segmentación semiautomática, la cual permitirá seleccionar los objetos que realmente son lumen, creando así el *ground truth*. Para llevar a cabo dicho proceso se ha implementado la función "*Auto_segment"* (ver lista de funciones) que hace uso del comando *bwselect* para crear una imagen binaria donde exclusivamente los elementos que se cliquen en la imagen de entrada aparecerán en blanco. De esta forma, se puede obtener una máscara binaria de lúmenes en la que únicamente se visualicen en blanco los elementos que se corresponden con lúmenes auténticos.

El *ground truth* realizado mediante el procedimiento anterior será utilizado para:

1) Que sirva como variable de respuesta a la hora de entrenar los clasificadores. Es decir, que sea una entrada de los clasificadores supervisados para determinar qué objetos son lumen y qué objetos no, y en función de ello, realizar el entrenamiento de las muestras de *training*.
2) Evaluar los resultados de la detección de lúmenes mediante parámetros como la sensibilidad, la especificidad, el valor predictivo positivo, el valor predictivo negativo, la precisión y el F1Score en las muestras de *test.* (Medidas definidas en el capítulo 3).

La función "*Auto_segment*" consta de dos *inputs* que hacen referencia a: (i) *read_directory,* que es el directorio donde se encuentran las imágenes con las que se va a trabajar (de *training o de test)* y (ii) *save_directory,* que corresponde al directorio donde se desea guardar las nuevas imágenes generadas (máscaras binarias que conforman el *ground truth* de lúmenes). Al ejecutar la función, el programa muestra por pantalla dos imágenes: por un lado, la imagen original en formato RGB y por otro lado, una máscara binaria procesada como la que se observa en la figura 2.15.C. De esta forma, observando en un lado la imagen original (donde perfectamente se distingue el lumen de una glándula) se puede ir clicando en la otra imagen los objetos que se correspondan con esos lúmenes para finalmente guardar la imagen en el directorio que se haya especificado. Se muestra un ejemplo del funcionamiento del *script* en la siguiente figura:

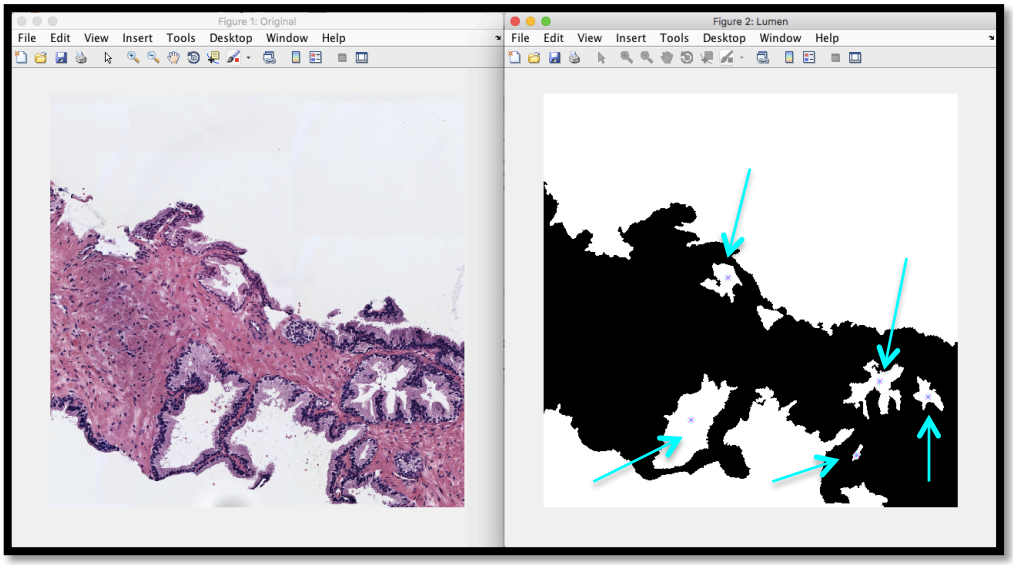


Figura 2.16: En la parte izquierda se observa la imagen original que se utiliza como guía para saber dónde se localizan los lúmenes de las glándulas y poder clicarlos en la imagen de la derecha. Se alcanza a ver cómo se ha marcado con una estrellita azul aquellos elementos que se consideran lumen.

Tras clicar todos los objetos considerados como lúmenes de glándulas, se pulsa la tecla "*Enter*" para generar las nuevas máscaras. En este punto, se ha diseñado la función de tal forma que se pueda visualizar en rojo el contorno de los objetos clicados y se pregunte por pantalla en una ventana si es correcta la segmentación. En caso de serlo, se debe pulsar "1" para que esa máscara se guarde y se muestre por pantalla la siguiente imagen del directorio. Este proceso se repite hasta que se obtenga correctamente la máscara de lúmenes de todas las imágenes del directorio que se está analizando. En caso de no ser correcta la segmentación, se debe pulsar "0" para que se vuelva a mostrar la misma imagen por pantalla y se repita el procedimiento. Ambos casos se observan en la siguiente figura:

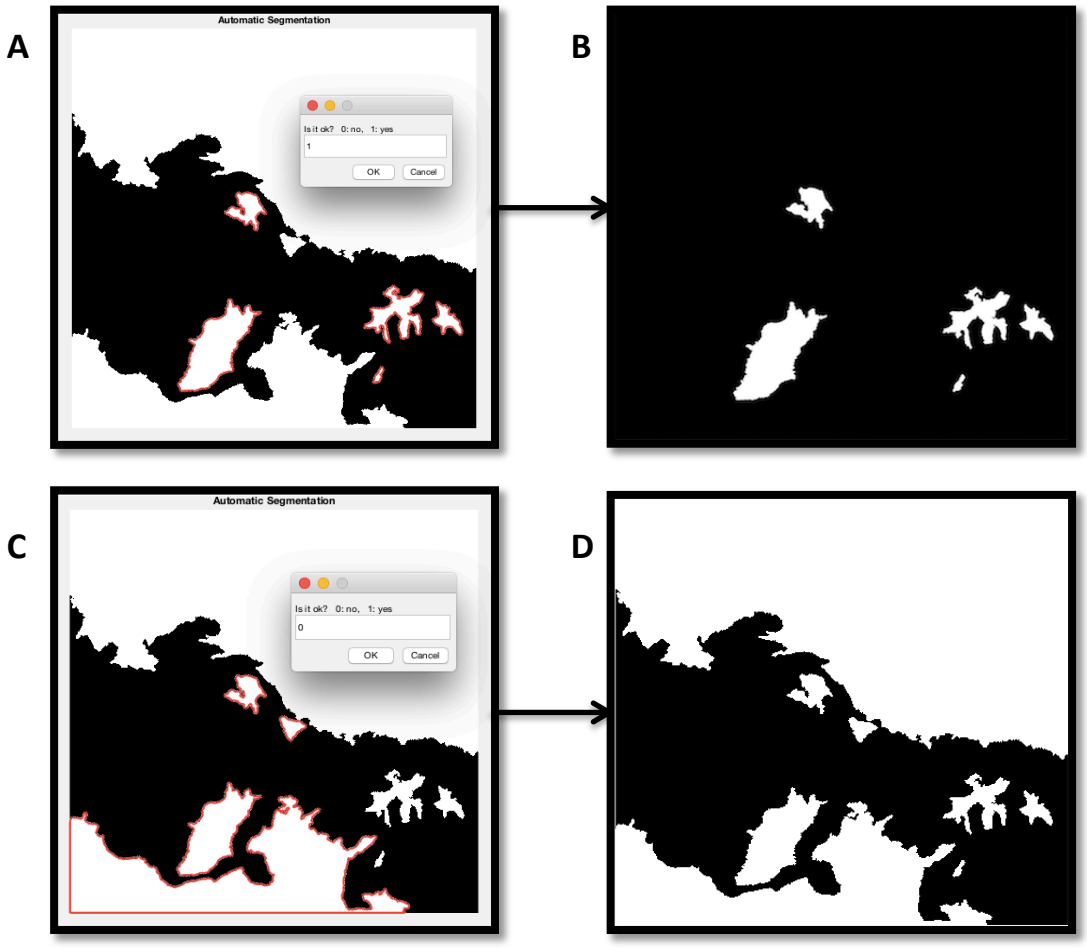


Figura 2.17: A) La segmentación semiautomática se ha realizado correctamente, por lo que se pulsa "1" y se guarda en el directorio especificado la imagen correspondiente al modelo B. B) Máscara binaria en la que los objetos en blanco han sido detectados manualmente. C) Modelo en el que la segmentación ha sido errónea, por lo que se pula el "0" para que no guarde la segmentación realizada y se muestra de nuevo la misma imagen (D) por pantalla para llevar a cabo la detección correcta.

Tras la ejecución completa del código, el objetivo es obtener el *output* "*manual_mask*" (máscara binaria como la de la figura 2.17.B) para todas las imágenes tanto de *training* como de *test*. De esta forma se puede entrenar a los clasificadores y también evaluar los resultados de la detección de lúmenes.

2.3.4.2 Clasificación supervisada

El objetivo de la clasificación consiste en establecer relaciones entre los elementos que se analizan y las categorías o las clases que se especifican. Además, la clasificación es una técnica muy útil, pues se puede aplicar en una gran cantidad de campos de estudio como el reconocimiento de patrones [22].

Concretamente, la clasificación supervisada es aquella que cuenta con un conocimiento a priori, es decir, consiste en un tipo de clasificación que se lleva a cabo basándose en otros ítems que ya han sido clasificados anteriormente. De esta forma, un clasificador supervisado puede valerse de la clasificación realizada en otros objetos para agrupar los nuevos elementos en las clases que se haya especificado. Por tanto, se define la clasificación supervisada como aquella basada en pares "dato-solución" que permite establecer relaciones entre los datos (características extraídas) y su clase correspondiente. Para esto se necesita tener datos etiquetados [23].

En este TFG, se ha dividido el total de imágenes disponibles en dos grupos bien diferenciados. Por un lado, se ha creado un directorio con un total de 63 imágenes (858 objetos) que se definirá como el conjunto de entrenamiento y corresponderá a las imágenes de *training* que se utilizarán para el diseño del clasificador. Y por otro lado, un directorio con un total de 23 imágenes (200 objetos) que se utilizarán para evaluar el clasificador y corresponderá a las imágenes de *test* que servirán para ver el comportamiento del clasificador a la hora de predecir las etiquetas.

Por tanto, en este apartado se hace necesaria la implementación de un código que permita extraer características de:

- las imágenes de *training,* para entrenar el mejor clasificador posible de forma que se distingan los lúmenes auténticos de los falsos positivos, y
- las imágenes de *test,* para predecir las etiquetas de sus elementos utilizando el clasificador seleccionado.

Etiquetado y extracción de características

Para llevar a cabo el etiquetado y la extracción de características de las imágenes del conjunto de entrenamiento se ha implementado la función "*Features_Training"* (ver lista de funciones) donde el único *input* que se le debe pasar es *"docum"* que se corresponde con el nombre del fichero ".txt" en el que se quiere almacenar la información. Por otra parte, esta función llama internamente a otras funciones implementadas anteriormente como es el caso de la función "*Clustering*" que permitirá trabajar con las máscaras binarias de los 4 elementos que se pueden encontrar en las imágenes histopatológicas teñidas con H&E. (Ver figura 2.12).

Una vez se han extraído las máscaras de los núcleos, el estroma, el citoplasma y el lumen (esta última con la función "*Mask_lumen*") lo que se hace es acceder al directorio donde están guardadas las imágenes binarias que se habían creado con la función "*Auto_segment"* para comparar dichas imágenes de lúmenes con las obtenidas de la función "*Mask_lumen".* De esta forma, se pueden asignar las etiquetas en función de si los elementos de la máscara se corresponden con lúmenes auténticos o con falsos positivos.

En la figura 2.18 se muestra esa comparación entre máscaras y, posteriormente, en la figura 2.19, también se visualizan todos los elementos del procesado que se identifican como lumen a cambio de tener la máxima sensibilidad. Es decir, se muestran también los falsos positivos.

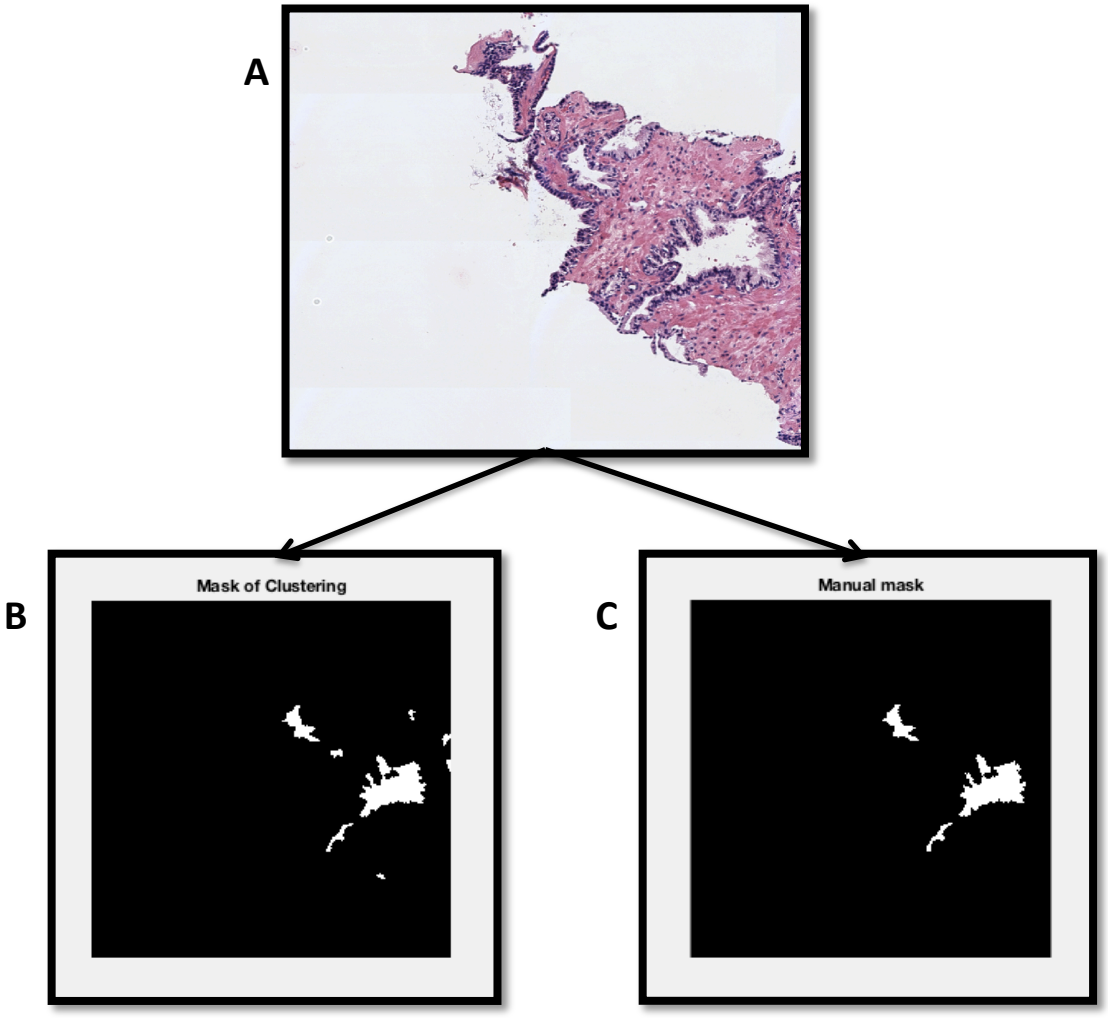


Figura 2.18: A) Imagen original. B) "Mask of Clustering" es la máscara binaria procesada con la función Mask_lumen. C) "Manual Mask" es la máscara binaria creada manualmente con la función Auto_segment.

Una vez se comparan ambas máscaras, se etiqueta con un 0 los falsos positivos y con un 1 los elementos considerados lumen, con la finalidad de que el clasificador sea entrenado con estas etiquetas, además de con las características extraídas de las imágenes de *training.* De esta forma, se pueden predecir las etiquetas de los objetos de las imágenes de *test* que se le pasarán posteriormente.

Es muy importante destacar en este punto que solo van a existir falsos positivos, puesto que el procesado de la función "*Mask_lumen*" ya se había diseñado con la idea de que la sensibilidad fuera máxima. En la siguiente figura se muestran, para la misma imagen que el caso anterior, los falsos positivos y los falsos negativos (estos últimos deberían generar una imagen en negro, pues si la sensibilidad es máxima no puede haber falsos negativos).

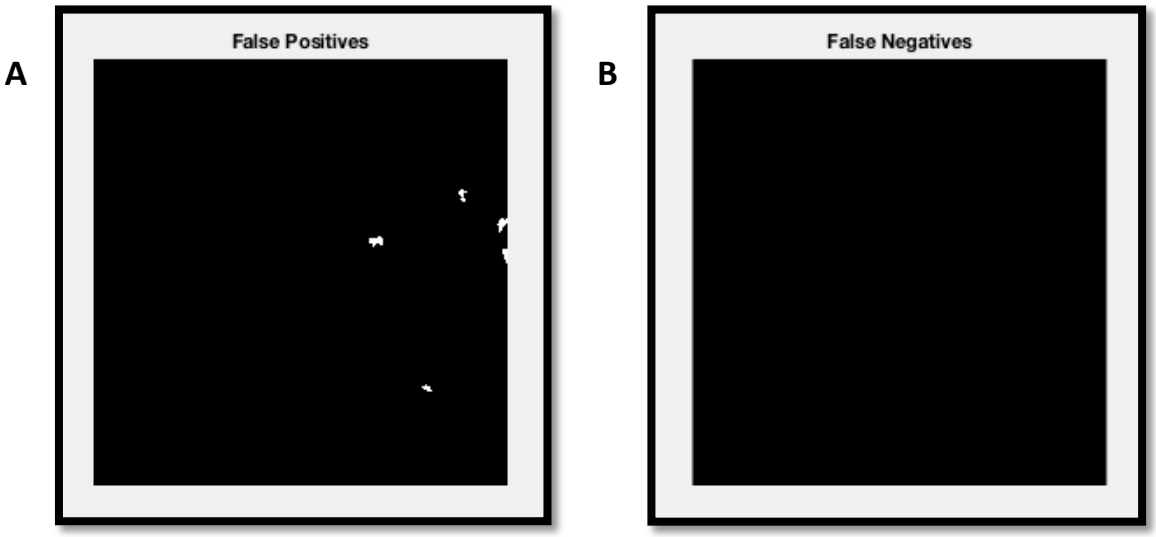


Figura 2.19: A) "False Positives" es la máscara binaria donde se muestran los objetos que en la imagen procesada se habían identificado como lúmenes, pero que realmente no lo son. B) "False Negatives" es la máscara binaria donde deberían verse en blanco los elementos que son lumen y que en la máscara procesada no habían sido identificados como tal.

Efectivamente se corrobora el correcto funcionamiento de los códigos implementados hasta ahora, pues no hay falsos negativos al comparar las máscaras y, por tanto, la sensibilidad es 1.

Para obtener las características con las que entrenar el clasificador se va a trabajar en una pequeña ventana alrededor de cada objeto identificado como posible lumen. Para ello se hace uso del comando "*regionprops*", especificando la opción "*boundingbox*" que permite crear esas ventanas alrededor de cada objeto detectado en la máscara. La técnica del *boundingbox* permite generar un rectángulo que sea el mínimo capaz de albergar por completo el contorno del objeto. A partir de este rectángulo, lo que se hace es aumentar su tamaño en las cuatro direcciones para poder tener en cuenta otros elementos presentes en la ventana que no sean el lumen, pero que estén relacionados con dicho objeto (núcleos, estroma y citoplasma). Esto se puede ver en la figura 2.20, donde en las imágenes A y B se muestra en rojo el rectángulo resultante de la *boundingbox* y en verde el resultante de su ampliación.

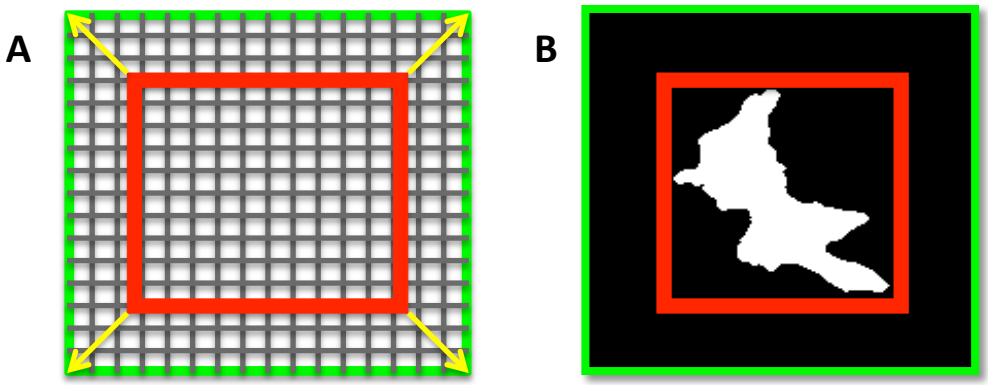


Figura 2.20: A) Imagen que muestra el espacio ocupado por la boundingbox (en rojo) y cómo se expande dicha ventana para poder visualizar los píxeles de alrededor contenidos en la ventana global (verde). B) Ejemplo real para comprender el funcionamiento de la boundingbox y su expansión.

Lo que se quiere conseguir con esto es la extracción de características que permitan establecer relaciones para distinguir los objetos que sean auténticos lúmenes y los que no. Dado que la máscara procesada tiene elementos de ambos tipos, se pueden detectar rasgos característicos alrededor de cada uno para encontrar patrones que correspondan a lúmenes relacionados con glándulas o relacionados con roturas de tejido. Para la imagen ejemplo que se está trabajando, las ventanas definidas alrededor de cada lumen se muestran en la siguiente figura, analizando los píxeles de izquierda a derecha y de arriba a abajo.

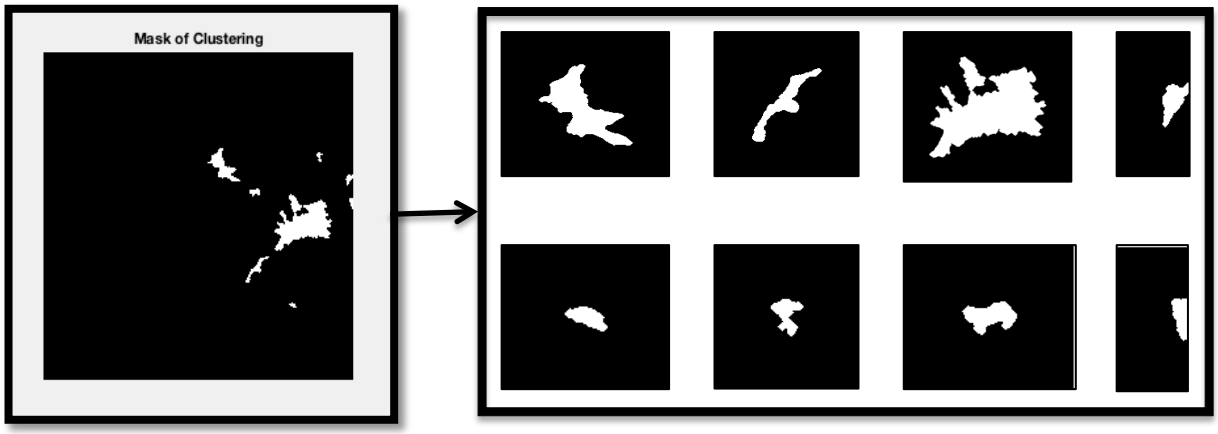


Figura 2.21: Ventanas obtenidas, a partir de ampliar las obtenidas con el comando "boundingbox" para cada posible lumen detectado.

Una vez definidas las ventanas alrededor de cada lumen, el siguiente paso es mostrar el resto de componentes diferenciados en las imágenes de hematoxilina y eosina presentes en dichas ventanas. Para la extracción de características se analizarán estas imágenes viendo qué diferencias son reconocibles en los alrededores de los objetos que son lúmenes de glándulas y los que no.

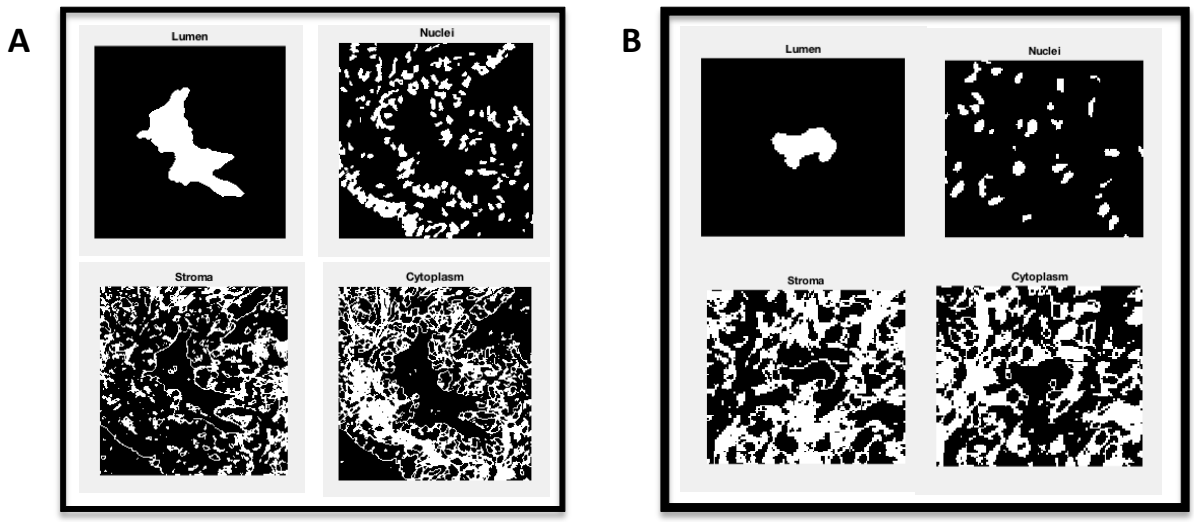


Figura 2.22: A) Máscara binaria de la ventana de los cuatro componentes diferenciables en una muestra de hematoxilina y eosina a partir de un lumen que pertenece a una glándula prostática. B) Mismas imágenes, pero originadas a partir de un elemento que no es lumen. (Ver figura 2.18 para comprobar que no es lumen).

Como se puede observar en la figura 2.22, la información que proporcionan los núcleos, el estroma y el citoplasma en cada ventana es demasiado abundante, lo cual ocasiona un mal entrenamiento, ya que no es posible establecer diferencias significativas entre los ejemplos A y B que permitan determinar que el caso A es un lumen de glándula y el B, no. Por tanto, lo que se hace es aprovechar esas ventanas para analizar la información de los componentes, pero no de toda la ventana (como se hace en la figura 2.22), sino únicamente del contorno que rodea a lo que supuestamente es lumen (ver figura 2.25). Para ello, en primer lugar es necesario llevar a cabo una técnica conocida como "*external gradient*". Esta técnica consiste en realizar una operación en la que primero se dilata la imagen y después se resta la original al resultado de la dilatación, es decir:

$$External\ Gradient\ =\ imagen_{dilatada} - imagen_{original} \tag{2.3}$$

En este punto es conveniente explicar qué son y para qué se utilizan las operaciones morfológicas como la dilatación, la erosión, la apertura y el cierre. El objetivo de las operaciones morfológicas es el estudio de las formas y estructuras geométricas de los objetos, de forma que se mantiene su aspecto esencial, pero se modifican aquellas características no deseadas [24]. La idea principal de esta metodología consiste en examinar las estructuras geométricas de una imagen utilizando lo que se conoce como elemento estructurante (EE). Un EE es una matriz binaria cuyos píxeles a 1 son los únicos que intervienen en las operaciones morfológicas. Además, puede tener varios tamaños y formas en función de la posición que ocupen los píxeles cuyo valor sea 1. Las formas más habituales son el cuadrado, la línea y, sobre todo, el círculo. A continuación se explicará brevemente en qué consisten las cuatro operaciones morfológicas nombradas anteriormente y se pondrá un ejemplo de ellas para diferentes formas de elemento estructurante en la figura 2.23.

- Dilatación. Es una operación morfológica extensiva que permite crecer (engrosar) los objetos blancos que se muestran en una imagen binaria. Se considera extensiva porque aumenta el tamaño de dichos objetos pudiendo llegar a unir aquellos que estén relativamente cerca [20]. La dilatación $\delta_B(X)$, siendo X la imagen y B el EE, puede expresarse matemáticamente de la siguiente forma:

$$\delta_B(X) = \sup\{X_B, b \in B\} = \{B_X, x \in X\} \tag{2.4}$$

$$\delta_B(X) = X \oplus B \tag{2.5}$$

- Erosión. Es una operación morfológica anti-extensiva que permite reducir (estrechar) los objetos blancos que se muestran en una imagen binaria. Se considera anti-extensiva porque disminuye la extensión de dichos objetos pudiendo incluso hacer desaparecer aquellos que son más pequeños que el propio EE [20]. La erosión $\varepsilon_B(X)$, siendo X la imagen y B el EE, puede expresarse matemáticamente de la siguiente forma:

$$\varepsilon_B(X) = \inf\{X_B, b \in B^*\} = \{z, B_Z \in X\} \tag{2.6}$$

$$\varepsilon_B(X) = X \ominus B \tag{2.7}$$

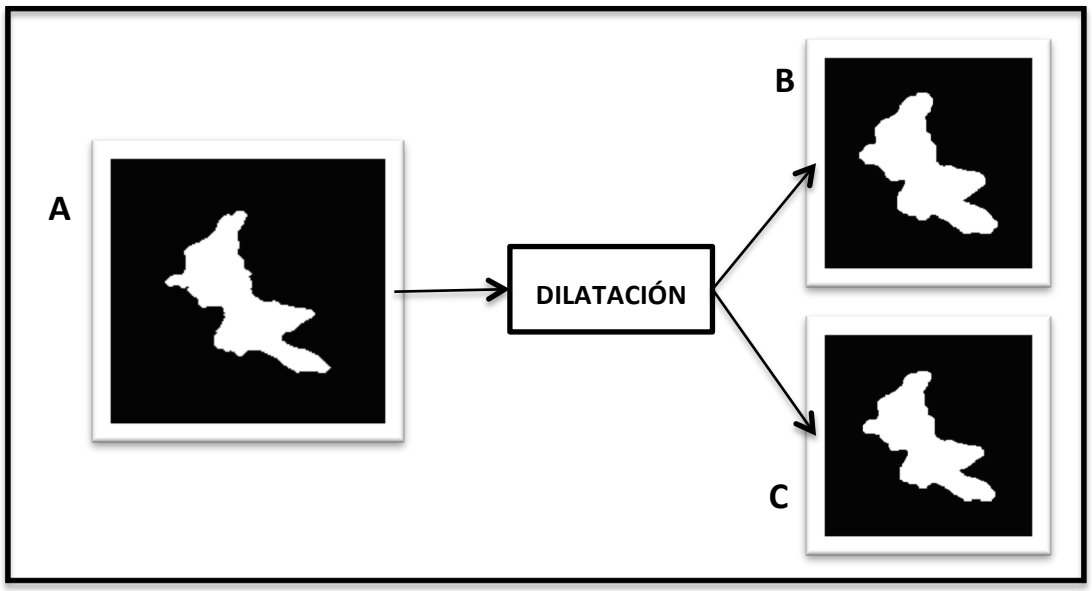


Figura 2.23: Ejemplo de dilatación con distintas formas de EE y con un tamaño constante de 10 píxeles. A) Imagen de partida. B) Imagen dilatada con un EE circular. C) Imagen dilatada con un EE cuadrado.

Para el ejemplo propuesto solo se aprecian pequeñas diferencias de grosor que no tienen una gran relevancia en el resultado final. No obstante, esta operación morfológica puede convertirse en un factor determinante cuando el mismo procedimiento se lleva a cabo por ejemplo para una imagen cuyos elementos están separados a una distancia muy pequeña. En ese caso, al realizar la dilatación, los objetos se harían más grandes y podrían entrar en contacto, de tal forma que se tendrían menos elementos más grandes en lugar de más elementos de menor tamaño.

En base a las características observadas, se puede determinar lo siguiente sobre la dilatación:

- Equivale a un máximo deslizante del ancho del EE.
- Elimina aquellas zonas oscuras (picos negativos) que son más estrechas que el EE.
- Estrecha las zonas oscuras que son más anchas que el EE.
- Ensancha las zonas claras, es decir, los picos positivos.
- Siempre está por encima de la señal original.

Por otra parte, en la figura 2.24 se muestra para el mismo ejemplo de la figura anterior, lo que sucedería si, en lugar de realizar una dilatación, se lleva a cabo una erosión con los mismos elementos estructurantes. Sobre el efecto de la erosión sobre la imagen se puede determinar que:

- Equivale a un mínimo deslizante del ancho del EE.
- Elimina aquellas zonas claras (picos positivos) que son más estrechas que el EE.
- Estrecha las zonas claras que son más anchas que el EE.
- Ensancha las zonas oscuras, es decir, los picos negativos.
- Siempre está por debajo de la señal original.

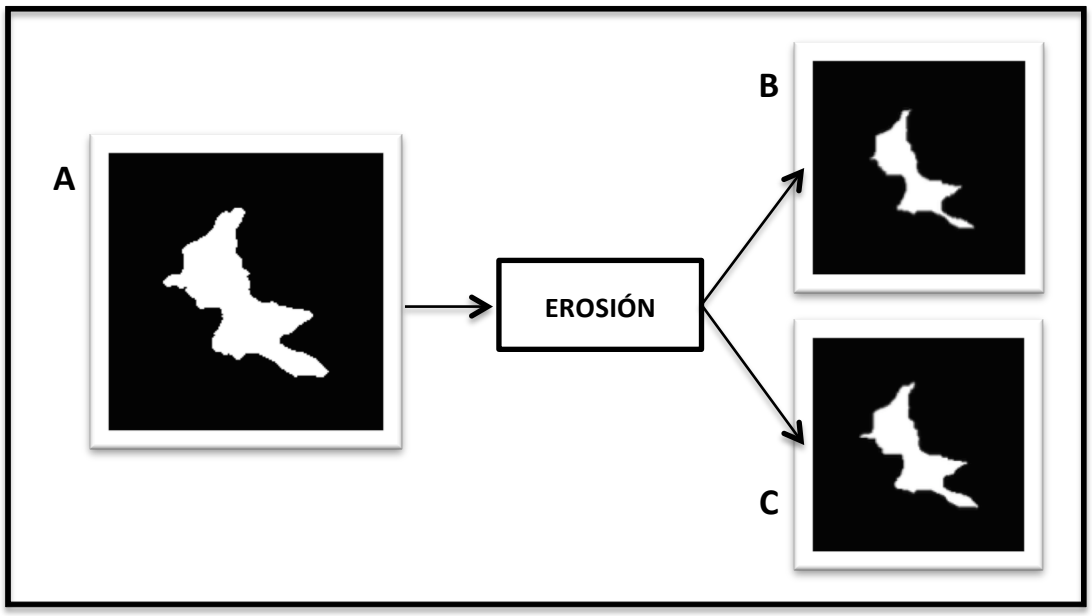


Figura 2.24: Ejemplo de erosión con distintas formas de EE y con un tamaño constante de 10 píxeles. A) Imagen de partida. B) Imagen erosionada con un EE circular. C) Imagen erosionada con un EE cuadrado.

Una vez conocido el funcionamiento de las operaciones morfológicas más básicas utilizadas para este proyecto, se puede entender la ecuación 2.3 que hace referencia a la técnica del "gradiente externo" mediante la siguiente figura:

Figura 2.25: Proceso en el que a partir de la imagen (A) compuesta por los cuatro componentes se obtiene, mediante la función "Mask_lumen" (a), la máscara binaria del lumen (B) y a partir de ella, mediante la técnica del "gradiente externo" (b), la imagen C).

Como ya se ha comentado, la idea de utilizar esta técnica consiste en estudiar características relacionadas con la distribución y la densidad de objetos como los núcleos, el citoplasma o el estroma presentes en la imagen, pero en lugar de en toda la ventana, únicamente alrededor del contorno del lumen. Para ello, la dilatación llevada a cabo en la técnica del *"external gradient"* se ha realizado con un EE de disco y 25 píxeles de tamaño. Esto permite tener en cuenta otras características como la presencia de componentes posicionados en la misma orientación del lumen. Un ejemplo de esto se muestra en la siguiente figura:

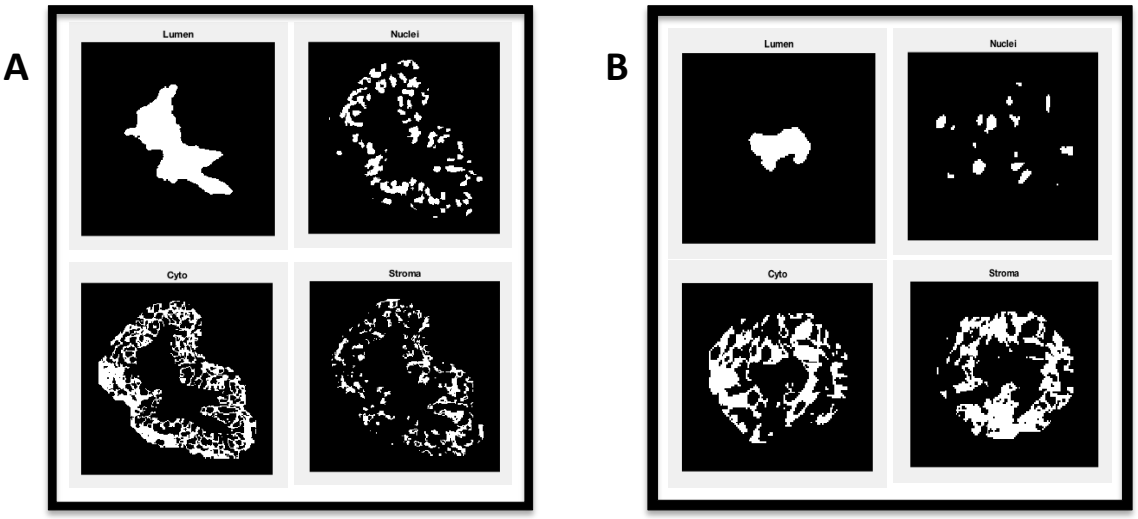


Figura 2.26: En el caso A se observan los núcleos, el citoplasma y el estroma presentes alrededor de un lumen auténtico, concretamente en el espacio generado por la técnica del "external gradient". En el caso B se observan los mismos componentes, pero para objeto que no es lumen, es decir, un falso positivo.

Si se compara la figura anterior con la 2.22 se pueden distinguir las diferencias entre ambas y determinar que efectivamente la extracción de características es más precisa utilizando la técnica del "gradiente externo" que analizando todos los componentes que aparecen en la ventana ampliada.

El siguiente paso de la función "*Features_Training*" consiste en escoger las características que se van a utilizar como entrada de los clasificadores para entrenarlos y para que diferencien automáticamente aquellos objetos que sean lúmenes y aquellos que no. Por tanto, lo que se tendrá que buscar serán patrones característicos y excluyentes en cada tipo de objeto identificado con la finalidad de realizar la mejor clasificación posible. Para ello, en la figura 2.27, a modo de ejemplo, se muestran dos elementos que son lúmenes auténticos y dos elementos que son falsos positivos (de la misma imagen que se analiza en todo el apartado). Además, para cada uno de esos cuatro elementos se observan también, en la figura 2.28, los cuatro componentes que aparecen en las ventanas creadas con el fin de identificar los patrones más significativos de cada tipo.

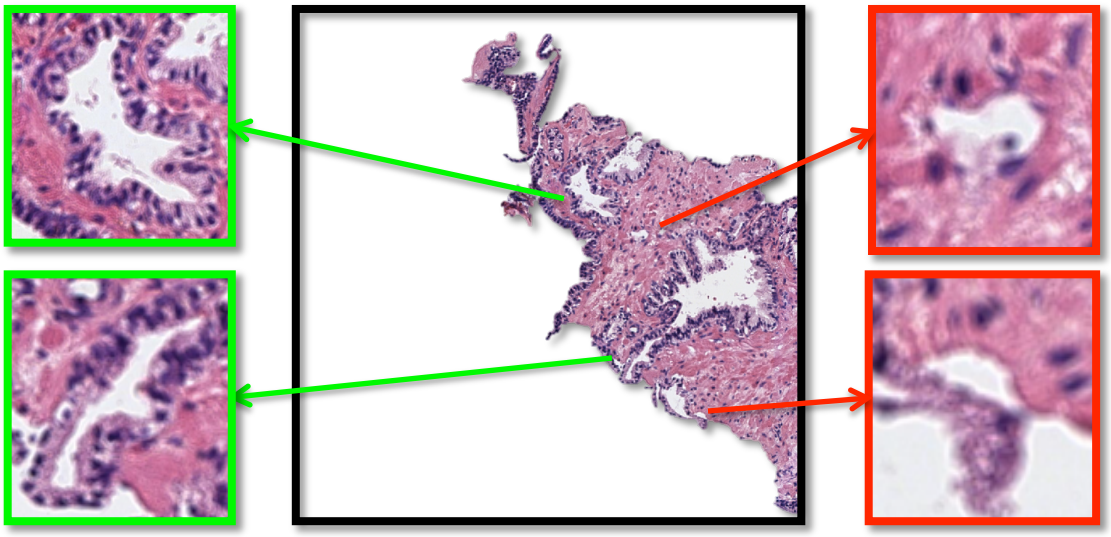


Figura 2.27: Glándulas que presentan lúmenes auténticos (en verde) y lúmenes falsos (en rojo).

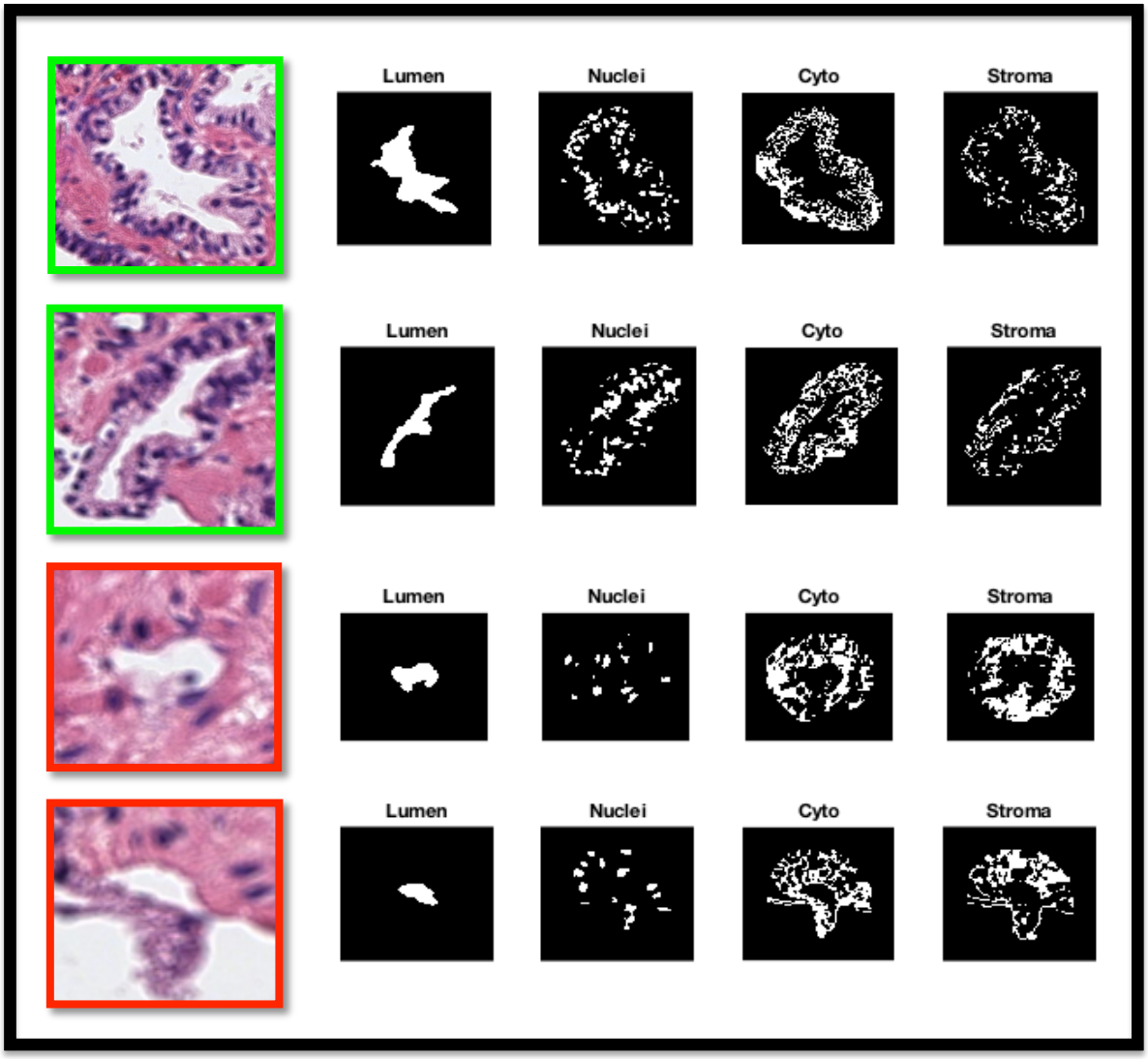


Figura 2.28: Visualización de las máscaras correspondientes a los cuatro componentes que se pueden observar en las ventanas creadas alrededor de cada objeto identificado como lumen.

Basándose en varios ejemplos como el que se muestra en la figura anterior, se define para la función "*Features_Training* una lista con las características que a priori pueden resultar interesantes. A partir de ahí, es posible realizar después una combinación de dichas características y quedarse con aquellas que sean más excluyentes, de tal manera que la detección sea más precisa. La lista de características se divide en cuatro grandes grupos según el componente del que se extraen, es decir, según si se extraen del lumen, de los núcleos, del citoplasma o del estroma. A continuación se detalla cada uno de ellos.

- Características extraídas del lumen:
  - "*Solidity*" → Hace referencia a cómo de puntiaguda es la forma del objeto que se está analizando. Aparentemente, aunque no siempre, los elementos correspondientes a lúmenes reales tienen un aspecto más fusiforme, mientras que los objetos que no son lúmenes auténticos presentan un aspecto más redondeado.

- *"Circularity"* → Hace referencia a la compacidad o circularidad que tiene el elemento identificado como lumen en este caso. Para calcular este parámetro es necesario utilizar la función *regionprops* para definir el área y el perímetro del objeto, ya que:

$$Circularidad = {Área}/{Perímetro^2} \quad (2.8)$$

- Características extraídas de los núcleos:
  - Número de píxeles totales → Se calcula el área que ocupa cada objeto detectado como núcleo independiente en la ventana (ver figura 2.28). Después se suman los píxeles que ocupan las áreas de todos los núcleos y se obtiene el número total de píxeles. El inconveniente que puede presentar esta característica es que el tamaño de la ventana viene determinado por el tamaño del lumen, por lo que el número de núcleos dependerá de las dimensiones del objeto identificado.
  - Proporción de píxeles → Para solventar el problema que presentaba la característica anterior, en este caso lo que se hace es dividir el total de píxeles entre el tamaño de la ventana. De esta forma, se puede tener en cuenta el número de píxeles ocupados por los núcleos independientemente del tamaño del lumen.

- Características extraídas del citoplasma:
  - Número de píxeles totales → Mismo procedimiento que para los núcleos, pero con la máscara del citoplasma.
  - Proporción de píxeles → Mismo procedimiento que para los núcleos, pero con la máscara del citoplasma.

- Características extraídas del estroma:
  - Número de píxeles totales → Mismo procedimiento que para los núcleos y el citoplasma, pero con la máscara del estroma.
  - Proporción de píxeles → Mismo procedimiento que para los núcleos y el citoplasma, pero con la máscara del estroma.
  - Número de objetos identificados → Hace referencia a la cantidad de elementos en blanco que aparecen separados en la imagen. Dos elementos están separados cuando no tienen componentes conexos entre ellos, es decir, cuando no tienen píxeles puestos a 1 en común.
  - Proporción de objetos identificados → Partiendo de la misma idea de analizar el número de elementos de estroma independientemente de cuál sea el tamaño del posible lumen identificado, se divide el total de objetos de estroma entre el tamaño de la ventana.
  - Área media de cada elemento → En lugar de calcular el área ocupada por todos los píxeles en blanco que aparecen en la ventana y después dividir entre el número de elementos, lo cual correspondería al área media de la ventana, lo que se hace es calcular el área de cada elemento y después hacer la media de esas áreas.

De toda la lista de características, finalmente solo se han tenido en cuenta cuatro de ellas, puesto que el resto eran redundantes y no aportaban información complementaria, lo cual puede ocasionar que se produzca lo que se conoce como *overfitting*[11]. Las cuatro características de la lista que proporcionan los mejores resultados son:

1. "*Solidity*" del lumen. Hace referencia, como ya se ha comentado, a la geometría que presenta el elemento identificado. Se define como la proporción de píxeles en la *convex hull* (casco convexo) que están presentes en la región, es decir, corresponde al número de píxeles que ocupa el elemento en relación con el área del polígono que se forma al unir sus vértices.

$$Solidity = {Área}/{Convex\ hull} \qquad (2.9)$$

De esta forma, cuanto más fusiforme es el aspecto que presenta el objeto, el parámetro *solidity* es menor, pues el área que ocupa es más pequeña comparada con la *convex hull.* Sin embargo, los elementos más circulares presentan un valor de *solidity* más cercano a 1. En la siguiente figura se muestran todos los valores de *solidity* para los objetos identificados en la imagen ejemplo y poder analizar visualmente la explicación.

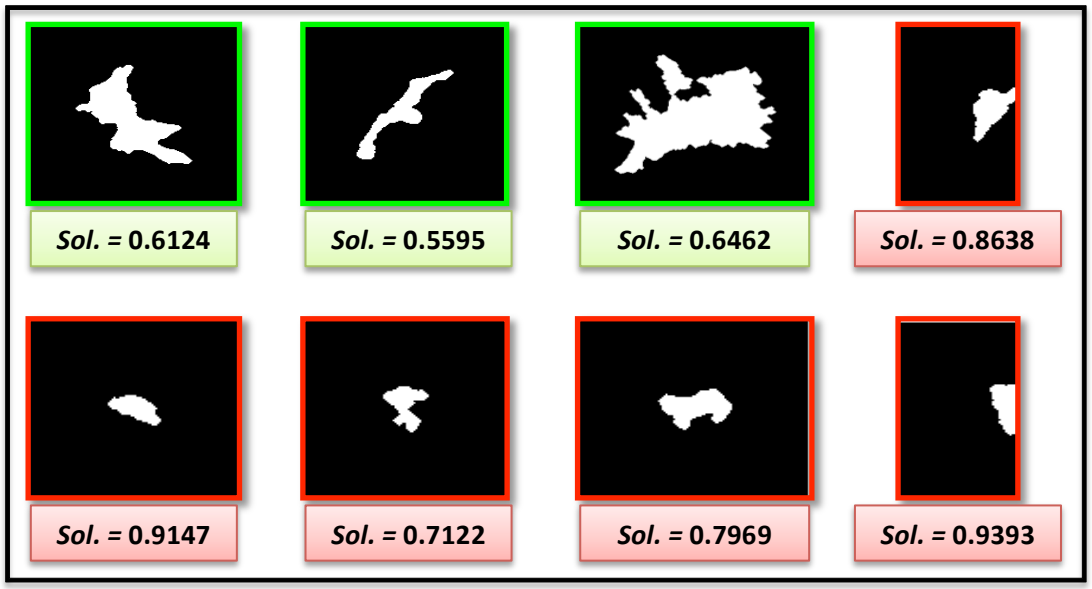


Figura 2.29: Objetos identificados como posibles lúmenes en una de las imágenes estudiadas donde se pueden observar los valores del parámetro "solidity" para cada uno de ellos. En verde se muestran aquellos que son lúmenes auténticos y en rojo los falsos positivos.

Se puede comprobar que, efectivamente, el parámetro *solidity* puede ser una característica interesante, pues normalmente se cumple que para un umbral por debajo de cierto valor los objetos presentan geometrías más fusiformes, lo cual se corresponde con la autenticidad de los lúmenes.

[11] *Overfitting*. Su traducción literal en castellano es "sobreajuste". Se da cuando el modelo se ajusta muy bien a los datos existentes, pero tiene un pobre rendimiento para predecir nuevos resultados [25].

2. Proporción de píxeles de los núcleos. Se procede de la misma forma que en el caso anterior para explicar el interés de esta característica. Se hará únicamente con los objetos de la figura 2.28 para no extenderse en exceso.

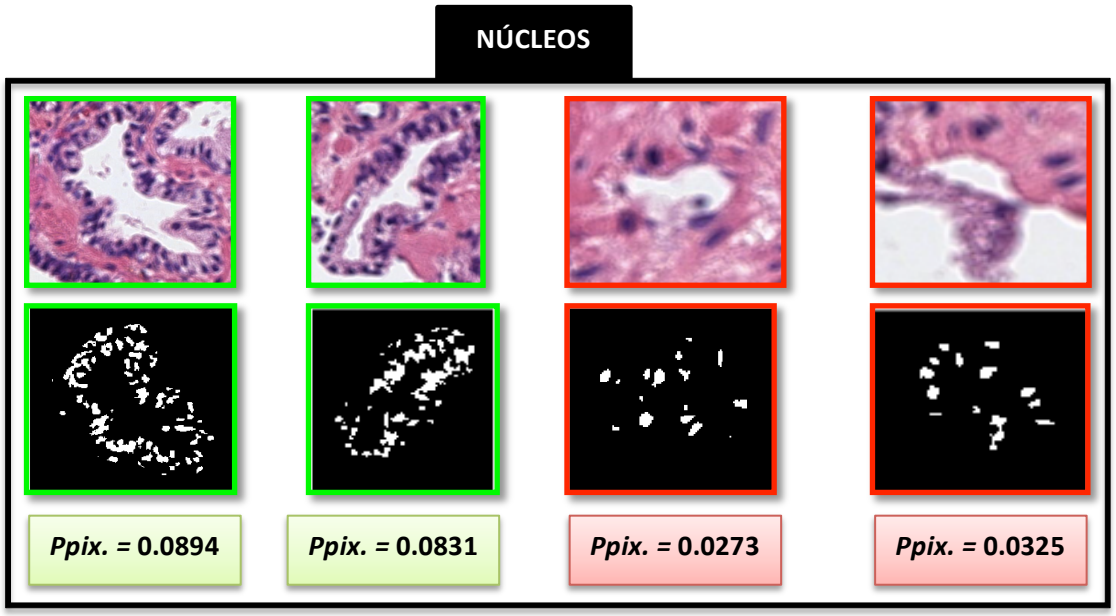


Figura 2.30: Modelo donde se muestran las máscaras de núcleos y los valores de la característica "Proporción de píxeles de núcleos" para cada elemento identificado como posible lumen. En verde se muestran aquellos que son lúmenes auténticos y en rojo los falsos positivos.

Se demuestra que esta característica también es excluyente y, por tanto, de interés para el procedimiento de clasificación, ya que puede existir un valor umbral por encima del cual las muestras correspondan a lúmenes auténticos y por debajo, no.

3. Proporción de píxeles de estroma. En este caso, la forma de actuar sería exactamente igual que para la característica de la proporción de píxeles de los núcleos, pero llevada a cabo con la máscara del estroma. Los valores para corroborar su elección se muestran en la figura 2.31 en combinación con la cuarta característica.

4. Número de objetos de estroma. La idea de elegir esta característica es fruto de observar que las glándulas necesariamente deben estar compuestas por citoplasma. Sabiendo esto, al utilizar la técnica del "gradiente externo", cuando un objeto es glándula debe presentar un alto contenido de componente de citoplasma, lo cual conlleva un escaso contenido de estroma. Por tanto, si aparece estroma serán pequeños elementos sueltos que se han confundido con el citoplasma, pero no habrá una gran cantidad de píxeles conexos, sino una mayor cantidad de objetos más pequeños como se muestra en la figura 2.31. La aparición de estroma en la región de interés, cuando se analiza una muestra correspondiente a una glándula, está relacionada con la tinción de las muestras, ya que en muchas ocasiones el color que representa el citoplasma se confunde con el del estroma y, por tanto a la hora de realizar el *clustering,* la clasificación no es óptima y aparece estroma cuando no debería. Un ejemplo de esto se puede observar en la figura 2.32, donde se muestran dos imágenes cuya tinción puede llevar a resultados diferentes, aunque se haya realizado con los mismos pigmentos.

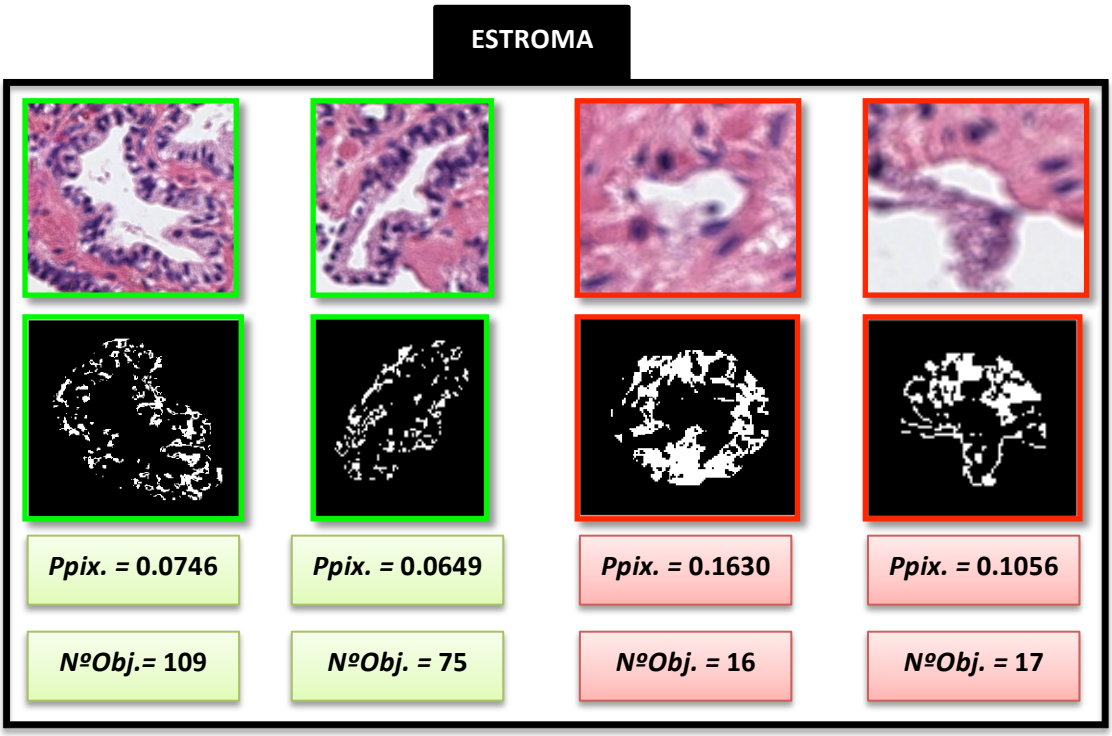


Figura 2.31: Modelo donde se muestran las máscaras del estroma y los valores de las dos características que se obtienen para esos casos concretos. En verde se muestran aquellos que son lúmenes auténticos y en rojo los falsos positivos.

Se corrobora también para estas características su carácter discriminante a la hora de distinguir lúmenes auténticos. Cabe destacar que los resultados para las imágenes del ejemplo han sido satisfactorios, habiendo utilizado como ejemplo una imagen escogida al azar. En la base de datos disponible hay imágenes en las que son mucho más diferenciables los lúmenes auténticos de los falsos y, por tanto, los valores de las características son todavía más extremos. Con la combinación de las cuatro características explicadas se entrenarán los clasificadores y se mostrará el resultado del procedimiento en el capítulo 3 del informe.

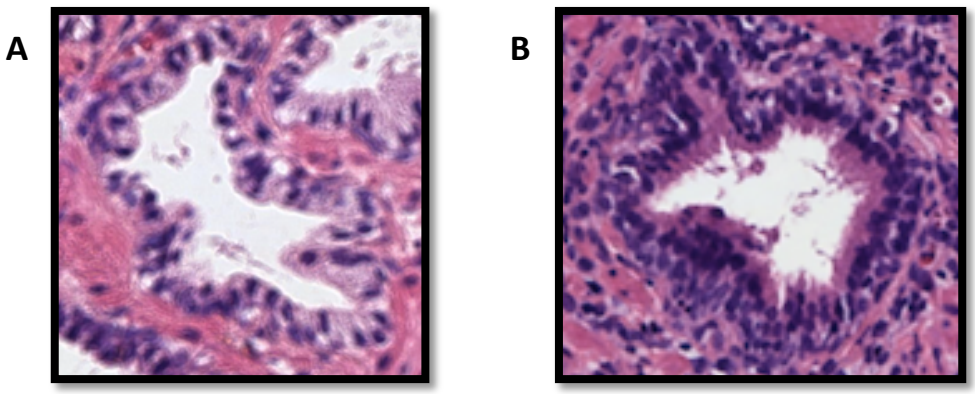


Figura 2.32: A) Glándula de una imagen utilizada para training. B) Glándula de una imagen utilizada para test. Se puede observar que la imagen A tiene una tinción más confusa, pues el estroma y el citoplasma presentan colores muy similares, mientras que en la imagen B las diferencias de color se aprecian más fácilmente. Por tanto, el resultado del clustering será mejor en el caso B.

Por último, en la función "*Features_Training"* se debe crear la etiqueta, es decir, la variable que indique si cada uno de los objetos identificados como posible lumen en una imagen es realmente un lumen o no. El objetivo es otorgar una etiqueta con un 0 a aquellos objetos que sean falsos lúmenes y con un 1 a los que sean auténticos. De este modo, la etiqueta será una de las entradas de los clasificadores que podrán entrenarse teniendo en cuenta muestras que ya están etiquetadas, de ahí el nombre de "clasificación supervisada". Para ello, lo que se debe hacer es ver qué objetos de la máscara creada con la función "*Mask_lumen"* coinciden con los objetos de la máscara creada con la función "*Auto_segment"*. Aquellos elementos que coincidan serán lúmenes auténticos y los que no, falsos positivos, es decir, se les pondrán las etiquetas 1 y 0, respectivamente.

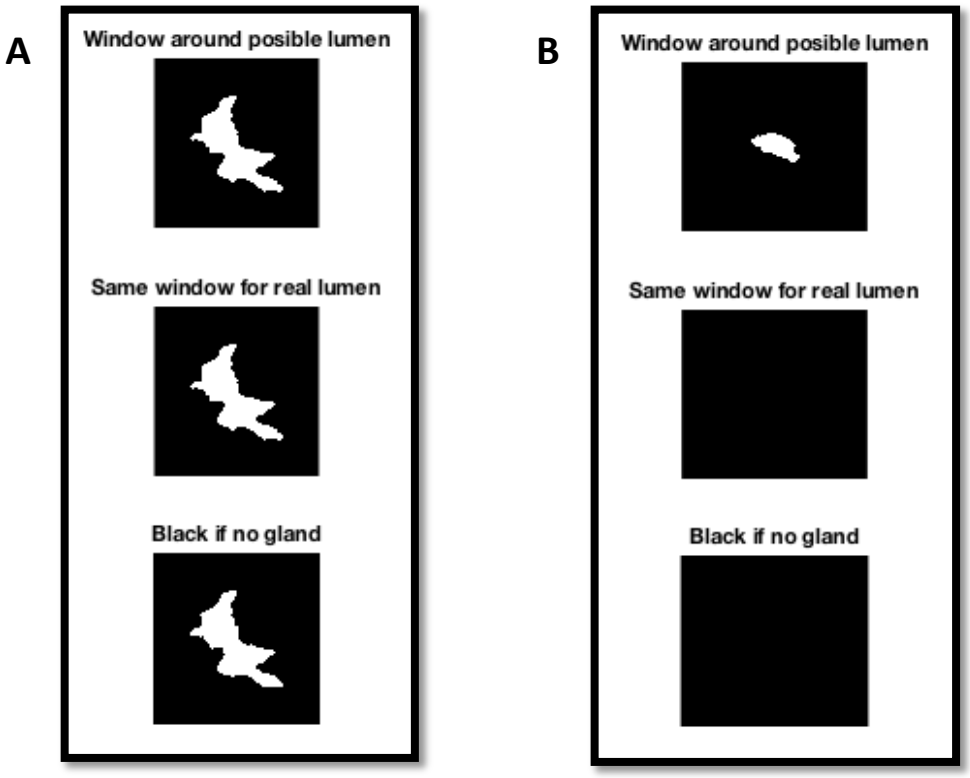


Figura 2.33: A) Modelo donde el objeto de la máscara de "Mask_lumen" coincide con el objeto localizado en la misma posición para la máscara de "Auto_segment". Por tanto, se etiqueta dicho objeto con un 1. B) Modelo donde el objeto de la máscara de "Mask_lumen" no coincide ningún objeto en la misma posición para la máscara de "Auto_segment". Por tanto se etiqueta dicho objeto con un 0.

Tras la ejecución final de la función "*Features_Training"* se obtiene un archivo *.mat que contiene la información de todos los objetos analizados. Es decir, el *output* de la función consiste en una matriz que tiene tantas filas como elementos detectados en las imágenes de *training* (un total de 860 para 63 imágenes) y tantas columnas como características, es decir, 5 columnas. Las cuatro primeras corresponden a las características con las que se entrenan los clasificadores y la quinta a la etiqueta (1 o 0) que indica si es un lumen o no.

Por otra parte, la extracción de características también se lleva a cabo para las imágenes de test, donde el procedimiento es exactamente igual que para las imágenes de *training*. La única diferencia es que no hay que tener en cuenta las máscaras obtenidas con la función *"Auto_segment",* ya que ahora no se necesita una variable respuesta. La etiqueta de 0s y 1s solo se utiliza para entrenar a los clasificadores, de tal forma que estos puedan predecir la etiqueta que le correspondería a los objetos de la nueva imagen que se analiza. Es decir, se hará uso de las cuatro características con las que se han entrenado los 858 elementos como los que se visualizan en la figura 2.29 para predecir si a los elementos de las nuevas imágenes (de test) les corresponderá la etiqueta 1 o la etiqueta 0 en función de si son lúmenes auténticos o no, respectivamente.

Para llevar a cabo la extracción de características de las imágenes utilizadas para *test* (un total de 23 imágenes con 200 objetos) se ha implementado la función *"Features_Test",* similar a la de *"Features_Training".* En este caso, los *inputs* son (i) *read_directory,* que se refiere al directorio donde se encuentra la imagen de test a analizar y (ii) *image,* que es la imagen en sí. En cuanto a los *outputs,* el principal es *"Data"* que consiste en una matriz similar al archivo *.mat que se generaba con la función *"Features_Training",* pero en este caso con las imágenes de test y sin la variable correspondiente a las etiquetas. Por tanto, la variable *"Data"* será una matriz compuesta por tantas filas como elementos hayan sido detectados en las imágenes de test y por tantas columnas como características se hayan extraído, que en este proyecto han sido cuatro. Este vector "*Data*" será el que se le pase posteriormente al clasificador para llevar a cabo la predicción de las etiquetas.

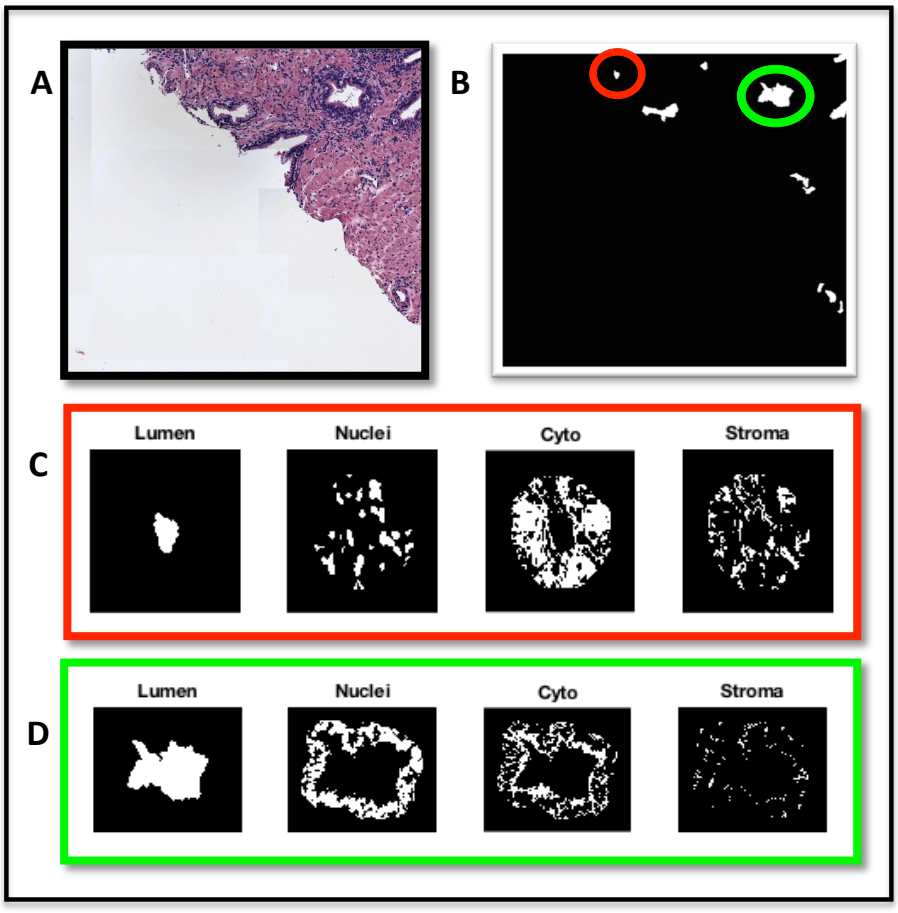


Figura 2.34: A) Imagen original utilizada para test. B) Máscara de lúmenes obtenida de la función "Mask_lumen". C) Máscaras de un falso lumen a partir de las cuales se extraen los valores de características que se comparan con los obtenidos del training. D) Mismas máscaras que el ejemplo C, pero para un lumen auténtico.

2.3.4.3 Clasificación y predicción

Para llevar a cabo la clasificación en sí se implementa una nueva función a la que se le ha denominado "*Classify".* En esta función, lo primero que se hace es llamar al archivo *.mat que se había creado con la función "*Features_Training"* en el primer apartado del 2.3.4. De esta forma se cargan tanto las características extraídas de las imágenes de *training* (x) como la variable respuesta (y) que indica para cada elemento si es un lumen auténtico o no. A continuación, es necesario realizar una normalización de las características (x) mediante el comando *zscore* que normaliza cada característica respecto a su media y su desviación típica, de forma independiente. Cuando ya se han preparado las características y la etiqueta procedentes de las imágenes del conjunto de *training*, es posible utilizarlas como entrada para entrenar los clasificadores. En este punto cabe destacar que se ha utilizado una de las nuevas funcionalidades de las que dispone la última versión del software de MATLAB (ver apartado 2.2.2) llamada "*Classification Learner"* que permite evaluar una gran cantidad de clasificadores (árboles de decisión, análisis discriminante, clasificadores de regresión logística, de *support vector machine*, de vecinos más cercanos, etc.) atendiendo a las características de entrenamiento y a la etiqueta que se le pasan como entrada.

Para generar el mejor clasificador lo que se hace es utilizar la técnica llamada *cross-validation* utilizando 5 *folds* [26], lo cual permite dividir el propio conjunto de *training* en dos grupos: un grupo, formado por el 80% de las muestras, que se utiliza como conjunto de entrenamiento; y otro grupo (formado por el 20% de las muestras) que se utiliza como conjunto de validación. Este procedimiento se realiza 5 veces (por los 5 *folds)*, de tal forma que todas las muestras son utilizadas para validación y para entrenamiento. Esto permite generar un modelo de clasificador muy generalizado con la finalidad de que, posteriormente, puedan pasársele como *input* las características de nuevas imágenes y ver cómo funciona el clasificador para esas nuevas muestras. En otras palabras, permite llevar a cabo una predicción de etiquetas para los objetos de nuevas imágenes que se necesite analizar en un futuro. Para simular este procedimiento, se habían separado, en un principio, 23 imágenes que el clasificador no ve en ningún momento para entrenar, y se les había asignado el nombre de "imágenes de *Test"*. De esta forma, cuando se haya generado el modelo del clasificador (mediante *cross-validation*) se le pasarán como entrada las características de las imágenes de test que permitirán obtener los resultados de especificidad, sensibilidad, etc. para dichas imágenes.

La nueva funcionalidad de MATLAB permite generar finalmente un valor de *accuracy* (precisión) para cada uno de los clasificadores evaluados. De esta forma, se puede seleccionar el clasificador que mejor precisión tenga y generar una nueva función llamada "*trainClassifier"* que tiene como *inputs* las características y la variable respuesta, y como *output* el clasificador entrenado*. (Nota: Para ver los resultados de la precisión, la curva ROC y la matriz de confusión del clasificador seleccionado, ir al capítulo 3, Resultados).*

El clasificador que proporciona mejores resultados tras realizar el *cross*-validation es el *support vector machine* lineal. Este tipo de clasificador (traducido como "máquinas de vector soporte") es de tipo no paramétrico basado en funciones discriminantes lineales. Además, los SVM son dicotómicos, pues se limitan a diferenciar dos clases distintas. Tratan de hallar un hiperplano que separe en dos regiones el espacio muestral, considerando como "hiperplano óptimo" aquel que se genera cuando la distancia entre dos puntos más cercanos de cada clase al hiperplano es máxima. El procedimiento del clasificador SVM consiste, en primer lugar, en definir un plano que separe los datos en las dos

clases. A continuación se debe establecer una frontera que permita obtener el mayor margen posible. Es decir, se busca el hiperplano que tenga el mayor margen, ya que ese será el mejor clasificador de los datos. En ese momento, se consideran "vectores soporte" a los puntos que tocan el límite del margen. Finalmente, con este modelo de clasificación se obtiene una frontera lineal que depende de un producto escalar entre vectores soporte. Gracias a que todo depende de ese producto escalar, se puede implementar una dimensión extra en la que los datos son separables. Esto presenta la ventaja de que en vez de tener que calcular cada vector proyectado en la nueva dimensión, se puede hacer la transformación directa del producto, de tal forma que no es necesario hacer la transformación de cada uno de los vectores escalares [27].

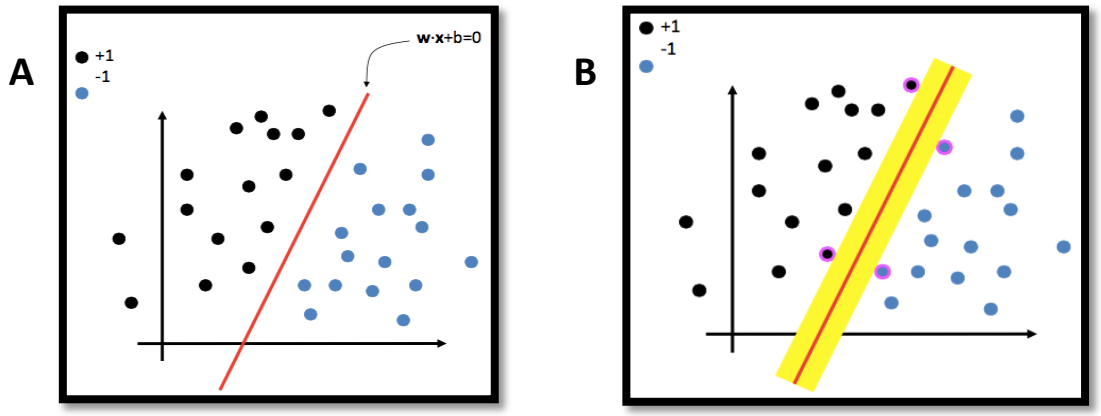


Figura 2.35: A) Se define el hiperplano. B) Se establece el hiperplano que tiene el mayor margen. Se alcanza a ver que los puntos en contacto con el límite del margen están identificados con otro color por ser esos los vectores de soporte.

Una vez se conoce que el mejor modelo de clasificación proporcionado es el SVM, y observando que la etiqueta que proporciona es binaria (0 o 1), lo que se ha hecho es implementar dentro de la función "*Classify*" el mismo clasificador SVM, pero obteniendo como resultado la probabilidad de pertenecer a la clase 0 o a la clase 1. De esta forma, es posible fijar un nivel umbral a partir del cual etiquetar, con un 0 o con un 1, la nueva muestra analizada. Lo que se pretende entonces es utilizar el modelo del clasificador óptimo proporcionado por la app de Matlab e implementar el mismo modelo de clasificador, pero modificando el umbral por defecto para estudiar cómo varía la especificidad, en función de la variación de dicho umbral, a la hora de segmentar las glándulas. Por tanto, si el modelo de la app utilizaba un umbral de 0.5, donde los elementos etiquetados por encima de dicho umbral eran clasificados como lúmenes, en este caso se fijará el umbral en 0.9 con el objetivo de tener más especificidad a partir del mismo modelo. De esta forma, los elementos que sean etiquetados con valores por encima de 0.9 serán clasificados como "lumen" y los que estén por debajo, como "no lumen". No obstante, aprovechando que el clasificador proporciona valores decimales en sus etiquetas, se puede hacer un subgrupo dentro de los objetos clasificados como "no lumen", tal que aquellos cuya etiqueta esté entre 0.5 y 0.9 sean definidos como "lúmenes dudosos". Esto permite llevar a cabo un procedimiento en el que la segmentación de glándulas sea menos sensible, pero más específica, a la vez que se tienen en cuenta posibles glándulas cuya clasificación no era tan clara. De este modo, esos elementos clasificados como "lúmenes dudosos" pueden analizarse en un cribado posterior donde se cuente con la ayuda de los patólogos para decidir, sin lugar a error, su clase correspondiente.

Destacar que la clasificación se ha realizado con el SVM para los dos umbrales (el de por defecto y el modificado) para estudiar los resultados que se obtienen con cada uno y analizar si la estrategia de perder en sensibilidad, a cambio de ganar en especificidad, merece la pena o no.

A continuación, lo que se lleva a cabo es la predicción de nuevas muestras pertenecientes a las imágenes de test. Para ello, los clasificadores necesitan como *input* las características de los objetos de dichas imágenes que se habían obtenido en la función "*Features_test*". Finalmente, la función "*Classify*" a partir de sus *inputs,* que son (i) *read_directory* y (ii) *image*, haciendo referencia al directorio donde se encuentran las imágenes de test y a la imagen de test en sí, permite obtener como *outputs (i) c_svm, (ii) app_svm,* que son las etiquetas que predice el clasificador para cada umbral, (iii) *mask_svm*, (iv) *mask_app* que son las máscaras donde solo los elementos clasificados con la etiqueta 1 se muestran en blanco y, por último (v) *pos_lum,* que corresponde a la máscara de los objetos clasificados como "lúmenes dudosos" con el SVM del umbral modificado.

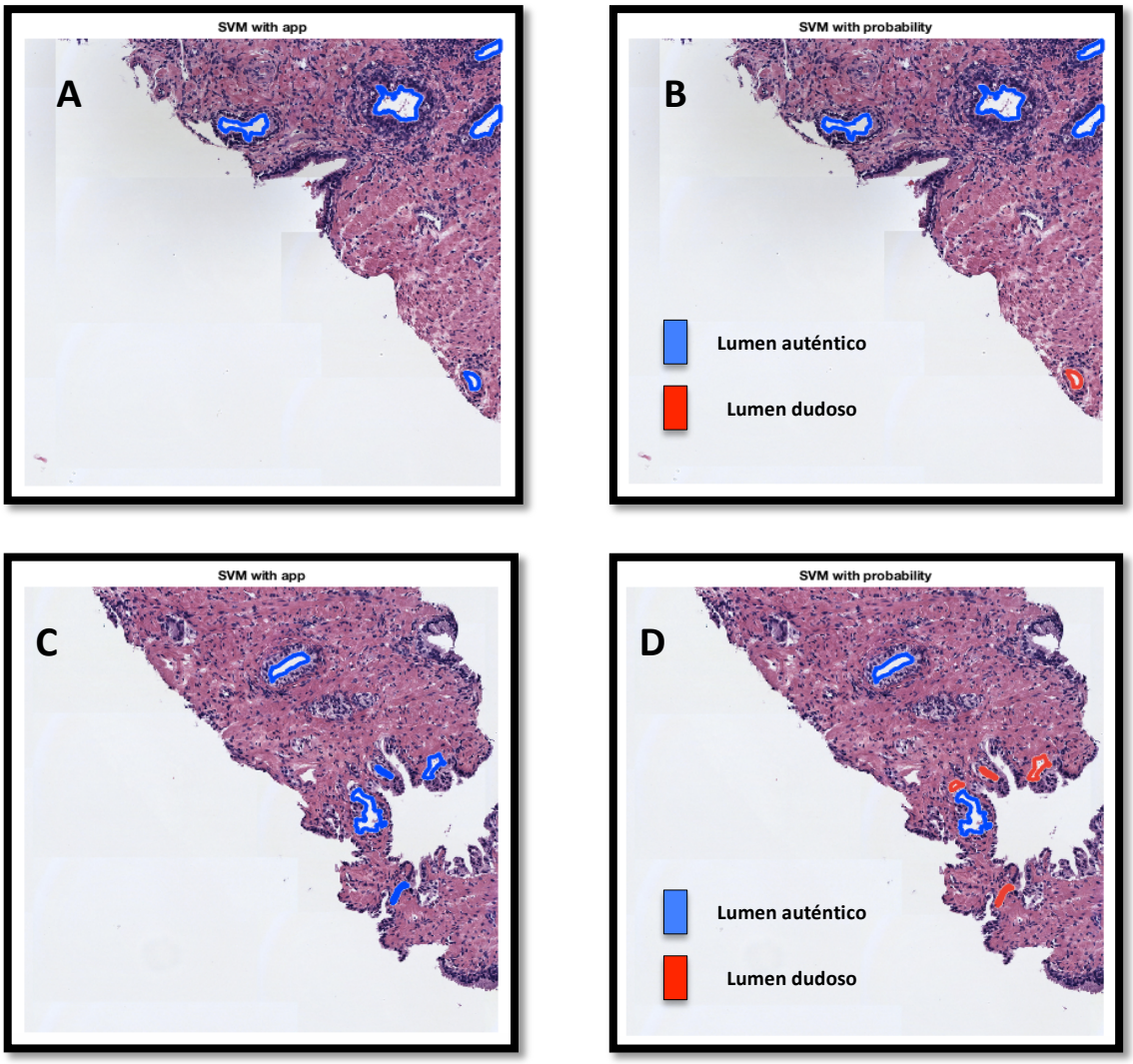


Figura 2.36: A) y C) Imágenes analizadas con el clasificador SVM y el umbral por defecto. B) y D) Imágenes obtenidas modificando el umbral para el mismo clasificador. Se observa en estas imágenes los lúmenes cuya etiqueta está entre 0.5 y 0.9, de tal forma que se marcan en rojo para su cribado posterior.

2.3.5 Segmentación automática de glándulas

En este TFG se ha utilizado el método de segmentación conocido como *watershed.* Dentro de las diferentes técnicas de segmentación, este tipo se basa principalmente en dividir la imagen de interés en regiones examinando los píxeles existentes y agrupándolos en caso de que cumplan unas condiciones especificadas en un principio.

2.3.5.1 Watershed

La técnica *watershed* se basa en considerar la imagen inicial en escala de grises como un relieve topográfico donde se pueden distinguir dos tipos de elementos: las cimas, que serían los máximos locales, y los valles, que harían referencia a los mínimos locales. De esta forma, se divide la imagen en las zonas de influencia de cada mínimo local simulando un proceso de inundación [20].

Figura 2.37: Se muestra el procedimiento de la segmentación mediante la técnica watershed. a) Imagen en escala de grises. b) Imagen considerada como un relieve topográfico. c) Resultado de la segmentación final.

Típicamente, la *watershed* se suele aplicar utilizando como imagen de entrada la imagen gradiente, ya que así las fronteras de las regiones son máximos locales. Uno de los problemas del gradiente si la imagen tiene mucho ruido se crean demasiados mínimos y máximos locales que se reflejan en una sobresegmentación. Si esto ocurre, aparece una gran cantidad de mínimos locales, por lo que se realiza una segmentación excesiva como la que se observa en la siguiente figura:

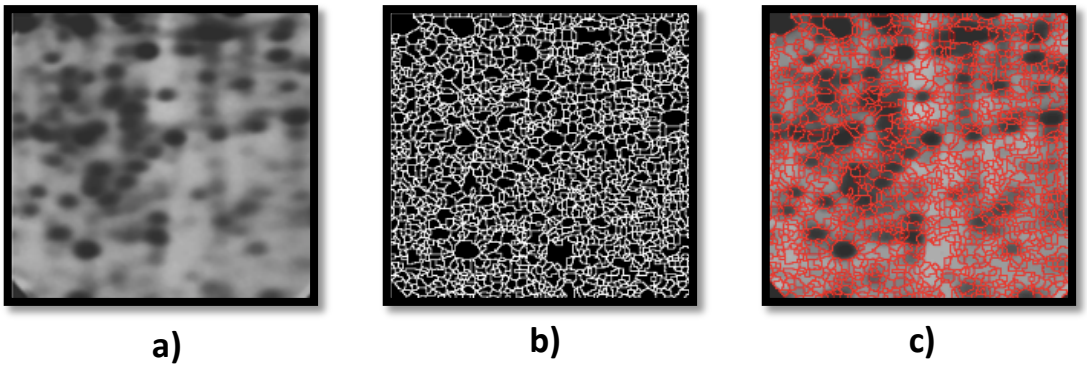


Figura 2.38: a) Imagen original. b) Imagen que muestra la sobresegmentación tras aplicar la técnica watershed. c) Imagen que visualiza el resultado de la segmentación sobre la imagen original.

Para solventar los problemas de la sobresegmentación surge la idea de transformar las imágenes para tener un único mínimo local por objeto. Esto se puede conseguir mediante diferentes métodos, por ejemplo utilizando gradientes filtrados, operaciones con distancias o a través de la técnica conocida como "*watershed* con marcadores", en la que se entrará con más detalle en el siguiente apartado.

2.3.5.2 Watershed con marcadores

En primer lugar se define un marcador como un conjunto de píxeles conectados pertenecientes a una imagen. Existen dos tipos: (i) marcadores internos y (ii) marcadores externos. Los primeros hacen referencia al objeto de interés, pues como su propio nombre indica consiste en colocarlos en el interior de dicho objeto. Los segundos se colocan fuera de la región de interés, normalmente en el fondo, y limitan la zona que debe ser segmentada [28].

Este método se basa en simular la inundación característica del valle a partir del marcador interno normalmente localizado como mínimo local en el objeto de interés. Por otra parte, se utiliza el marcador externo para fijar la frontera en el máximo local (la cima) y evitar que el agua con la que se inunda el valle se extienda fuera de él. Es decir, el marcador externo limita la segmentación que comienza con el marcador interno. Este procedimiento se realiza mediante la imposición de los marcadores como mínimos de la imagen. En la siguiente figura se puede observar cómo mejora el resultado de la segmentación utilizando este método, en lugar del llevado a cabo en la figura 2.38.

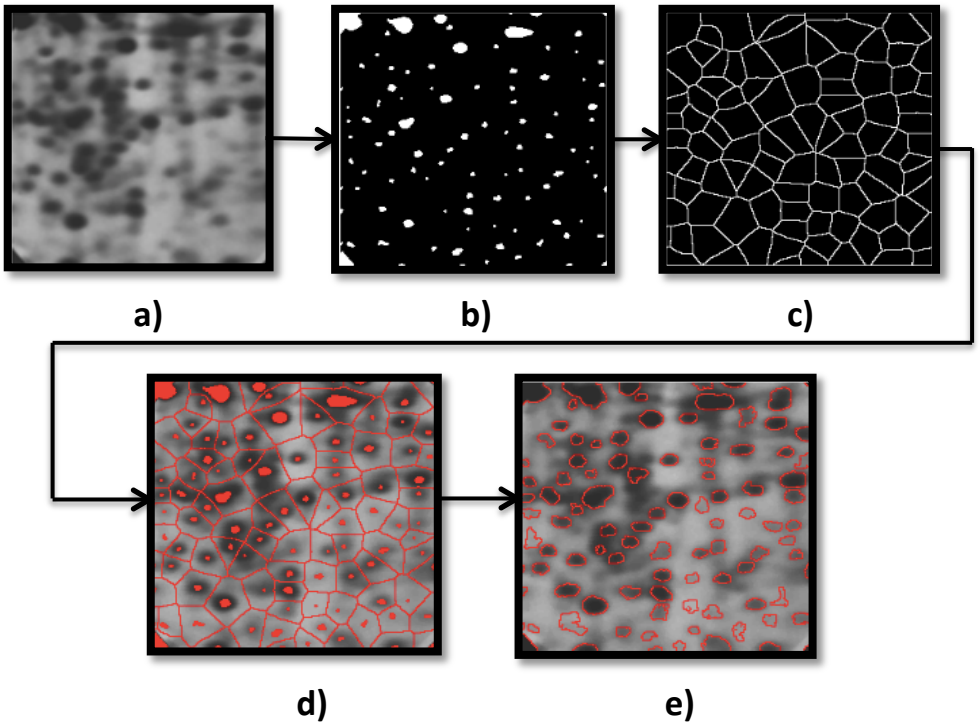


Figura 2.39: Ilustración donde se muestra con un ejemplo el proceso de la segmentación con marcadores. A) Imagen original. B) Imagen correspondiente a los marcadores internos. C) Imagen que hace referencia a las fronteras de la segmentación. D) Combinación de marcadores sobre la imagen original. e) Resultado de la segmentación.

2.3.5.3 Constrained Watershed

Este método consiste en una variante del *watershed* con marcadores, ya que el anterior solo funciona cuando las regiones de interés contienen fronteras cerradas. En el caso de la segmentación de glándulas de próstata esto no ocurre, pues la frontera debe venir marcada por los núcleos que rodean al elemento considerado como glándula y cada uno de esos núcleos está formado por un conjunto de píxeles conectados entre sí, pero separados del conjunto de píxeles de los núcleos de alrededor. Por tanto, si se utiliza el símil de la inundación, se podría analizar este caso viendo la imagen como un relieve topográfico donde los máximos locales fueran montañas individuales separadas. De esta forma, la inundación no podría darse porque al intentar llenar la cuenca de agua, esta se escaparía por el espacio que dejan las montañas. En la siguiente figura se muestra una comparación entre lo que sería una región delimitada por una frontera cerrada y por una frontera abierta.

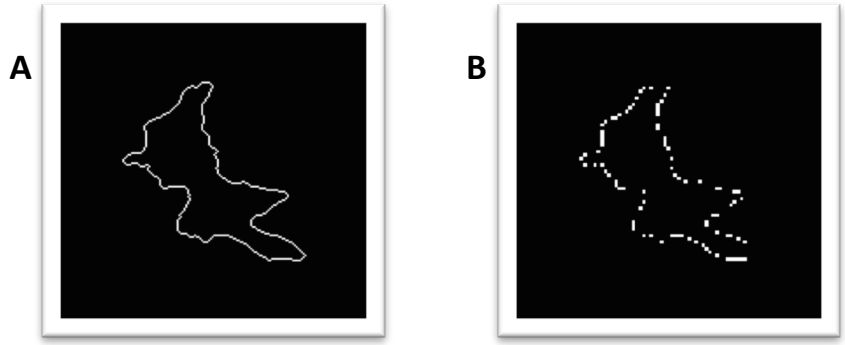


Figura 2.40: A) Imagen donde aparece una frontera cerrada delimitando la región de interés. B) Misma imagen con una frontera abierta.

Se observa que para la imagen B de la figura anterior no funcionaría la técnica normal de *watershed* con marcadores, puesto que al imponer el mínimo en la región de interés para inundarla, el agua no quedaría retenida, sino que escaparía por los huecos que quedan en negro entre las fronteras. Para solucionar este problema surge la variante conocida como "*Locally Constrained Watershed Transform"* que consiste en una herramienta potente y flexible para la segmentación, a partir de morfología matemática. Este método se basa en la imposición de restricciones en la frontera modificando la definición de la trayectoria subyacente [29]. Para ello, se ha implementado una función llamada "*watershedConstrained*" (ver lista de funciones) cuyos *inputs* son (i) *imin,* (ii) *fg,* (iii) *bg*, (iv) *tamFg* y (v) tamBg, que se refieren respectivamente a la imagen de entrada, al marcador interno, al marcador externo, al tamaño del EE para el marcador interno y al tamaño del EE para el marcador externo. El *output* que se proporciona es *"imout"* cuyo resultado puede visualizarse en la figura 2.44.A. Por tanto, la idea de la función para solventar el problema de las fronteras abiertas consiste en definir, en primer lugar, tres tipos de elementos: 1. elementos que actúen como marcadores internos, 2. elementos que actúen como marcadores externos y 3. una imagen de entrada de la *watershed*, cuyos elementos actuarán como restricciones. Además, es necesario definir dos EE, uno para el marcador externo y otro para el interno. El objetivo básico es que los marcadores internos y externos avancen por la imagen y al final se encuentren, de tal forma que cuando eso ocurra, los píxeles que contacten con ambos marcadores serán los que se definan como límites de frontera.

Por tanto, el funcionamiento del método consiste en que los píxeles de cada marcador analizan su vecindario e intentan incorporar nuevos píxeles a su región. Así, para avanzar por la imagen y recaudar esos píxeles, se define un tamaño de elemento estructurante, de tal forma que cada vez que el marcador intente expandirse, se analizarán los píxeles de las regiones definidas por los elementos estructurantes en busca de alguna restricción, la cual viene determinada por las intensidades de la imagen de entrada. Si no hay restricciones, o las restricciones no impiden la expansión del marcador, entonces este avanza e incorpora esos píxeles del vecindario a su región. La expansión se realiza en función del tamaño del EE definido, ya que dependiendo de lo grande que sea ese tamaño se restringirá más o menos el paso. Es decir, cuanto mayor sea el tamaño del EE, más difícil será avanzar por la imagen, ya que es más probable que las intensidades de la imagen de entrada analizadas con el EE le impongan mayores restricciones.

Esto es lo que ocurriría en la figura que se muestra a continuación, donde el marcador externo (cuadrado rojo) intenta incorporar píxeles de su alrededor avanzando por la imagen (cuadrado negro) con un elemento estructurante de grandes dimensiones (línea roja). Al mismo tiempo, el marcador interno (cuadrado verde) intenta de la misma forma incorporar píxeles de su vecindario, pero avanzando por la imagen con un EE más pequeño (línea verde). El conflicto se resuelve cuando el marcador externo, al intentar avanzar, se encuentra con dos restricciones (círculos azules) cuya separación es más pequeña que el tamaño del EE. Por tanto, el marcador externo no podrá continuar avanzando con su EE porque las restricciones le impiden el paso. Sin embargo, el EE del marcador interno es de dimensiones más pequeñas y por lo tanto su paso no estará restringido y podrá pasar a través de las restricciones. No obstante, enseguida entrará en contacto con el marcador externo, y por tanto la frontera quedará definida muy cerca de los elementos que se han seleccionado como restricciones. En el caso real de la segmentación de glándulas de próstata, esos elementos serán los núcleos, ya que son los que constituyen la región que delimita las glándulas.

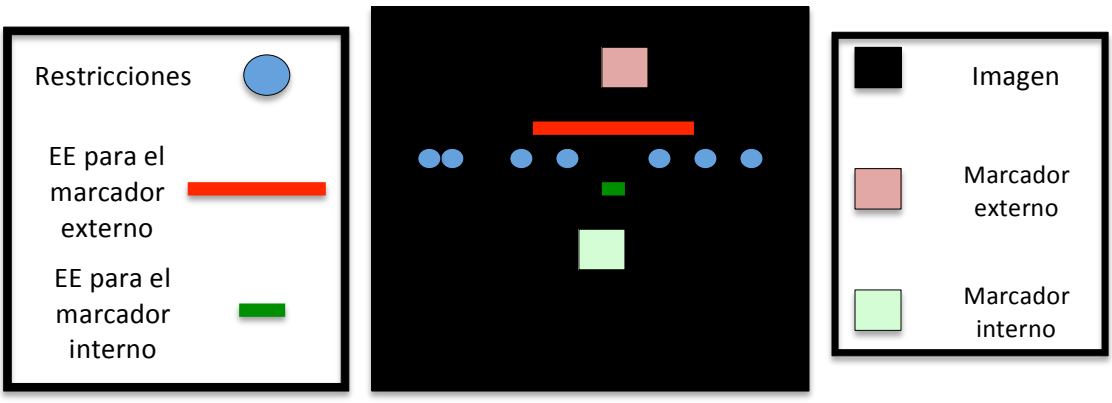


Figura 2.41: Ejemplo ilustrativo en el que se explica visualmente el funcionamiento de la técnica "Locally Constrained Watershed Transform".

Trasladando esta información esquemática al caso real del proyecto, se debe explicar qué objetos o imágenes van a funcionar como los elementos 1, 2 y 3 definidos anteriormente. Para ello, se implementa una función llamada "*Markers_watershed*" (ver lista de funciones) cuyos *inputs* son (i) *folder_img,* que es el directorio donde se encuentran las imágenes de test, y (ii) *image,* que es la imagen de test de la que se van a extraer todos los elementos descritos anteriormente.

De esta forma, con la función "*Markers_watershed*" se hace un procesado de las máscaras obtenidas anteriormente para conseguir los *outputs* (i) *mask_nuclei,* (ii) *mark_int* y (iii) *mark_ext* que se corresponden respectivamente con la imagen de entrada, los marcadores internos y los marcadores externos. Por tanto, en la función se definen:

(i) Como marcadores internos: los lúmenes de la máscara de lúmenes obtenida tras la predicción realizada en el proceso de clasificación supervisada.

(ii) Como marcadores externos: el fondo de la imagen obtenido mediante la función "*Mask_lumen"* y el centroide de las regiones del estroma que presentan una mayor cantidad de píxeles conectados. Este centroide se adquiere mediante el procesado que se implementa dentro de la propia función *"Markers_watershed"*. Cabe destacar en este punto que es importante definir marcadores externos en regiones del tejido correspondientes al estroma para que la función *"watershedConstrained"* proporcione buenos resultados.

(iii) Como imagen de entrada: los núcleos de la imagen correspondiente a la máscara de lúmenes extraída tras ejecutar la función "*Clustering".*

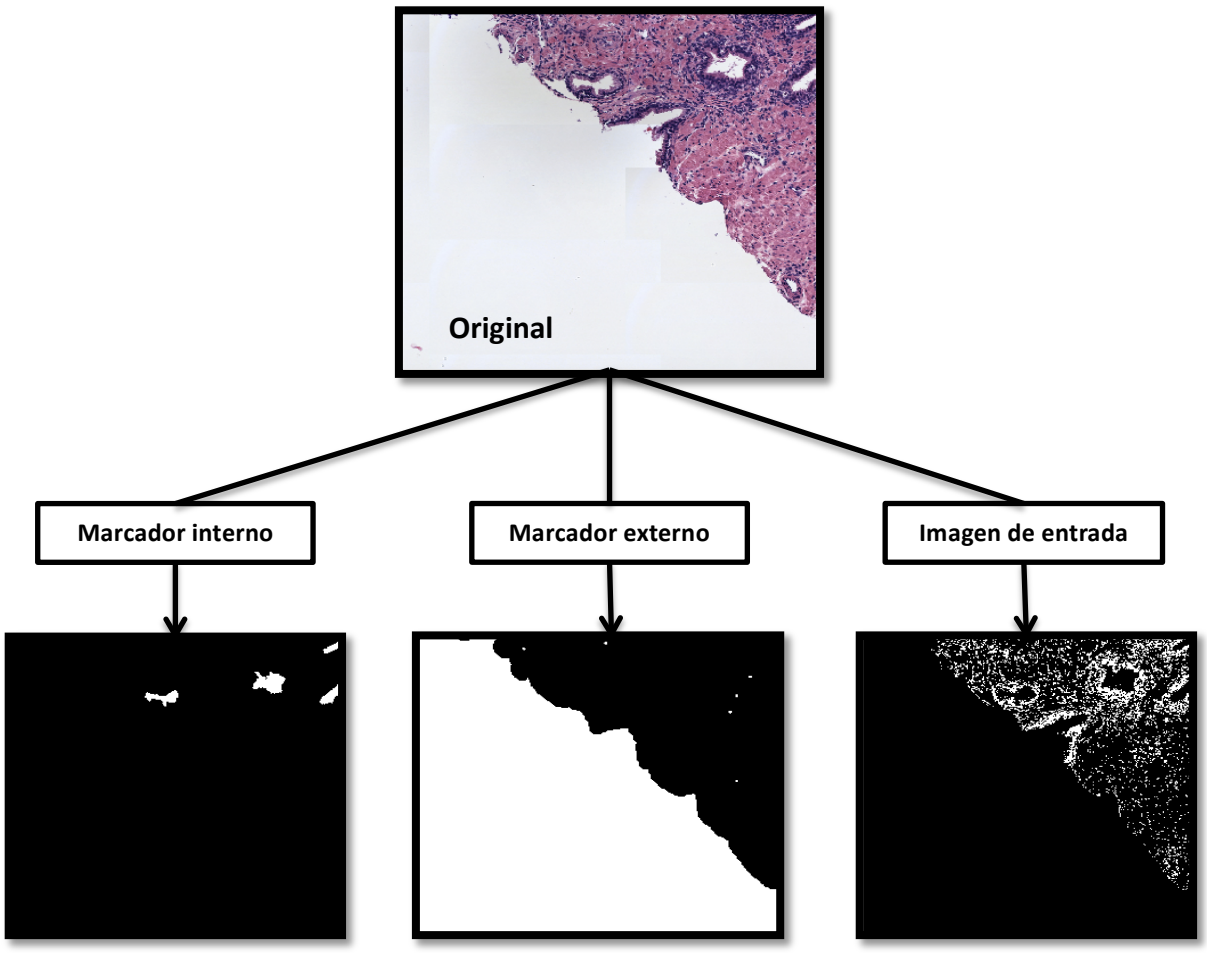


Figura 2.42: Modelo ilustrativo que muestra, a partir de la imagen original, las imágenes obtenidas como outputs de la función "Markers_watershed".

De esta forma, con la función "*Markers_watershed"* se obtienen las imágenes correspondientes a los elementos necesarios para utilizar como entradas de la función "*watershedCosntrained"*. Además, esta función necesitará también el *input* correspondiente al tamaño del elemento estructurante para cada marcador, el cual es determinante en el buen funcionamiento del algoritmo general. Para analizar el caso real, con el mismo ejemplo que en la representación esquemática de la figura 2.41, se muestra la siguiente figura para una imagen aleatoria del conjunto de *test.*

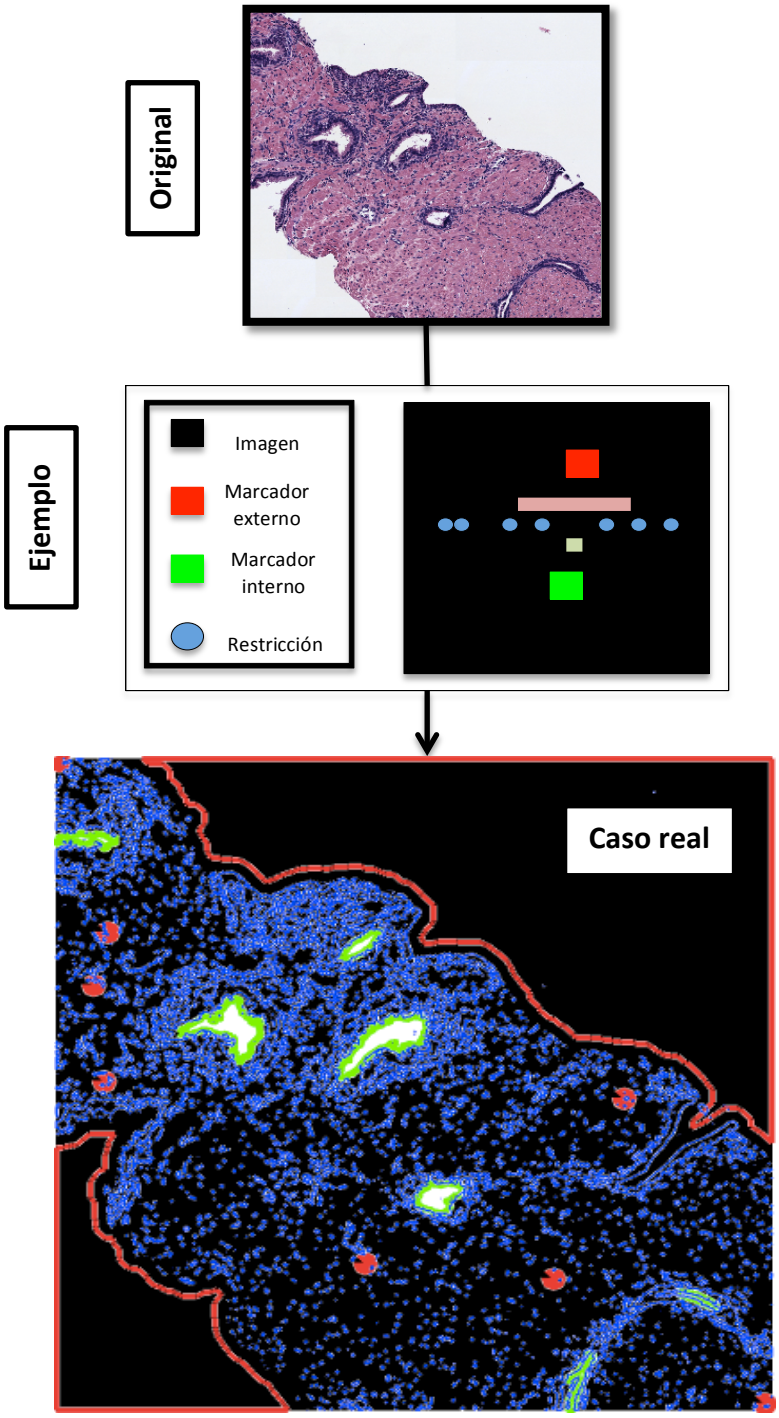


Figura 2.43: Modelo real donde se visualizan los bordes de los lúmenes como marcadores internos (en verde), los bordes del estrato de la muestra y algunas regiones con alto contenido de estroma como marcadores externos (en rojo) y los núcleos como imagen de entrada (en azul).

El siguiente y último paso para llevar a cabo la segmentación automática es implementar las funciones *"Markers_watershed"* y *"watershedConstrained"* en un mismo *script* llamado *"Segment_glands"* (ver lista de funciones) en el que se vuelve a hacer un re-escalado de las imágenes para mejorar el coste computacional. En la siguiente figura se muestra, para el mismo ejemplo de la figura 2.43, la imagen que se obtendría como resultado de la función "*watershedConstrained*" (ejemplo A). Por otra parte, se implementa también la función "*Mark_zones*" (ver lista de funciones) cuyos *inputs* son: (i) *img,* que es la imagen original en escala de grises y (ii) *limits,* que se corresponde con los bordes de la máscara obtenida a partir de la imagen A de la figura. Como *output* de esta función se obtiene *"immarc"* cuyo resultado se muestra en el ejemplo B.

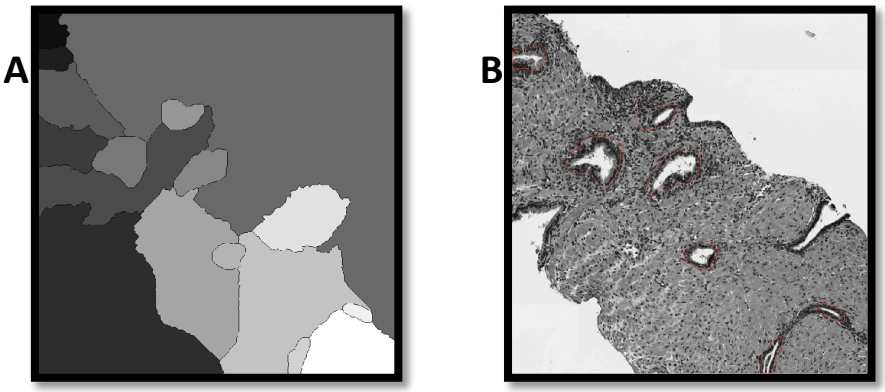


Figura 2.44: A) Imagen generada a partir de la función "watershedConstrained". B) Imagen generada a partir de la función "Mark_zones".

Por último, y para demostrar el cumplimiento del objetivo final de este proyecto, se muestra la segmentación final realizada para una imagen de test, a partir del clasificador SVM de umbral 0.9.

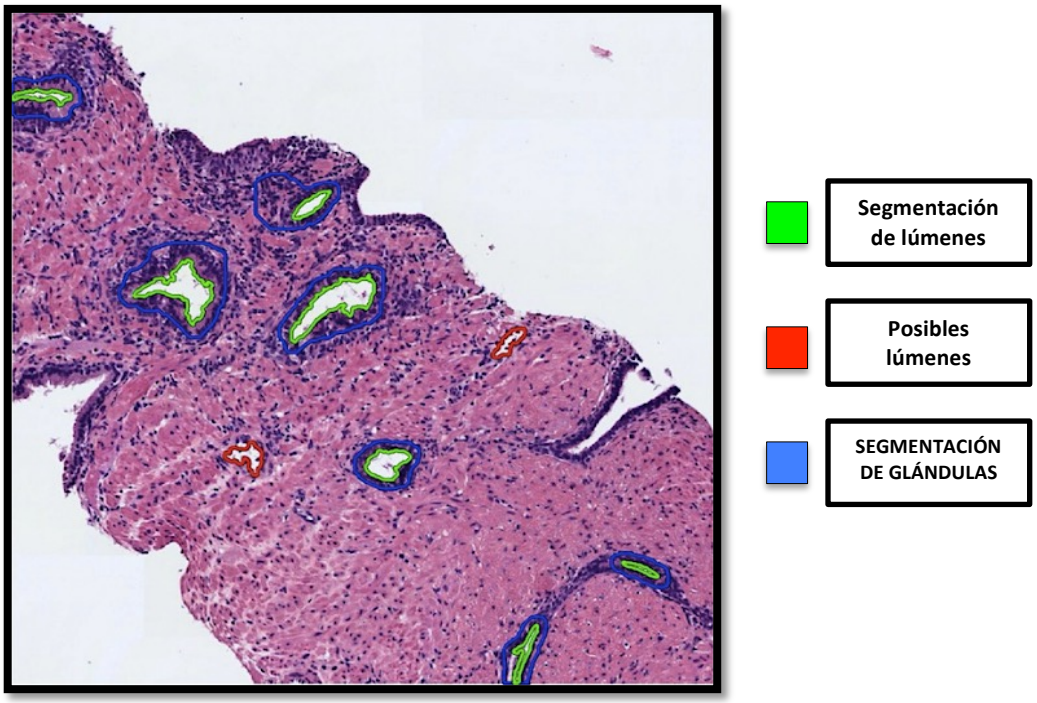


Figura 2.45: "Segmentación de estructuras glandulares en una imagen histológica de próstata sana".

2.3.6 Segmentación manual de glándulas

La segmentación manual es un proceso necesario para encarar la evaluación de los resultados obtenidos con el algoritmo de segmentación automática. Lo ideal sería que las imágenes originales fueran segmentadas manualmente por un patólogo especialista en tejidos histológicos, de tal manera que al comparar los resultados, estos fueran lo más correctos posible. En ese caso, si los resultados obtenidos al compararlos con los del *ground truth* fueran buenos, esto indicaría que se ha logrado el objetivo y, por tanto, se ha conseguido el diseño de una herramienta que los patólogos podrían emplear para mejorar su eficiencia y rendimiento a la hora de analizar las muestras de tejido histológico de próstata. No obstante, para este proyecto, la segmentación manual se ha llevado a cabo personalmente con el conocimiento adquirido en la carrera de Ingeniería Biomédica y con la ayuda de los expertos en patología que colaboran en el proyecto SICAP.

Para realizar la segmentación manual, en primer lugar, ha sido necesario implementar dos funciones llamadas "*Change_to_BMP*" y "*Create_mask"* (ver lista de funciones). La primera permite cambiar el formato de las imágenes de *.jpg a *.bmp, pues es necesario trabajar con este formato concreto para llevar a cabo la segmentación manual en la aplicación de la tablet. Esta aplicación fue desarrollada por los ingenieros del CVBLab como una de las fases del proyecto "Minerva" y sirve para facilitar la tarea de marcado manual de las fronteras de las glándulas. Por otra parte, la otra función "*Create_mask"* consiste en un código sencillo cuyos *inputs* son (i) *read_directory* y (ii) *save_directory* que son, respectivamente, el directorio del que se adquieren las imágenes en formato *.bmp y el directorio donde se van a guardar las máscaras que se generen. De esta forma, la función "*Create_mask"* permite comparar las máscaras generadas de la segmentación manual con las generadas de la segmentación automática y así obtener posteriormente los resultados correspondientes al coeficiente *Dice*, al coeficiente *Jaccard* y a sus desviaciones estándar (ver capítulo 3, apartado "segmentación").

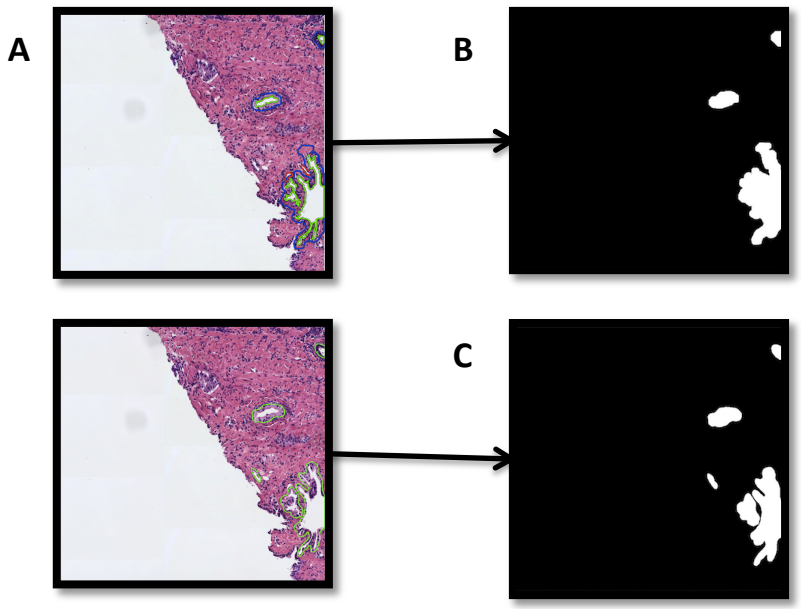


Figura 2.46: A) Imagen final segmentada con el algoritmo automático. B) Máscara extraída de la imagen A. C) Máscara de la segmentación manual que forma parte del ground truth.

CAPÍTULO 3

3. Resultados y discusión

Índice de contenidos

3.1 Clasificación 62

3.1.1 Resultados del mejor clasificador 62

3.1.2 Resultados de la mejor clasificación 64

3.2 Segmentación 67

3.2.1 Resultados cuantitativos 67

3.2.2 Visualización de resultados 69

3.1 Clasificación

3.1.1 Resultados del mejor clasificador

En el capítulo anterior se hace referencia al tipo de clasificador empleado para llevar a cabo este proyecto. Para seleccionar el mejor clasificador posible, como ya se ha comentado, se utiliza la técnica de *cross-validation* con las imágenes de *training* (que simulan la cantidad total de imágenes disponibles, pues la de test se utilizarán a modo de predicción). El resultado del *cross-validation* proporciona un valor de precisión (*accuracy)* para varios clasificadores entrenados a partir de las características extraídas (de cada posible lumen) que se le pasan como *inputs*. En total se dispone de 858 muestras etiquetadas con las que los diferentes clasificadores pueden ser entrenados. De esta forma, el clasificador que proporciona el mejor valor de precisión, utilizando la funcionalidad "*classification learner"* de la app de MATLAB, es el SVM lineal con un 86,1% de *accuracy.*

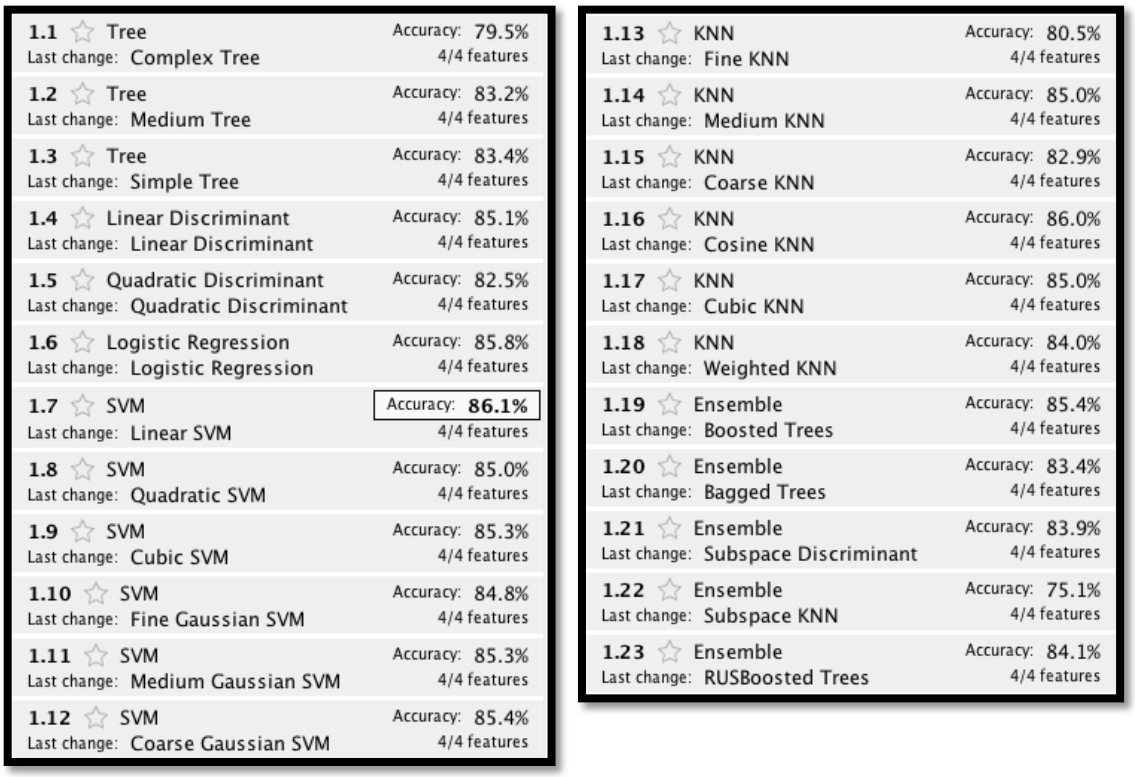


Figura 3.1: Lista de clasificadores con su valor correspondiente de precisión tras entrenar con las 858 muestras etiquetadas que se le pasan como input.

Se observa que se han entrenado 23 clasificadores distintos obteniendo finalmente como resultado que el mejor es el *Linear SVM.* Por tanto, para este clasificador se mostrarán a continuación los resultados obtenidos de la curva ROC[12] y la matriz de confusión.

[12] Curva ROC es una figura en la que el eje vertical viene representado por la sensibilidad y el eje horizontal por (1 – especificidad). Por tanto, el punto óptimo se encuentra en la esquina superior izquierda, ya que se intenta maximizar conjuntamente tanto la sensibilidad como la especificidad. Finalmente se proporciona el valor del área bajo la curva (*AUC*) cuyo resultado óptimo sería 1.

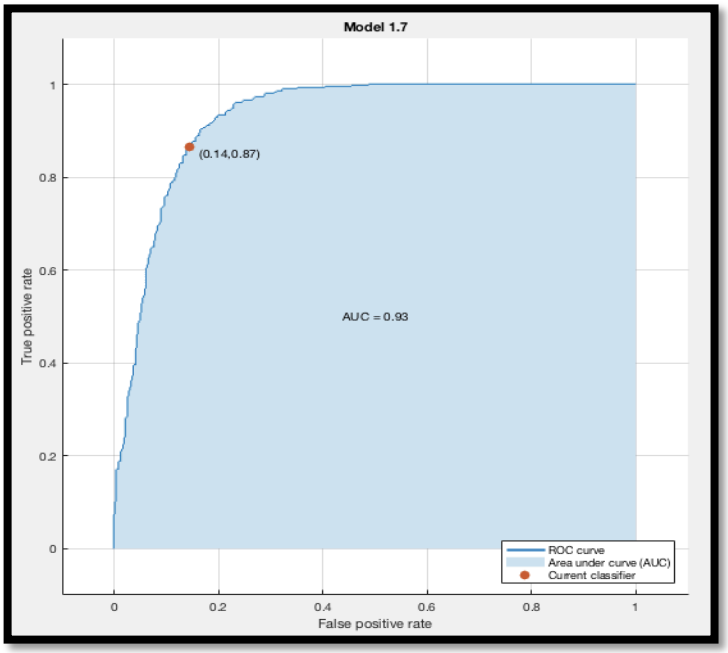


Figura 3.2: a) Curva ROC proporcionada como resultado del clasificador de la app de MATLAB al entrenar con las muestras de training.

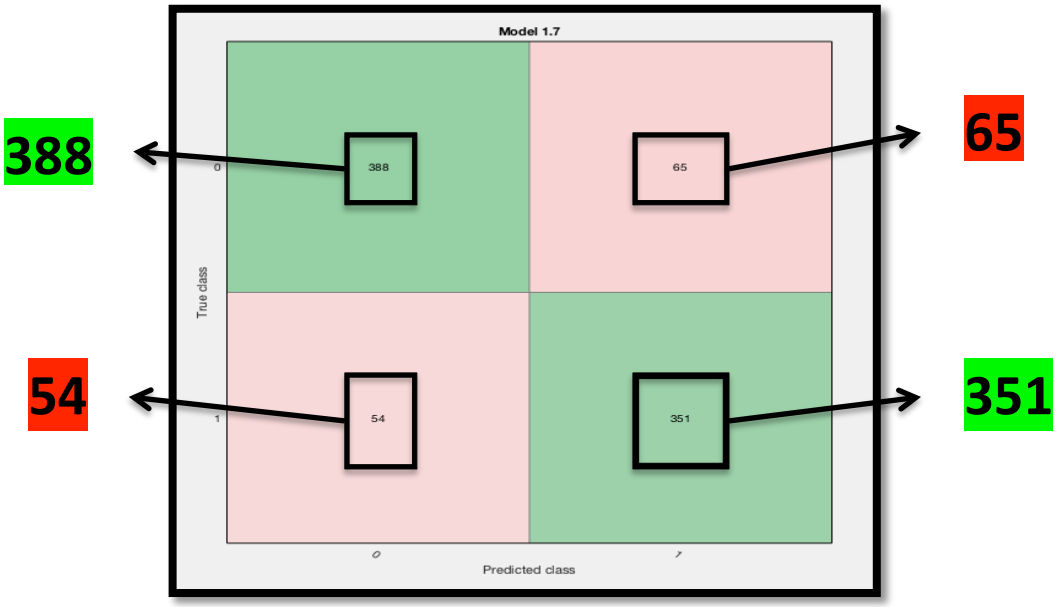


Figura 3.3: "Confusion Matrix" del SVM lineal, donde se muestran las 858 observaciones relacionándolas con el número de verdaderos positivos (388), falsos positivos (65), falsos negativos (54) y verdaderos negativos (351).

Los verdaderos positivos y los verdaderos negativos se corresponden con los aciertos del clasificador, de modo que se clasifican como "lumen" aquellos objetos que realmente lo son y como "no lumen", los que no lo son. En cambio, los falsos positivos y los falsos negativos hacen referencia a los fallos que comete el clasificador, de tal forma que los falsos positivos son aquellos objetos que se clasifican como lumen cuando realmente no lo son, y los falsos negativos los que se clasifican como "no lumen" cuando en realidad sí lo son.

3.1.2 Resultados de la clasificación

Para obtener los resultados referentes a la clasificación, lo que se ha hecho en este punto es implementar la función llamada "*Result_classification*" (ver lista de funciones) que utiliza como *inputs:* (i) *read_directory,* para referirse al directorio donde se encuentran las imágenes de test, y (ii) *folder_man_seg,* correspondiente al directorio donde se encuentran las máscaras creadas manualmente con la función "*Auto_segment"*, como se explica en el capítulo anterior en el apartado de generar el *ground truth* para la clasificación. Como *outputs* de esta función se obtienen los parámetros correspondientes a los verdaderos positivos, falsos positivos, falsos negativos y verdaderos negativos de cada imagen de test analizada con los dos umbrales establecidos para el clasificador SVM. De esta forma, se definen los *outputs "lumen_results"* y "*lumen_results2"* para el umbral 0.5 y 0.9, respectivamente. De la misma forma, "*indicators"* e "*indicators2"* se corresponden con los *outputs* que hacen referencia a los parámetros de la sensibilidad, especificidad, VPP, VPN, AI y F1Score obtenidos para el total de las imágenes de test con cada umbral. Por último, también se obtienen como *outputs* las matrices de confusión (como "*matrix_confusion*" y "*matrix_confusion2"*).

Por tanto, con la función "*Result_classification"* se obtiene la clasificación global de las imágenes de test. Es decir, lo que se hace es simular qué ocurriría si, en el futuro, es necesario utilizar este algoritmo automático para segmentar glándulas de nuevas imágenes. Para ello, se analizan las 23 imágenes de test y se predice cuáles serían los resultados de la clasificación de lúmenes. En base a ello, se puede obtener una tabla de contingencia con verdaderos positivos, falsos positivos, falsos negativos y verdaderos negativos como las que se muestran en las tablas 4 y 5. A partir de esos valores, también es posible obtener los parámetros relacionados con la sensibilidad, la especificidad y el resto de valores que se definen posteriormente para la tabla 6.

A continuación, se analiza el clasificador SVM lineal, obtenido de la app de MATLAB, para el umbral por defecto (0.5) y para el umbral modificado (0.9) con la intención de ganar en especificidad.

	Lumen real	**Lumen falso**	Total
Lumen +	VP = **109**	FP = 28	**Pt = 137**
Lumen -	FN = 2	VN = 61	**Nt = 63**
Total	**Vt = 111**	**Ft = 89**	**200**

Tabla 4: Clasificación con umbral en 0.5.

	Lumen real	**Lumen falso**	Total
Lumen +	VP = 105	FP = 11	**Pt = 116**
Lumen -	FN = 6	VN = 78	**Nt = 84**
Total	**Vt = 111**	**Ft = 89**	**200**

Tabla 5: Clasificación con umbral en 0.9.

Observando ambas tablas, se puede determinar el número de aciertos (en verde) y de fallos (en rojo) para cada umbral en la predicción de los objetos de las 23 imágenes de test a clasificar.

- SVM con umbral en 0.5. **Aciertos**: 170. **Fallos**: 30. **Porcentaje de éxito:** 85%.
- SVM con umbral en 0.9. **Aciertos**: 183. **Fallos:** 17. **Porcentaje de éxito:** 91,5%.

Claramente se puede afirmar que se observan mejores resultados al utilizar un umbral más alto para el clasificador SVM lineal, aunque el número de verdaderos positivos disminuye, como cabía esperar. Por tanto, fijando el umbral por encima del que viene por defecto se obtiene un mayor número de aciertos y un menor número de fallos para las mismas muestras analizadas.

Por otra parte, se muestran a continuación los parámetros que pueden calcularse a partir de las matrices de confusión presentadas en las tablas 4 y 5. Sus fórmulas matemáticas son:

$$Sensibilidad = \frac{VP}{VP + FN} \tag{3.1}$$

$$Especificidad = \frac{VN}{VN + FP} \tag{3.2}$$

$$Valor\ Predictivo\ Positivo\ (VPP) = \frac{VP}{VP + FP} \tag{3.3}$$

$$Valor\ Predictivo\ Negativo\ (VPN) = \frac{VN}{VN + FN} \tag{3.4}$$

$$Índice\ de\ precisión\ (AI) = \frac{VP + VN}{VP + VN + FP + FN} \tag{3.5}$$

$$F1\ Score = \frac{2 * VPP * Sensibilidad}{VPP + Sensibilidad} \tag{3.6}$$

Estos parámetros se evalúan atendiendo a una puntuación donde el mejor clasificador proporcionaría un valor de 1, lo cual se correspondería con el resultado óptimo. En la siguiente tabla se muestran los valores proporcionados por el clasificador SVM utilizando los dos umbrales para la obtención de los parámetros definidos en las ecuaciones.

	Sensibilidad	Especificidad	VPP	VPN	AI	F1Score
SVM con umbral de (0.5)	**0.982**	**0.685**	**0.796**	**0.968**	**0.850**	**0.879**
SVM con umbral de (0.9)	**0.946**	**0.876**	**0.905**	**0.929**	**0.915**	**0.925**

Tabla 6: Parámetros de estimación obtenidos para cada umbral fijado en el clasificador SVM.

Finalmente, se observa que utilizando como clasificador el SVM lineal con un umbral más alto, los resultados obtenidos son mejores, ya que aunque la sensibilidad y el VPN presentan valores más bajos (0.946 y 0.929, respectivamente) siguen siendo cercanos a 1. Además la diferencia es mucho menor que la obtenida para los otros cuatro parámetros, donde los valores son considerablemente más altos.

Cabe destacar que los parámetros de la especificidad y del VPN son tenidos en cuenta, pero realmente no son representativos en estudios de este tipo, ya que el número de objetos correspondientes al grupo "Lumen falso" (Ft = 89, tablas 4 y 5) puede variar en función del procesado que se haga para tener más o menos objetos detectados. Por otra parte, puesto que los verdaderos lúmenes siempre van a ser los mismos, al detectar más objetos se estarían detectando más lúmenes falsos. De esta forma, se obtendría un número más elevado de verdaderos negativos y, por tanto, la especificidad y el valor predictivo negativo aumentarían, pero no sería un modelo realmente representativo. En este proyecto, teniendo en cuenta esa "singularidad", se ha diseñado el modelo de modo que haya un número de lúmenes falsos similar al número de lúmenes auténticos, con la finalidad de que los parámetros sean representativos. Como se puede observar en la figura 3.3, el número de lúmenes reales es 442 y el de falsos 416, en total 858 objetos para las muestras de *training*. Del mismo modo, al predecir las de test, se observa en las tablas 4 y 5 que el número de lúmenes reales es 111 y el de falsos, 89. En total: 200 objetos para las muestras de *test.*

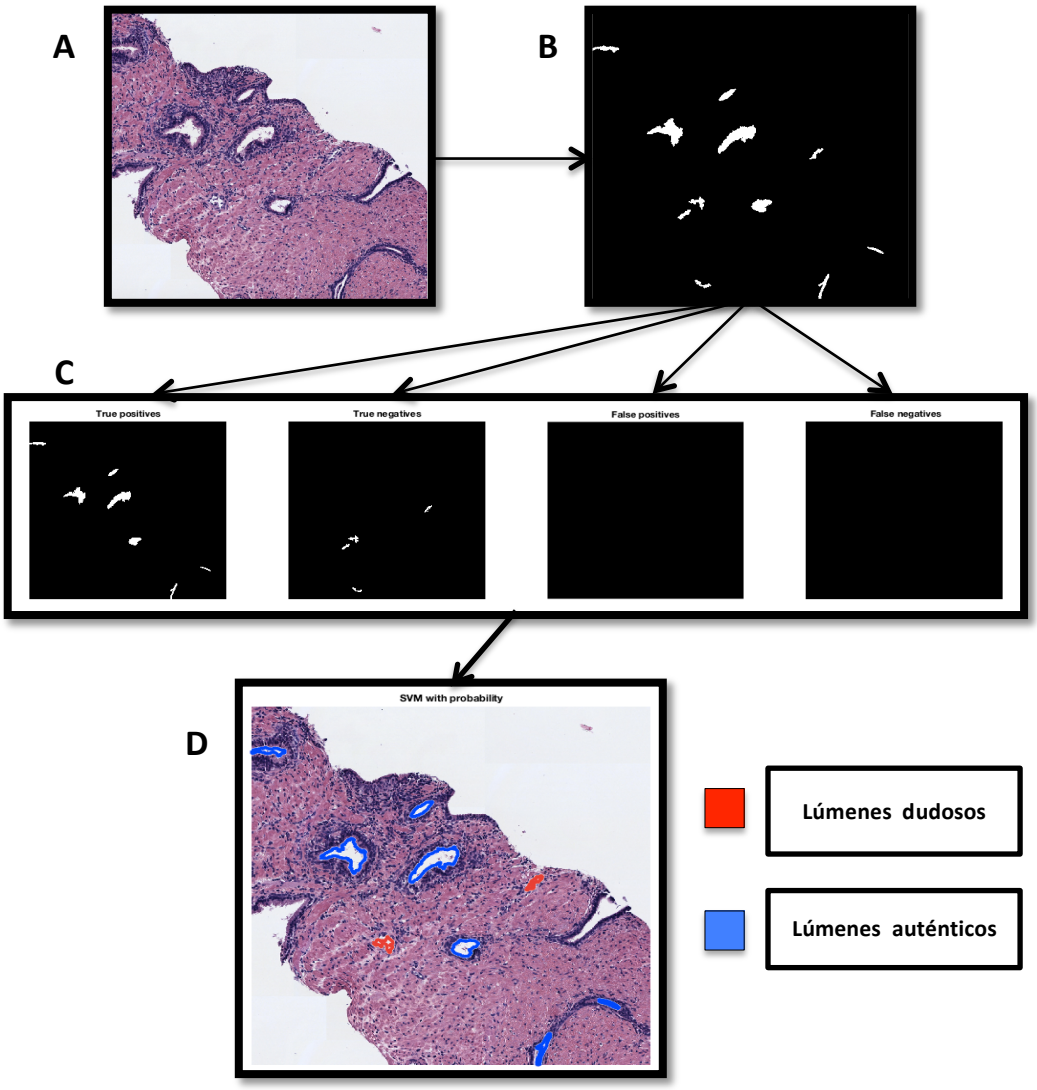


Figura 3.4: Proceso en el que se muestra cómo a partir de una imagen original (A), se extrae la máscara binaria (B) para clasificar los lúmenes, y en función de dicha clasificación (C), se representan los objetos sobre la imagen original para observar el resultado (D). A partir de los objetos azules se llevará a cabo la segmentación.

3.2 Segmentación

3.2.1 Resultados cuantitativos

Para llevar a cabo la evaluación de la segmentación se han implementado dos funciones llamadas "*Evaluate_coefficients"* y "*Result_segmentation"* (ver lista de funciones). En la primera se utilizan como *inputs* (i) *auto_mask*, que hace referencia a las máscaras correspondientes a la segmentación automática y (ii) *man_mask*, que hace referencia a las máscaras correspondientes a la segmentación manual. Como *output* se obtienen los parámetros con los que se va a evaluar el sistema diseñado: "*Dice*" y "*Jaccard*":

Coeficiente de *Dice:* es un estadístico que se utiliza para comparar la similitud entre dos regiones de la imagen y se expresa matemáticamente con una fórmula aplicable a datos de presencia/ausencia.

$$Dice(A,B) = \frac{2|A \cap B|}{|A| \cup |B|} \tag{3.7}$$

donde A y B son el número de regiones que hay en las muestras A y B, respectivamente, y Dice es el cociente de similitud que varía entre 0 y 1 [30].

Coeficiente de *Jaccard:* es un parámetro que mide el índice de similitud entre dos conjuntos, sea cual sea el tipo de elementos. La formulación matemática es la siguiente:

$$Jaccard(A,B) = \frac{|A \cap B|}{|A \cup B|} \tag{3.8}$$

donde $Jaccard$ representa la cardinalidad de la intersección de ambos conjuntos dividida entre la cardinalidad de su unión, y siempre toma valores entre 0 y 1 [31]. Ambos parámetros pueden relacionarse mediante las siguientes equivalencias:

$$Jaccard = \frac{Dice}{2-Dice} \quad ; \quad Dice = \frac{2*Jaccard}{1+Jaccard} \tag{3.9}$$

Así, con la función "*Evaluate_coefficients"* se obtienen ambos parámetros y se muestran las máscaras generadas de la segmentación automática y la manual. Véase un ejemplo en la figura 3.5.

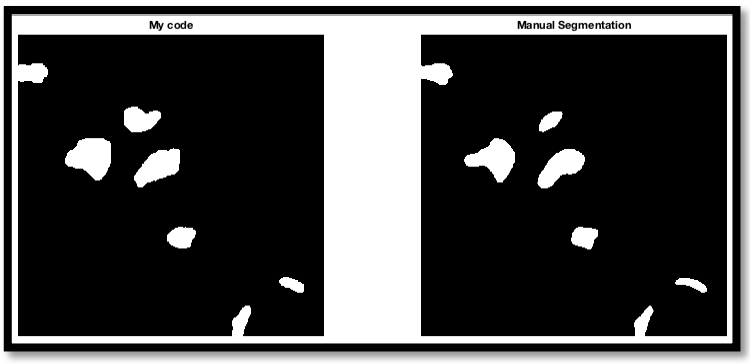


Figura 3.5: Ejemplo donde se muestran la máscara segmentada automáticamente y la segmentada manualmente. Los resultados para este caso concreto serían: Dice = 0.869 y Jaccard = 0.768.

Por otra parte, en la función "*Result_segmentation",* se adquiere el *output* "*Coefficients"* correspondiente a la media de los coeficientes de *Dice* y de *Jaccard.* Como *inputs* se utilizan (i) *folder_my_result*, correspondiente al directorio donde se encuentran las máscaras de la segmentación automática y (ii) *folder_manual_seg*, que se corresponde con el directorio donde están guardadas las imágenes de la segmentación manual. De esta forma, además de los coeficientes *Dice* y *Jaccard*, también se obtienen las medidas de sus respectivas desviaciones estándar.

	Coeficiente Dice	**Coeficiente Jaccard**	**Desviación estándar del coeficiente Dice**	**Desviación estándar del coeficiente Jaccard**
Resultados	0.875	0.782	0.061	0.089

Tabla 7: Resultados finales para los coeficientes Dice y Jaccard, y sus respectivas desviaciones estándar.

Finalmente,

$$\boldsymbol{Dice = 0.875 \pm 0.061}$$
$$\boldsymbol{Jaccard = 0.782 \pm 0.089}$$

En la siguiente figura se muestra el diagrama de *Box-Whisker* como otro parámetro estadístico para obtener resultados de características tales como la dispersión y la simetría.

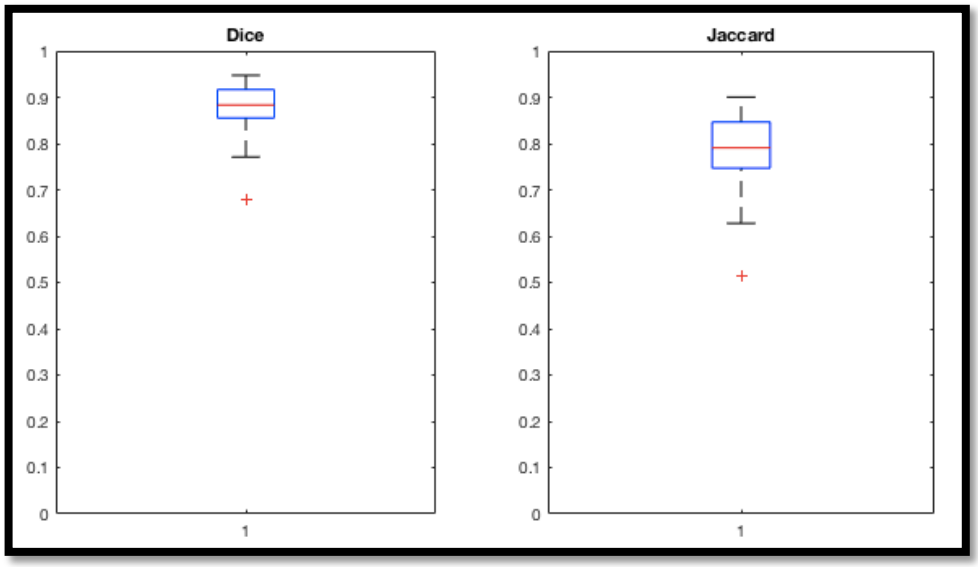


Figura 3.6: Diagrama de Box-Whisker para el coeficiente Dice y el coeficiente Jaccard, tras analizar las 200 muestras presentes en las 23 imágenes de test.

El diagrama de *Box-Whisker*, también llamado "Diagrama de Caja y Bigotes" consiste en un esquema visual donde se ordenan los datos y se obtienen sus valores mínimo y máximo, los cuartiles (Q1, Q2 y Q3) y el rango intercuartílico (RIC). El RIC hace referencia a la caja rectangular y sus dimensiones vienen determinadas por la expresión: RIC = Q3-Q1, siendo Q3 y Q1 el 75% y el 25% de los datos, respectivamente. En cuanto a los valores mínimo y máximo, estos vienen representados por los segmentos llamados "bigotes", los cuales pueden extenderse hasta un máximo de 1,5 veces

el RIC. Si hay datos que no se encuentran dentro de las dimensiones de los bigotes, se definen como valores atípicos (*outliers*). En cuanto a la línea roja que se visualiza dentro de la caja, esta se corresponde con el cuartil 2 (Q2) y coincide con el valor de la mediana.

En base a los resultados proporcionados con esta herramienta descriptiva se puede determinar que existe una cierta simetría en la distribución de los datos, pues la mediana está más o menos centrada en el rectángulo. Además, el rango intercuartilico en ambos parámetros es relativamente pequeño, pues abarca un intervalo de valores muy estrecho y solo hay un valor atípico para todas las muestras analizadas.

Por último, se puede afirmar que los valores medios obtenidos para los coeficientes Dice y Jaccard son considerablemente buenos teniendo en cuenta que cuanto más cerca esté su valor de 1, indicará que más se parece la segmentación automática a la manual. El valor óptimo, por tanto, sería 1, pero ese resultado es prácticamente inalcanzable, pues aunque la segmentación automática sea muy buena, la segmentación manual no deja de ser un método impreciso. De hecho, si un especialista llevase a cabo una segmentación manual de un conjunto de muestras y después volviera a repetir el proceso, al comparar las máscaras sería muy improbable que el resultado final de los coeficientes fuera igual a 1.

3.2.2 Visualización de resultados

A continuación, se muestran algunos ejemplos de la segmentación realizada para las imágenes de *test.* Se recuerda que estas imágenes se han utilizado para predecir el funcionamiento del sistema desarrollado ante posibles nuevas imágenes de tejido histopatológico de próstatas sanas.

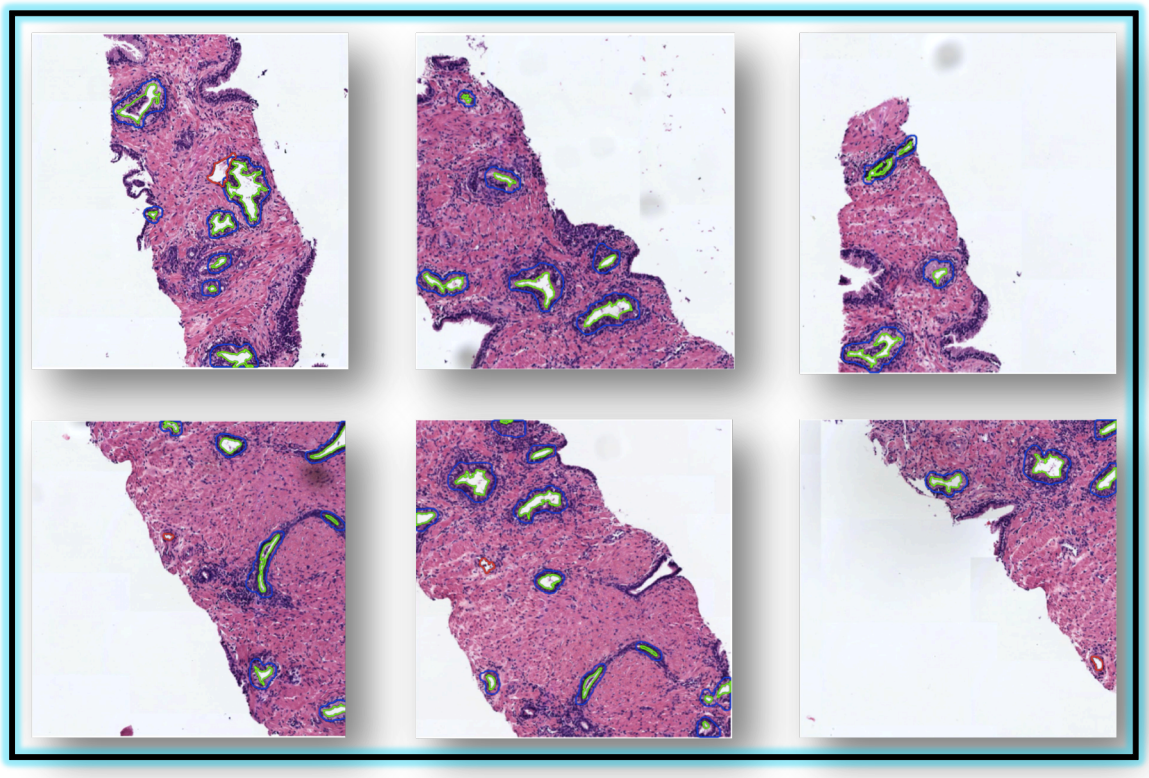


Figura 3.7: Ejemplos de la segmentación automática llevada a cabo para las 23 imágenes de test.

CAPÍTULO 4

4. Conclusiones y líneas futuras

Índice de contenidos

4.1 Conclusiones..... 72
4.2 Líneas futuras 74

4.1 Conclusiones

En primer lugar se ha conseguido implementar un algoritmo escalable y robusto que permite generar imágenes de alta resolución a partir de la concatenación de sub-imágenes en tantas filas y columnas como se especifique. Gracias a esto, es posible trabajar con diferentes muestras combinando los parámetros de resolución espacial, tamaño de imagen y número de glándulas presentes en una misma imagen.

Por otra parte, se ha realizado un *clustering* para abordar la clasificación de los tejidos presentes en una imagen de hematoxilina y eosina. Esta clasificación no supervisada se basa en la utilización del método *kmeans* para obtener las máscaras binarias de los cuatro componentes que aparecen en la imagen de próstata. Estos componentes son los núcleos, el estroma, el citoplasma y el lumen. La técnica empleada tiene como ventajas su fácil implementación y la rapidez en el cómputo, pero el principal inconveniente es que depende mucho de lo buena que sea la tinción. Esto es que, en ocasiones, los tejidos correspondientes a los diferentes elementos de la imagen pueden presentar un color similar conduciendo a una clasificación errónea en algunos píxeles. Por ejemplo, en algunas imágenes, la componente del estroma y del citoplasma se confunde con facilidad.

Posteriormente se realiza una clasificación supervisada en la que se lleva a cabo un etiquetado y una extracción de características de las imágenes con las que se entrenará el clasificador que posteriormente se utilizará para predecir las etiquetas de los objetos correspondientes a las imágenes de *test.* Para ello, en primer lugar se dividen las muestras disponibles en dos grupos: conjunto de *training* y conjunto de *test.* A continuación, para las imágenes de *training* se define de forma semiautomática un *ground truth,* a partir del cual se pueden etiquetar las muestras con 0s o 1s, en función de si los elementos analizados corresponden a objetos que son lúmenes auténticos o a falsos positivos. Además, empleando el mismo método para el conjunto de *test*, se consigue un *ground truth* con el que evaluar los resultados de la clasificación.

Con respecto a la extracción de características, para cada objeto detectado en las imágenes de *training* se analizan cuatro atributos con la finalidad de discriminar dicho objeto en una clase u otra. Después, se crea un fichero en el que a cada elemento se le asignan los valores de sus cuatro características analizadas y también la etiqueta que indica si es lumen o no. Con esta información, se hace uso de la app de "*classification learner*" de MATLAB para entrenar varios clasificadores con la técnica de *cross-validation* y 5 *folds*, resultando el mejor de ellos el SVM lineal. Una vez se dispone del mejor clasificador, se modifica el umbral por defecto para estudiar cómo varía la especificidad en función de ello. Por otra parte, se lleva a cabo el mismo procedimiento de extracción de características para las imágenes del conjunto de *test,* de forma que se predicen las etiquetas de sus objetos utilizando el clasificador SVM lineal con los dos umbrales definidos. Posteriormente, utilizando el *ground truth* creado para las imágenes de *test* se obtienen los resultados proporcionados por el clasificador tanto para el umbral que venía establecido por defecto como para el utilizado con la intención de ganar en especificidad.

En cuanto a la segmentación realizada, la técnica que se ha implementado es una variante del método "*watershed* con marcadores" que permite partir de la misma idea, pero pudiendo aplicarse en el caso de objetos cuyas fronteras no están cerradas. Para ello se definen los marcadores internos

y externos y sus respectivos elementos estructurantes, de forma que cada marcador intentará ir incorporando píxeles de la imagen analizando los que hay en su vecindario. La forma de avanzar por la imagen dependerá de la forma y del tamaño del EE definidos para cada marcador. Esta técnica utiliza también otra variable correspondiente a la imagen de entrada que vendrá definida por la imagen de núcleos. Los componentes de esta imagen actuarán como elementos de restricción al avance de los marcadores interno y externo con sus respectivos elementos estructurantes. De esta forma, partiendo de los objetos detectados como lúmenes auténticos por el clasificador, se inicia la segmentación definiendo: como marcador interno dichos lúmenes, como marcadores externos regiones con alto contenido de estroma y el estrato de la muestra (parte de la imagen donde no hay tejido), y como imagen de entrada la imagen de núcleos.

Finalmente, con respecto a la metodología, se lleva a cabo una segmentación manual de las imágenes de test que permitirán evaluar los resultados obtenidos de la segmentación automática. Para ello se emplea la app diseñada por los ingenieros del CVBLab en cooperación con los patólogos para facilitar la tarea de marcar fronteras de las glándulas presentes en imágenes histopatológicas como las que se trabajan en este proyecto. En este punto es destacable que se podría haber abordado el etiquetado de lúmenes a partir de las máscaras obtenidas de la segmentación manual. Para ello hubiera sido necesario segmentar a mano no solo las imágenes de *test*, sino también las de *training*, pudiendo con ello prescindir de la detección semiautomática, lo cual hubiese mejorado el rendimiento.

En lo referente a los resultados obtenidos, en primer lugar se evalúa la lista de clasificadores entrenados para ver cuál presenta un mayor porcentaje de precisión. De esta evaluación se concluye que el clasificador que mejores resultados proporciona es el SVM lineal, concretamente con un porcentaje de precisión de 86,1%. Se presentan también para este clasificador los resultados de la curva ROC y de la tabla de contingencia para abordar la evaluación de los parámetros de la sensibilidad y la especificidad. A continuación, utilizando ese clasificador se lleva a cabo la evaluación de la clasificación de las imágenes de test en términos de sensibilidad, especificidad, VPP, VPN, índice de precisión (AI) y F1Score para los dos umbrales fijados. Finalmente se obtienen mejores resultados utilizando el clasificador SVM con un umbral más alto, pues aunque los valores de sensibilidad y del VPN son peores, alcanza valores más altos para el resto de parámetros.

Por último, para llevar a cabo la evaluación de la segmentación se comparan las máscaras binarias de la segmentación automática con las máscaras generadas tras segmentar manualmente las imágenes con la app “Minerva”. Finalmente se obtienen los resultados para los coeficientes *Dice* y *Jaccard*, así como para sus respectivas desviaciones estándar. También se visualiza un diagrama de Caja y Bigotes para ambos coeficientes que permite estudiar otras magnitudes estadísticas relacionadas con la simetría y la dispersión.

En resumen, se puede concluir que se ha conseguido diseñar y desarrollar un sistema de segmentación de estructuras glandulares en imágenes histológicas de próstata sana, cuyos resultados han sido evaluados proporcionando valores considerablemente buenos, que pueden ser optimizados en un futuro, partiendo de la base de que este TFG corresponde a la primera etapa del proyecto nacional SICAP que dio comienzo en enero de 2017.

4.2 Líneas futuras

Como ya se ha comentado, este trabajo se corresponde con una de las primeras fases que forma parte de un proyecto nacional de mayor envergadura. Por tanto, la principal línea de futuro a contemplar se correspondería con los objetivos finales del proyecto SICAP. El paso inmediato tras este TFG, es la implementación de un procedimiento similar al que se ha llevado a cabo, pero trabajando con las imágenes de próstata que tienen cáncer, en lugar de con las sanas. La idea sería comenzar con aquellas cuyos grados de cáncer son más bajos en la escala *Gleason* (pues tienen un aspecto más parecido a las glándulas sanas) y, posteriormente, encarar el problema con las imágenes que presentan un cáncer de próstata en estadios avanzados (grados 4 y 5).

Basándose exclusivamente en lo que respecta a este proyecto de fin de carrera, se han ido encontrando posibles mejoras a partir de inconvenientes observados a lo largo de la implementación de los métodos. En primer lugar, se podría optimizar la parte del pre-procesado utilizando por ejemplo una deconvolución de color en el espacio LAB que permitiese solventar los problemas relacionados con la tinción de las imágenes. Es decir, se podría llevar a cabo un procesado de la imagen tal que, al realizar el *clustering* para la clasificación de tejidos, no se confundiesen los colores que representan los distintos componentes de la imagen (núcleos, estroma, citoplasma y lumen).

Por otra parte, en el ámbito de la clasificación supervisada de lúmenes, se podría llevar a cabo una optimización en el proceso de extracción de características, de manera que se profundizara en la selección de más y mejores. Esta selección se debería realizar en términos de proporcionar unos valores más discriminatorios a la hora de clasificar los objetos en la clases: “lumen” o “no lumen”. De este modo, se podría implementar un método de optimización en el que a partir de una gráfica resultase posible determinar cuál es la mejor combinación de características para entrenar los clasificadores.

De cara al futuro, también será posible entrenar y testear con más imágenes, pues a día de hoy se está esperando la llegada de nuevas imágenes por parte de los patólogos para llevar a cabo los algoritmos de automatización, tanto en imágenes de próstata sana como de próstata con cáncer. Así, se podría evaluar nuestro sistema de clasificación y segmentación con las nuevas imágenes, o bien ampliar el conjunto de imágenes de *training* y el conjunto de imágenes de *test* con el objetivo de mejorar el algoritmo.

En fases venideras, sería interesante contar con la ayuda de los patólogos que colaboran en el proyecto SICAP para determinar a qué clase corresponden los lúmenes clasificados como “dudosos”. De esta forma, cuando se haya conseguido realizar el mismo procedimiento llevado a cabo en las glándulas sanas, pero en las glándulas enfermas, se podrán comparar las segmentaciones realizadas para todos los casos y determinar qué factores son determinantes a la hora de distinguir las glándulas sanas y las glándulas cancerígenas. Esto contribuye al objetivo final que consiste en generar un sistema automático de ayuda al diagnóstico precoz del cáncer de próstata.

Por último, como posible línea de futuro se podría implementar un algoritmo que permitiese detectar y segmentar aquellas glándulas que aparecen “cortadas” en contacto con el estrato de la muestra (fondo de la imagen). Este inconveniente es, en muchas ocasiones, debido al proceso físico de preparación de las muestras al utilizar el micrótomo.

Bibliografía

[1] "OMS | Cáncer," *WHO*, 2017.

[2] M. Plummer, C. de Martel, J. Vignat, J. Ferlay, F. Bray, and S. Franceschi, "Global burden of cancers attributable to infections in 2012: a synthetic analysis," *Lancet Glob. Heal.*, vol. 4, no. 9, pp. e609–e616, 2016.

[3] "Tipos comunes de cáncer - National Cancer Institute." [Online]. Available: https://www.cancer.gov/espanol/tipos/comunes. [Accessed: 23-Jun-2017].

[4] "Factores de riesgo: Edad - National Cancer Institute." [Online]. Available: https://www.cancer.gov/espanol/cancer/causas-prevencion/riesgo/edad. [Accessed: 23-Jun-2017].

[5] "Cáncer de próstata: Estadísticas | Cancer.Net." [Online]. Available: http://www.cancer.net/es/tipos-de-cáncer/cáncer-de-próstata/estadísticas. [Accessed: 23-Jun-2017].

[6] "Cáncer de próstata: MedlinePlus enciclopedia médica." [Online]. Available: https://medlineplus.gov/spanish/ency/article/000380.htm. [Accessed: 20-Jun-2017].

[7] "Tratamiento del cáncer de próstata." [Online]. Available: https://www.cancer.org/es/cancer/cancer-de-prostata/tratamiento.html. [Accessed: 20-Jun-2017].

[8] "Siemens-Magnetom-Aera.jpg (1502×1179)." [Online]. Available: http://www.elizechesac.com/wp-content/uploads/2014/04/Siemens-Magnetom-Aera.jpg. [Accessed: 20-Jun-2017].

[9] "¿Qué es un MRI (Imagen por Resonancia Magnética)?" [Online]. Available: http://www.sanjuanmri.net/que-es-un-mri-imagen-por-resonancia-magnetica/. [Accessed: 20-Jun-2017].

[10] LABEQUIM, "MICROTOMOS."

[11] G. Rolls, "An Introduction to Specimen Preparation." 29-Nov-2011.

[12] "Histology sample preparation system / tissue / staining / bench-top - Artisan™ Link Pro - Dako." [Online]. Available: http://www.medicalexpo.com/prod/dako/product-80232-535315.html. [Accessed: 20-Jun-2017].

[13] "El Microscopio – RLbuhos – Medium." [Online]. Available: https://medium.com/@Rogerloaeza/el-microscopio-875d288a6c97. [Accessed: 20-Jun-2017].

[14] K. Nguyen, B. Sabata, and A. K. Jain, "Prostate cancer grading: Gland segmentation and structural features," in *Pattern Recognition Letters*, 2012, vol. 33, no. 7, pp. 951–961.

[15] J. Gordetsky and J. Epstein, "Grading of prostatic adenocarcinoma: current state and prognostic implications.," *Diagn. Pathol.*, vol. 11, p. 25, Mar. 2016.

[16] P. M. Pierorazio, P. C. Walsh, A. W. Partin, and J. I. Epstein, "Prognostic Gleason grade grouping: data based on the modified Gleason scoring system.," *BJU Int.*, vol. 111, no. 5, pp. 753–60, May 2013.

[17] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[18] "OpenSlide." [Online]. Available: http://openslide.org/. [Accessed: 21-Jun-2017].

[19] "R2017a - Actualizaciones de las familias de productos MATLAB y Simulink - MATLAB & Simulink." [Online]. Available: https://es.mathworks.com/products/new_products/latest_features.html. [Accessed: 24-Jun-2017].

[20] Naranjo Ornedo, V. (2016). Apuntes de Imágenes Biomédicas: Segmentación de imágenes. *Universitat Politècnica de València.*

[21] S. Dasgupta and Y. Freund, "Random projection trees for vector quantization," May 2008.

[22] "Clasificación supervisada y no supervisada | Advanced Tech Computing Group UTPL." [Online]. Available: https://advancedtech.wordpress.com/2008/04/14/clasificacion-supervisada-y-no-supervisada/. [Accessed: 26-Jun-2017].

[23] Manjón Herrera, J.V. (2016). Apuntes de Imágenes Biomédicas: Análisis y reconocimiento de patrones. *Universitat Politècnica de València.*

[24] R. M. Haralick, S. R. Sternberg, and X. Zhuang, "Image Analysis Using Mathematical Morphology," *Pattern Anal. Mach. Intell. IEEE Trans.*, vol. 9, no. 4, pp. 532–550, 1987.

[25] M. Cárdenas-Montes, "Sobreajuste - Overfitting."

[26] R. Kohavi, "A Study of Cross-Validation and Bootstrap for Accuracy Estimation and Model Selection," 1995.

[27] Fuster-García, E. (2017). Apuntes de Sistemas de información y telemedicina II. Fundamentos de los sistemas de ayuda a la decisión basados en datos biomédicos: Métodos de apredizaje. *Universitat Politècnica de València.*

[28] R. Gonzalez and R. Woods, *Digital image processing*. 2002.

[29] R. Beare, "A locally constrained watershed transform," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 7, pp. 1063–1074, 2006.

[30] "F-scores, Dice, and Jaccard set similarity | AI and Social Science – Brendan O'Connor." [Online]. Available: https://brenocon.com/blog/2012/04/f-scores-dice-and-jaccard-set-similarity/. [Accessed: 02-Jul-2017].

[31] "Índice Jaccard - Wikipedia, la enciclopedia libre." [Online]. Available: https://es.wikipedia.org/wiki/Índice_Jaccard. [Accessed: 02-Jul-2017].

II. PRESUPUESTO

ÍNDICE DEL PRESUPUESTO

1. PRESUPUESTOS PARCIALES 5
   1.1 COSTE DE PERSONAL 5
   1.2 COSTE DE LOS MATERIALES 5
   1.3 COSTE DE LAS HERRAMIENTAS HARDWARE Y SOFTWARE 6
2. PRESUPUESTO TOTAL 7

1. PRESUPUESTOS PARCIALES

En este apartado del informe se detallan las diferentes partes que conforman los presupuestos parciales de este proyecto basado en el diseño y el desarrollo de un sistema de segmentación de estructuras glandulares en imágenes histológicas de próstata. Para cuantificar estos presupuestos parciales se han desglosado los costes en tres grupos claramente diferenciados: (i) coste de personal, (ii) coste de los materiales y (iii) coste de las herramientas hardware y software, utilizadas para llevar a cabo el desarrollo de este proyecto.

1.1 COSTE DE PERSONAL

En este apartado se tienen en cuenta los recursos humanos necesarios para el desarrollo del TFG. De esta manera, se detalla la remuneración de cada participante atendiendo a una estimación de los costes que viene determinada por el tiempo dedicado al proyecto, en unidades de horas. Por tanto, debe considerarse la contribución al trabajo de: Dª Valery Naranjo Ornedo (como tutora del proyecto), D. Francisco José Peñaranda Gómez (como cotutor) y D. José Gabriel García Pardo (como alumno y autor del proyecto).

Tabla 1: Presupuesto para el coste de personal.

Denominación	Uds.	Cantidad	Precio unitario (€)	Total (€)
Directora del TFG (Profesora)	h	30	29,50	885,00
Cotutor del TFG (Doctorando)	h	30	17,20	516,00
Autor del TFG (Estudiante de GIB)	h	300	12,50	3.750,00
			TOTAL	5.151,00

1.2 COSTE DE LOS MATERIALES

Para llevar a cabo el presupuesto de este apartado se ha tenido en cuenta el coste de la realización de una prueba de biopsia, a partir de la cual se obtienen las imágenes con las que se trabaja en este proyecto y, por otra parte, el coste que supone la adquisición de cada muestra correspondiente al *slide* donde se presentan las imágenes teñidas con hematoxilina y eosina.

Tabla 2: Presupuesto para el coste de los materiales.

Denominación	Uds.	Cantidad	Precio unitario (€)	Total (€)
Biopsia	u	1	600,00	600,00
Muestras	u	3	10,00	30,00
			TOTAL	630,00

1.3 COSTE DE LAS HERRAMIENTAS HARDWARE Y SOFTWARE

En este apartado se detalla el presupuesto parcial correspondiente a los recursos hardware y software necesarios para llevar a cabo la totalidad del proyecto. Dado que estas herramientas no se han obtenido específicamente para la elaboración del trabajo, es necesario tener en cuenta el periodo de amortización para cada uno de ellos. Ese periodo se relaciona con la vida útil del material y el intervalo de tiempo amortizado para cada herramienta. Cabe destacar en este punto que toda la parte correspondiente al software empleado para la realización de este proyecto ha tenido un coste cero, ya que únicamente se han utilizado Microsoft Word y MATLAB 2017a®, cuyas licencias han sido proporcionadas gratuitamente por la Universitat Politècnica de València a sus estudiantes.

Tabla 3: Presupuesto de las herramientas hardware y software.

Denominación	Uds.	Cantidad	Precio unitario (€)	Periodo de amortización (años)	Intervalo amortizado (meses)	Total (€)
Licencia de Office 365	u	1	0,00	1	6	0,00
Licencia de MATLAB 2017®	u	1	0,00	1	6	0,00
Portátil Mackbook Air Intel Core i5	u	1	1.299,00	4	6	162,38
					TOTAL	162,38

2. PRESUPUESTO TOTAL

Para el presupuesto total del proyecto es necesario tener en cuenta la suma de los presupuestos parciales definidos en el capítulo anterior. Además, deben añadirse al cálculo dos porcentajes establecidos: el de gastos generales (13%) y el asociado al beneficio industrial (6%). Por último, es preceptivo repercutirle al resultado final el 21% de IVA, obteniendo de esta forma el presupuesto total que supondría la realización de este Trabajo Fin de Grado.

Tabla 4: Presupuesto total.

CAPÍTULOS	IMPORTE (€)
1. Coste de personal	5.151,00
2. Coste de los materiales	630,00
3. Coste de herramientas hardware y software	162,38
PRESUPUESTO DE EJECUCIÓN DE MATERIAL	**5.943,38**
13% de gastos generales	772,64
6% de beneficio general	356,60
SUMA	**7.072,62**
21% de IVA	1.485,25
PRESUPUESTO TOTAL	**8.557,87**