



UNIVERSIDAD DE LA REPÚBLICA
FACULTAD DE INGENIERÍA



Evaluación de la salud uterina en yeguas a partir de biopsias endometriales utilizando técnicas de procesamiento de imágenes y aprendizaje automático

TESIS PRESENTADA A LA FACULTAD DE INGENIERÍA DE LA
UNIVERSIDAD DE LA REPÚBLICA POR

Nicolas Aguilera, Agustina Diaz, Sofía Zimmer

EN CUMPLIMIENTO PARCIAL DE LOS REQUERIMIENTOS
PARA LA OBTENCIÓN DEL TÍTULO DE
GRADO EN INGENIERÍA ELÉCTRICA.

DIRECTOR DE TESIS

María José Estradé Universidad de la República
Federico Lecumberry Universidad de la República
Pablo Musé Universidad de la República

TRIBUNAL

Guillermo Carbajal Universidad de la República
Esley Torres Universidad de la República
José Manuel Verdes Universidad de la República

Montevideo
viernes 10 noviembre, 2023

Evaluación de la salud uterina en yeguas a partir de biopsias endometriales utilizando técnicas de procesamiento de imágenes y aprendizaje automático, Nicolas Aguilera, Agustina Diaz, Sofía Zimmer.

ISSN 1688-2806

Esta tesis fue preparada en L^AT_EX usando la clase iietesis (v1.1).

Contiene un total de 133 páginas.

Compilada el viernes 10 noviembre, 2023.

<http://iie.fing.edu.uy/>

Resumen

El presente trabajo propone investigar formas de incorporar técnicas de procesamiento de imágenes y modelos de aprendizaje automático en el estudio de biopsias endometriales de yeguas con tinción hematoxilina y eosina (H&E). Las muestras de tejido se encuentran en formato *Whole Slide Image* (WSI), que son de imágenes de muy alta resolución que pueden alcanzar fácilmente 7.2 millones de píxeles por canal y ocupar un tamaño de 1.2 GB.

La fertilidad animal es un tema ampliamente estudiado con el fin de promover la crianza de animales que hereden características de sus progenitores. Especialmente en los equinos, se tienen yeguas muy cotizadas, utilizadas para diferentes deportes ecuestres, cuyo material genético es muy valioso. Los equinos no se seleccionan por sus características reproductivas como otros animales de producción, sino por sus aptitudes deportivas o fenotípicas.

La salud uterina es fundamental para una gestación saludable, por lo que es de suma importancia conocer el estado del útero de la yegua. Para esto, entre otras técnicas, se realizan las biopsias endometriales. En estas biopsias los patólogos pueden estudiar la presencia y disposición de las diferentes estructuras en el endometrio, y estimar el potencial grado de fertilidad del animal.

Este proyecto implementa un flujo de trabajo secuencial que abarca las siguientes tareas: realizar inferencia con modelos de segmentación sobre toda una WSI, alinear y cortar la WSI y las máscaras generadas en parches, pos-procesar las mismas y extraer datos cuantitativos. De los modelos de aprendizaje automático aplicados, dos de ellos se tratan de modelos pre-entrenados sobre otros dominios (tejidos humanos), y segmentan glándulas y núcleos. El tercero se trata de un modelo de segmentación de fibrosis, que es una de las patologías principales que afecta la capacidad de las yeguas de llevar un embarazo a término. Este modelo fue entrenado con una base de datos generada durante el desarrollo de este proyecto. En cuanto a los datos cuantitativos a extraer, se explora sobre la densidad de glándulas y se busca clasificar los núcleos en dos clases: linfocito y no linfocito.

Los resultados iniciales se consideran exitosos. Se entrenó un modelo de segmentación de fibrosis aplicando técnicas de *transfer learning*, normalización de color y aumentado de datos. Se logró procesar de punta-a-punta una WSI completa, en un mismo entorno y en una única ejecución. Se obtuvieron mapas de calor identificando zonas de la muestra que presentan alta densidad de las estructuras biológicas segmentadas. Se calculan datos cuantitativos del área de las glándulas y características de los núcleos, por medio del flujo de trabajo implementado.

Esta página ha sido intencionalmente dejada en blanco.

Tabla de contenidos

Resumen	I
Resumen	I
1. Introducción	1
1.1. Objetivos del proyecto	1
1.2. Contexto del problema	2
1.2.1. Biopsias endometriales	2
1.2.2. Patologías y anomalías	5
1.3. Antecedentes	7
1.3.1. Análisis de imágenes en medicina para la identificación de patologías	8
1.3.2. Aplicación de técnicas de aprendizaje profundo en medicina	9
1.4. Estudio de herramientas	11
1.4.1. Modelos de segmentación	11
1.4.2. Fully Convolutional Network (FCN)	11
1.4.3. U-Net	12
1.4.4. SegNet	13
1.5. Frameworks	13
1.5.1. Quick Annotator	13
1.5.2. NoCodeSeg	15
1.6. Organización del documento	16
2. Imágenes y técnicas de preprocesamiento	19
2.1. Descripción de los datos	19
2.2. Normalización de color	22
2.2.1. Investigación de algoritmos	24
2.2.2. Pruebas realizadas	33
3. Segmentación de estructuras	41
3.1. Segmentación de glándulas	41
3.2. Segmentación de núcleos	45
3.3. Segmentación de fibrosis	48
3.3.1. Primer acercamiento	49
3.3.2. Problemas con Quick Annotator	54
3.3.3. Cambio de implementación: Pytorch-UNet	60

Tabla de contenidos

3.3.4. Modificaciones a F-DB	64
3.3.5. Conclusiones	66
4. Flujo de trabajo	67
4.1. Inferencia de los modelos de segmentación	68
4.2. Alineación y extracción de parches	70
4.3. Refinamiento de las máscaras de segmentación	71
4.4. Extracción de datos cuantitativos	72
4.4.1. Extracción de información de glándulas	73
4.4.2. Extracción de información de núcleos	73
5. Resultados	77
5.1. Entrenamiento de HistAuGAN	78
5.2. Entrenamientos del modelo de fibrosis	80
5.2.1. Aumentaciones de datos conservadas de QA	80
5.2.2. F-DB aumentada con HistAuGAN	84
5.2.3. <i>Transfer learning</i> entre BLCA-DB y F-DB	84
5.2.4. Conjunto de prueba	84
5.2.5. Resultados	85
5.2.6. Conclusiones	86
5.3. Extracción de información cuantitativa	86
5.3.1. Experimentos realizados con máscaras de glándulas	87
5.3.2. Experimentos realizados con máscaras de núcleos	92
6. Conclusiones	99
6.1. Conclusiones generales	99
6.2. Trabajo a futuro	102
Anexos	103
A. Experimentos del modelo de fibrosis	103
A.1. Primeros experimentos del modelo de fibrosis	103
A.2. Migración de la base de datos de QuickAnnotator	104
A.3. Resultados de la inferencia sobre los datos de prueba de fibrosis	106
A.3.1. Mejor modelo	106
B. Configuración del ambiente de entrenamiento en ClusterUY	107
B.1. Pipelines de FastPathology	107
B.1.1. Pipeline para uso del modelo segmentador de núcleos	107
C. Cálculo de características con funciones implementadas por OpenCV	109
Referencias	111
Índice de tablas	117
Índice de figuras	118

Capítulo 1

Introducción

La fertilidad animal es un tema ampliamente estudiado por veterinarios con el principal fin de la crianza de animales que hereden características de sus progenitores. Especialmente en los equinos, se tienen yeguas muy cotizadas, utilizadas para diferentes deportes ecuestres, cuyo material genético es muy valioso. Los equinos no se seleccionan por sus características reproductivas como otros animales de producción, sino por sus aptitudes deportivas o fenotípicas.

Las yeguas son animales poliéstricos estacionales; es decir, que su ciclo estral está presente en cierta estación del año. El fotoperíodo regula la actividad reproductiva de manera que el período del ciclo de alta actividad ocurre durante las estaciones con mayor cantidad de luz solar. Mientras que el anestro, que es cuando no hay actividad reproductiva, tiene lugar con la reducción del fotoperíodo. En el caso del hemisferio sur, esto ocurre de octubre a febrero aproximadamente, en las estaciones de primavera y verano. De esta forma el parto se da en las temporadas más adecuadas para la supervivencia de sus crías, según la revisión [1].

El ciclo estral dura aproximadamente 21 días y puede dividirse en dos etapas: el estro y el diestro. El estro dura de 4 a 7 días, siendo la ovulación su evento clave. Por otra parte, luego de la ovulación, el diestro es la parte restante del ciclo, donde el sistema se prepara fisiológicamente para nutrir un embrión en caso de gestación.

La salud uterina es fundamental para el establecimiento de la gestación, por lo que es de suma importancia conocer el estado del útero de la yegua. Para esto, entre otras técnicas, se realizan las biopsias endometriales. En estas biopsias los patólogos pueden estudiar la presencia y disposición de las diferentes estructuras en el endometrio y estimar el potencial grado de fertilidad del animal.

1.1. Objetivos del proyecto

El objetivo general del proyecto es estudiar la posibilidad de incorporar nuevas técnicas para el análisis de biopsias endometriales de yeguas. Las técnicas a analizar están principalmente abocadas al área de procesamiento de imágenes y aprendizaje automático. También se incluye el estudio de herramientas de trabajo, como *frameworks* para el análisis de las muestras biológicas y librerías para el

Capítulo 1. Introducción

procesamiento digital de las mismas.

Dentro del alcance del trabajo se encuentra la tarea de identificar estructuras biológicas de interés en las muestras, por medio de la aplicación de modelos de aprendizaje automático de segmentación.

Adicionalmente, se profundiza el análisis extrayendo información de interés de las estructuras identificadas. La automatización de la extracción de datos cuantitativos aporta a la base de conocimientos del área para generar estadísticas y unificar criterios al respecto de las patologías estudiadas.

Por último, se organizan las tareas propuestas por medio de un flujo de trabajo secuencial que integra las herramientas y los entornos de trabajo estudiados.

Creemos que la implementación de nuevas técnicas en el área sería de gran asistencia para la labor cotidiana de los patólogos. Contar con herramientas que señalen una región de la muestra que requiere un análisis más detallado podría contribuir a enfocar los esfuerzos y el tiempo de evaluación de la muestra en las regiones más relevantes.

1.2. Contexto del problema

1.2.1. Biopsias endometriales

La biopsia endometrial es un procedimiento que consiste en extraer una sección del endometrio de una yegua y estudiar la muestra a nivel microscópico. Inicialmente se coloca a la yegua en un cubículo restringido, colocando la cola de manera de no estorbar en la técnica. Se realiza un lavado de los genitales externos y se introduce una pinza para biopsias esterilizada en el tracto reproductivo. Mediante el uso de un guante esterilizado que cubre todo el brazo, se introduce la mano por el recto con el fin de asistir y guiar la pinza. Al llegar a las paredes del útero se cierra la pinza, extrayendo una muestra. Posteriormente la muestra es conservada con fijadores, como por ejemplo formalina al 4% neutra tamponada, para finalmente laminarla y colocarla en el portaobjetos. El proceso de extracción de la muestra se detalla en [2], y se ve ilustrado en la figura 1.1.

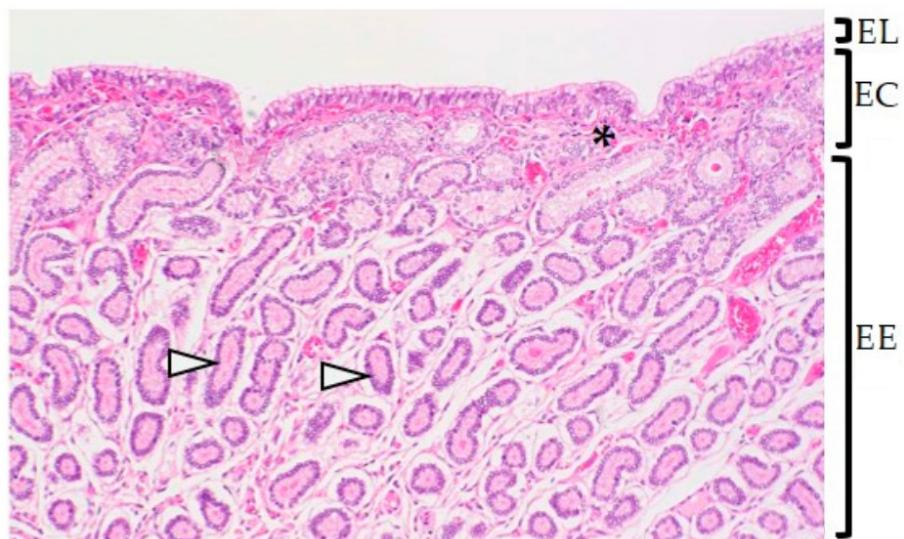
La muestra es teñida mediante hematoxilina y eosina (H&E), con el propósito de visualizar diferentes estructuras de interés. La hematoxilina es un componente básico que tiñe estructuras ácidas (basófilas) en tonos azulados, como pueden ser los núcleos celulares, mientras que la eosina tiñe estructuras de pH básico, como son las componentes citoplásmicas, con tonos rosa. Este tipo de técnica permite la visualización de las diferentes estructuras endometriales, que varían su estado y disposición. La muestra está compuesta por el epitelio luminal y la lámina propia. Esta última contiene una capa más superficial llamada estrato compacto y otra más profunda, el estrato esponjoso. El estrato compacto se caracteriza por una mayor densidad en el estroma, mientras que en el estrato esponjoso la densidad disminuye. En la figura 1.2 se pueden ver las estructuras mencionadas.

Durante el inicio del estro el epitelio luminal se presenta estratificado y con mayor altura (entre $15\mu m$ a $20\mu m$), y a medida que avanza el ciclo la altura del epitelio disminuye progresivamente. Asimismo presenta edema (retención de líqui-

1.2. Contexto del problema



Figura 1.1: Imágenes que ilustran el proceso de obtención de una biopsia uterina de yegua.



(a) Epitelio luminal (EL). Estrato compacto (EC). Estrato esponjoso (EE). Glándulas más espaciadas (flechas blancas). Glándulas mas compactadas (asterisco).

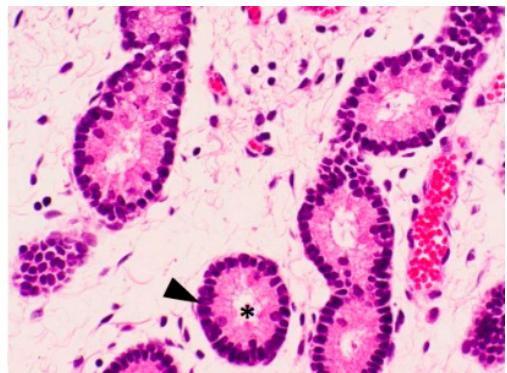
Figura 1.2: Muestra de endometrio de yegua teñida con H&E [3]. Se señalan las distintas estructuras de interés.

dos) en el estroma. Las glándulas proliferan caracterizándose por una disminución en la luz, los núcleos presentan forma ovoide y son mas elongados.

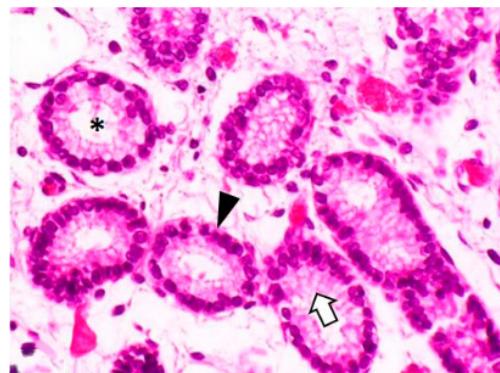
Durante el período de diestro se presenta un aumento de la densidad glandular, que se acentúa debido a la disminución del edema. El epitelio luminal disminuye en altura, en algunos casos hasta $10 \mu m$. En cuanto a la forma de las glándulas, aumenta su luz, sus núcleos tienen forma más circular, y el citoplasma tiende a tener una tinción mas pálida (aunque la intensidad de la coloración también depende de la tinción de cada muestra). Mientras que en el estro las glándulas se presentan mas proliferativas, en el diestro se vuelven mas secretoras. En la figura 1.3 se visualizan estas diferencias.

En el período de anestro, la disposición celular es sumamente diferente a las etapas anteriores. Las células del epitelio luminal se encuentran atrofiadas y se

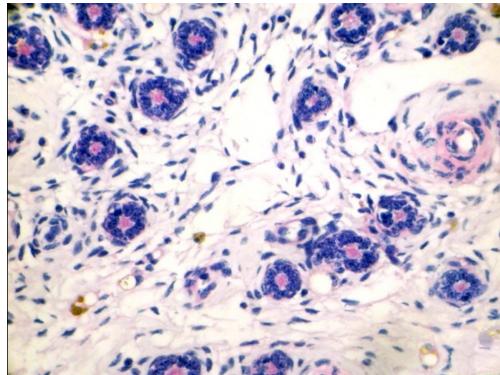
Capítulo 1. Introducción



(a) Estro: Los núcleos se presentan más elongados (flecha negra) y la luz de la glándula es más reducida (asterisco).



(b) Diestro: Núcleos más circulares (flecha negra), aumenta la luz de la glándula (asterisco) y el citoplasma dentro de la glándula tiene una textura más espumosa (flecha blanca).



(c) Anestro: Las células son más basofílicas que en otros períodos. La luz de las glándulas tiene elevado material eosinofílico (rosado). La densidad glandular es mucho menor que en el estro y el diestro.

Figura 1.3: Diferencias de la disposición glandular entre cada período del ciclo reproductivo. Adaptada de [3].

presentan de forma cúbica, mientras que el citoplasma es más basofílico (azulado). Por otra parte, la luz de las glándulas tiene elevado material eosinofílico (rosado) y la densidad glandular es mucho menor.

Artefactos

Como menciona Snider et al [4], la calidad de las muestras puede estar alterada por diferentes factores vinculados a su extracción, que generan los denominados artefactos. Estos generan alteraciones en el tejido, pero no están vinculados con su estado real, sino con la técnica de extracción. Por ejemplo, demorar la inmersión en el fijador puede causar la perdida de integridad de la muestra. Otros artefactos son las roturas de la muestra, invaginación del tejido (superposición), hemorragias causadas en la extracción, entre otros.

1.2. Contexto del problema

1.2.2. Patologías y anomalías

A continuación se presentan algunos ejemplos de las patologías más frecuentes y relevantes que se observan en la biopsia uterina.

Endometrosis

La endometrosis es una enfermedad equina, cuya frecuencia y severidad aumentan con la edad de la yegua. Se basa en la generación de una fibrosis periglandular la cual afecta su funcionalidad. También es común encontrar fibrosis bajo el epitelio luminal. Las fibras de colágeno (generadas por los fibroblastos) se disponen de forma concéntrica alrededor de una o varias glándulas, en forma de nido, ver figura 1.4c. En consecuencia, las glándulas quedan aisladas de los vasos sanguíneos y no reciben la nutrición adecuada como tampoco el estímulo hormonal normal. Frecuentemente las glándulas comienzan a dilatarse, generando deformaciones las cuales se observan agrupadas en estructuras llamadas nidos glandulares, rodeados de fibrosis.

Es importante señalar que los nidos glandulares patológicos deben ser diferenciados de los nidos que se dan en el estro a causa de la proliferación de las glándulas. Además, la endometrosis no debe ser confundida con la *endometriosis*, enfermedad que padecen los humanos y primates.

Endometritis

La endometritis es la inflamación del endometrio, muchas veces causada por una infección, aunque en otras ocasiones puede ser causada por la orina o el semen. Esta patología uterina es la mas frecuente y la de mayor impacto económico en la cría equina. La endometritis es más frecuente en yeguas de mayor edad y en aquellas que hayan tenido partos. A nivel microscópico la endometritis se presenta como un exceso de células inmunitarias respecto a la etapa del ciclo que se encuentra. Puede tratarse de una endometritis superficial, en el caso que la acumulación de células se encuentre en el estrato compacto, o endometritis profunda si también están presentes en el estrato esponjoso. Asimismo, la inflamación puede caracterizarse como aguda o crónica dependiendo del tipo de célula del sistema inmune presente en el tejido. En caso de la inflamación aguda, predominan los polimorfonucleares (PMN), mientras que en estado crónico predominan los linfocitos. En las biopsias podemos encontrar diversos tipos de células inflamatorias, como son por ejemplo los polimorfo-nucleares, neutrófilos, linfocitos, eosinófilos, basófilos, entre otros. Muchas de estas presentan características diferenciables, como por ejemplo, los linfocitos tienden a ser más pequeños, oscuros y redondos, mientras que los neutrófilos tienen un núcleo multilobulado (de 2 a 5 lóbulos conectados por delgados puentes). En la figura 1.5 se presentan algunas de ellas.

Lagunas linfáticas

Las lagunas linfáticas son dilataciones acentuadas en los vasos linfáticos del endometrio. Pueden tener múltiples formas y medir varias decenas de micróme-

Capítulo 1. Introducción

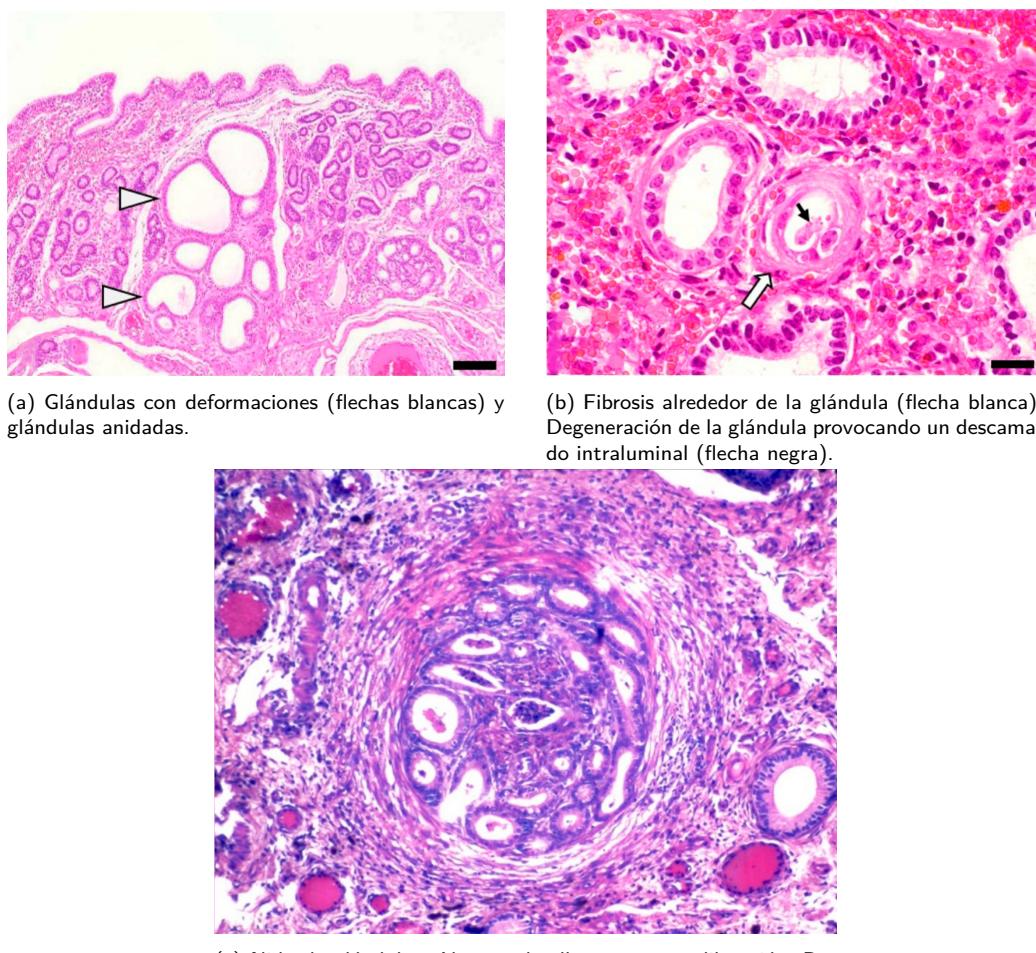


Figura 1.4: Muestras de epitelio con endometrosis. Adaptado de [3].

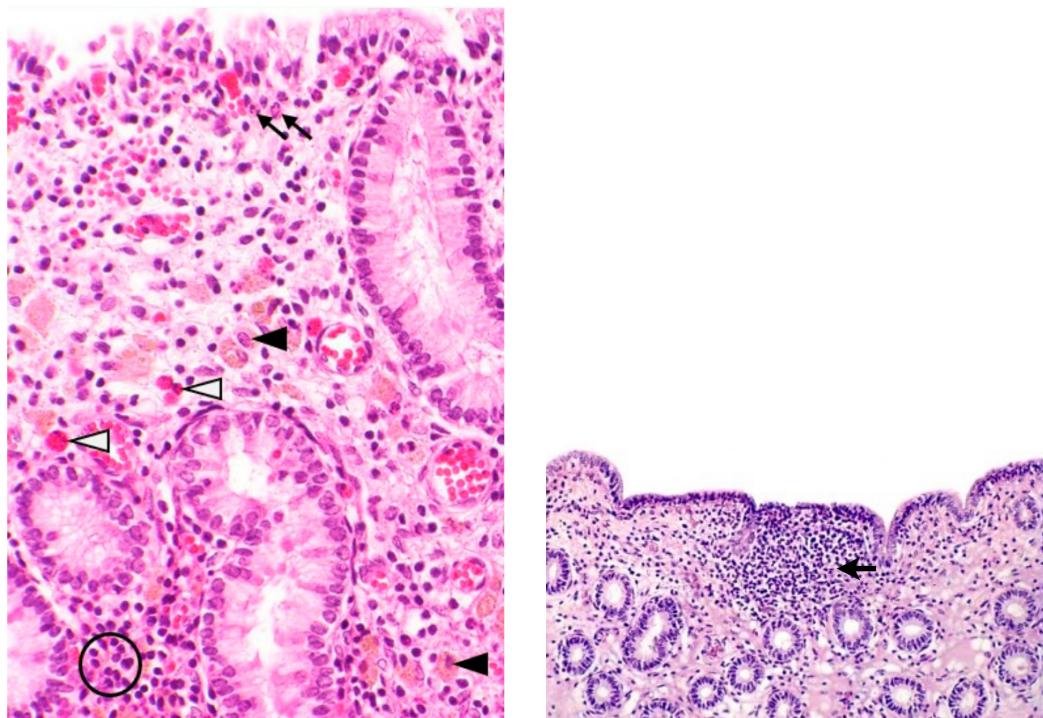
tros, además de presentarse de forma aislada o distribuidas en toda la muestra. Su presencia es importante cuando aparecen en gran número. Es importante diferenciarlas de artefactos de la muestra debido a su forma. En la figura 1.6 se presenta un epitelio con esta estructura.

Clasificación Kenney-Doig

Las biopsias endometriales son estudiadas por veterinarios patólogos. En cada muestra se estudian la presencia y disposición de las estructuras y se reportan anomalías. En el año 1986 se presenta la escala de Kenney-Doig [5] que intenta describir la particularidades de la muestra con el fin de clasificarla en cuatro categorías. Estas categorías se han correlacionado con la capacidad esperada de la yegua de llevar a cabo una gestación (tasa de parto o *Foaling Rate*). Esta clasificación se presenta en la tabla 1.1.

La escala de Kenney-Doig es una referencia a la hora de dar un diagnóstico, sin

1.3. Antecedentes



(a) Presencia de múltiples linfocitos en toda la muestra (círculo), también se encuentran algunos neutrófilos (flecha fina). Contiene siderocitos (flecha negra), que indican una hemorragia previa, y presencia de eosinófilos (flecha blanca).

(b) Acumulación de linfocitos en el estrato compacto (flecha negra).

Figura 1.5: Muestras de epitelio con endometritis [3].

embargo esta sujeta a la interpretación y subjetividad de los patólogos (experiencia y criterio). La falta de datos cuantitativos sobre la muestra provoca que los criterios estén abiertos a discusión. Estudiar estas muestras puede ser complejo ya que contienen gran cantidad de información y observarlas en su totalidad requiere tiempo y meticulosidad.

1.3. Antecedentes

La búsqueda de antecedentes inicialmente procuró identificar trabajos que utilizaran alguna técnica de aprendizaje automático para el análisis de imágenes de tejidos de animales. Sin embargo, se observó que dicha tarea recién se está comenzando a desarrollar con más fuerza en la medicina humana, y aún no ha sido aplicada en el campo de la veterinaria. De esta forma, los antecedentes con los que se trabajó están enfocados al estudio de patologías humanas.

Capítulo 1. Introducción

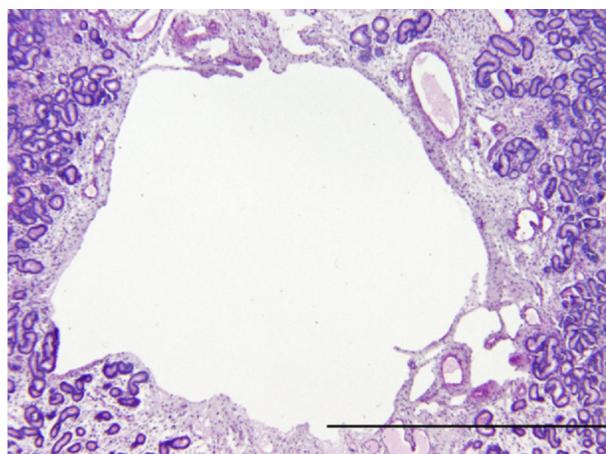


Figura 1.6: Laguna linfática. Adaptada de [4].

Grado	Características encontradas	Tasa de Parto
I	No muestra alteraciones significativas	>80 % - 90 %
IIa	Endometritis leve o, Endometrosis leve o, Lagunas linfáticas leve o, Atrofia parcial del endometrio durante el estro	50 % - 80 %
IIb	Endometritis moderada o, Endometrosis moderada o, Lagunas linfáticas moderadas o, Infecundidad >2 años o, Presencia de varios items de la categoría IIa	10 % - 50 %
III	Endometritis severa o, Endometrosis severa o, Marcadas lagunas linfáticas o, Marcada atrofia del endometrio durante el estro o, Presencia de 3 o más items de IIa o 2 o más de la categoría IIb o III	<10 %

Tabla 1.1: Escala de Kenney-Doig. Adaptada de [4].

1.3.1. Análisis de imágenes en medicina para la identificación de patologías

Permanentemente se están desarrollando nuevos *software* y aplicaciones integradas al flujo de trabajo de los médicos, para asistir y agilizar algunas tareas específicas. En el área de la patología, se pueden encontrar interfaces gráficas para visualizar, anotar y procesar imágenes de tejido en alta resolución. Un ejemplo reciente es Halo AI [6]. Estas imágenes se denominan WSI por sus siglas en inglés

1.3. Antecedentes

(*Whole Slide Images*) y son obtenidas a partir de un escaneo completo de un preparado histológico. Las WSIs tienen un amplio rango de resolución y requieren un espacio de almacenamiento considerable, pudiendo alcanzar valores de resolución como por ejemplo 180.000×180.000 píxeles y 4GB de tamaño. Por estos motivos, cargar en memoria RAM y procesar estas imágenes no se puede realizar por medio de los programas normalmente utilizados, sino que requieren de optimizaciones y métodos de trabajo especiales. En la sección 1.5 se pueden encontrar más detalles de las herramientas existentes para ello.

En cuanto al uso de WSIs en el estudio de patologías se observa que, al tratarse de imágenes de tan alta resolución, permiten caracterizar mejor el fenotipo de las células o estructuras analizadas y estudiar relaciones espaciales en el tejido. Las células o estructuras se pueden analizar de forma individual o en su conjunto, aplicando prácticas de análisis estadísticos. Más aún, las WSIs facilitan la obtención de información cuantitativa de toda la muestra. Previamente, las muestras se comparaban en términos cualitativos y en base a la experiencia de cada profesional, por lo que era difícil alcanzar un criterio único y objetivo en esta práctica.

Dos trabajos que fueron tomados como referencia y señalan el valor que tiene el uso de las WSI en el campo de la patología son [7] y [8]. Estos trabajos proponen formas de integrar estas técnicas al procedimiento de los patólogos. A modo de ejemplo, en [7] se menciona la posibilidad de utilizar el análisis de tejidos por medio de WSIs para obtener diagnósticos más precisos de pacientes y prever la respuesta a los tratamientos.

1.3.2. Aplicación de técnicas de aprendizaje profundo en medicina

Con el avance de las redes convolucionales (*Convolutional Neural Network - CNN*) y su principal uso enfocado en el análisis de imágenes, se han desarrollado múltiples algoritmos en el campo de la medicina para detectar, segmentar, etiquetar y clasificar distintos tipos de estructuras biológicas en WSIs. La intervención de los patólogos en el desarrollo de los algoritmos es fundamental para definir qué es lo que se quiere identificar, sus principales características, fijar parámetros propios del problema a tratar, como por ejemplo umbrales, y validar el desempeño de las redes entrenadas.

Un problema recurrente al querer entrenar una CNN para el análisis de imágenes médicas es la falta de bases de datos etiquetadas y relacionadas al problema a resolver. Anotar una WSI de forma completa requiere de varias horas de trabajo de un técnico especializado, por lo que etiquetar una cantidad de WSIs suficiente para entrenar una red resultaría muy costoso. Una solución habitual a este problema es trabajar con modelos pre-entrenados sobre bases de datos genéricas, como por ejemplo ImageNet [9], y mantener los pesos de las capas más bajas que corresponden a aspectos de bajo nivel (*transfer learning*). Luego se realiza un ajuste fino entrenando las últimas capas con imágenes específicas de la problemática a tratar (*fine tuning*).

Para complementar lo anterior, en [10] se sugiere un proceso iterativo con intervención humana para el entrenamiento de los modelos (*active learning*): El modelo

Capítulo 1. Introducción

se inicializa haciendo *transfer learning* con un modelo entrenado sobre *ImageNet*. Se comienza anotando unas pocas WSIs y se entrena con dichas anotaciones. Con el resultado obtenido luego del primer entrenamiento se realiza inferencia sobre nuevas WSIs y un patólogo corrige las predicciones. Utilizando las WSIs anotadas inicialmente y las correcciones realizadas por el patólogo, se vuelve a entrenar el modelo, refinando los pesos obtenidos del entrenamiento anterior. Se repiten los mismos pasos hasta obtener una desempeño satisfactorio al realizar inferencia.

Una de las tareas más recurrentes en las que se han aplicado prácticas de Aprendizaje Profundo en el campo de la medicina ha sido el diagnóstico de cáncer. En la tesis [11] se propone un flujo de trabajo de cinco pasos para identificar la presencia o ausencia de tumor, a partir de muestras de ganglios centinela de mama. Tomamos esta tesis como ejemplo para profundizar en la metodología de trabajo de análisis de WSIs de tejidos con tinción H&E, mediante técnicas de aprendizaje profundo. Los cinco pasos desarrollados en el trabajo consisten en:

1. Detección de la región de interés (ROI)

Procesar una WSI completa es una tarea muy costosa computacionalmente, por lo que en la etapa inicial se busca reducir el espacio de trabajo convenientemente para optimizar el procesamiento posterior. Esto se hace generando una máscara binaria que identifique y separe el fondo del tejido.

2. Construcción de base de datos de entrenamiento

Las WSIs completas no conforman una entrada aceptable para ningún modelo de aprendizaje automático, por lo que es necesario fraccionarlas en parches más chicos. Luego se aplican técnicas de aumentación de datos, como recortes y rotaciones, para aumentar la variabilidad y evitar el sobreajuste a los datos de entrenamiento.

3. Entrenamiento del modelo de aprendizaje automático

Dependiendo de la tarea a realizar se selecciona el modelo adecuado. En el trabajo [11] se utiliza una red con arquitectura Inception-V3 [12]. El criterio para seleccionar este modelo se basó en la velocidad y estabilidad del mismo. La entrada a la red son imágenes de 256×256 , y la salida es una probabilidad de presencia de tumor en el parche evaluado.

4. Construcción de mapas de calor basados en las probabilidades estimadas por el modelo

Para continuar, se divide toda una WSI en parches de 256×256 y se infiere la probabilidad de presencia de tumor en cada uno de ellos. A partir de dicha información se elabora un mapa de calor de la WSI completa, señalando la zona de la muestra donde se localiza el tumor.

5. Post-procesamiento de los mapas de calor para la clasificación de la muestra

A partir del mapa de calor generado se obtiene una única probabilidad por WSI señalando la presencia o ausencia de tumor en la muestra. Para esta

1.4. Estudio de herramientas

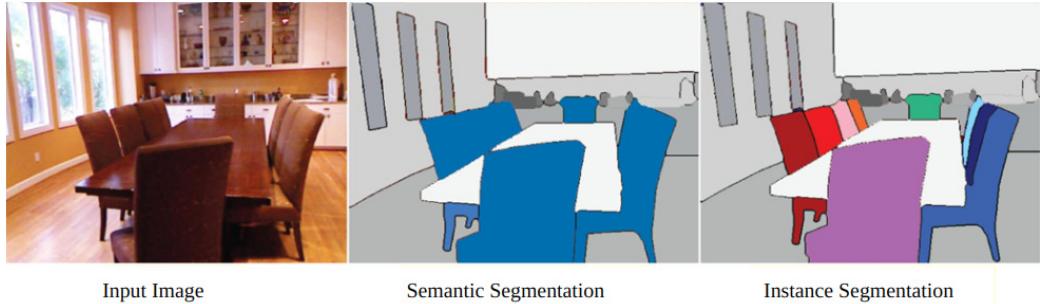


Figura 1.7: A la izquierda imagen original. En el medio segmentación semántica, se reconoce la clase silla y dónde están ubicadas. A la derecha segmentación de instancias, además de reconocer la clase silla, se diferencia cada instancia con un color distinto [13].

última parte se trabaja con modelos de aprendizaje automático clásicos, como por ejemplo Support Vector Machine (SVM) y Random Forest (RF). Se extraen del mapa de calor 28 características diseñadas a mano, y se corre el modelo para predecir el resultado final.

A modo de resumen se observa que esta metodología es comúnmente utilizada al trabajar con WSIs. A grandes rasgos, se selecciona la región de la muestra donde se encuentra el tejido, se fracciona la WSI en parches, se procesa cada imagen por separado y se extraen características globales de la muestra.

1.4. Estudio de herramientas

En esta sección se presentan las herramientas y técnicas que resultan de importancia en la desarrollo del proyecto, ya sea porque son utilizadas como parte de la implementación final, o porque parte del trabajo está basado en ellas.

1.4.1. Modelos de segmentación

La segmentación semántica de imágenes es la tarea de agrupar en regiones los píxeles de una imagen según la clase a la que pertenecen. Para resolver este problema se debe reconocer qué hay en la imagen, y dónde está ubicado. A diferencia de la segmentación de instancias, en esta tarea no se distinguen diferentes instancias de la misma clase. En la imagen 1.7 se presenta un ejemplo de lo mencionado. También es diferente a la tarea de detección de objetos, ya que en esta última se reconoce la posición del objeto en la imagen, pero no se especifican cuáles son los píxeles que conforman el objeto.

1.4.2. Fully Convolutional Network (FCN)

Es una red neuronal profunda que consiste únicamente de capas convolucionales. Fue presentada en el trabajo *Fully Convolutional Networks for Semantic Segmentation* [14] del año 2015. Su objetivo es la segmentación semántica de la

Capítulo 1. Introducción

entrada, con especial énfasis en el detalle de la segmentación. Es la base para otras arquitecturas que se presentan en las secciones posteriores.

Esta arquitectura tiene menor cantidad de parámetros que una con capas *fully-connected* que brinde resultados similares, lo que resulta en menor tiempo de entrenamiento e inferencia, y requiere menor cantidad de datos para entrenar. Estas redes a diferencia de las redes convolucionales más tradicionales, sustituyen las últimas capas de la red por convoluciones de *kernels* de tamaño 1×1 en lugar de capas densas. Además, es capaz de aceptar una entrada de tamaño arbitrario, y sacar una salida de tamaño acorde a la entrada [14], lo que es útil a la hora de trabajar con imágenes.

1.4.3. U-Net

Esta red fue presentada en el trabajo *U-Net: Convolutional Networks for Biomedical Image Segmentation* [15] en el año 2015, con el objetivo de realizar segmentación semántica en imágenes biomédicas. El proyecto fue tan novedoso que el mismo año de su lanzamiento ganó el ISBI cell tracking challenge¹. Hasta ese momento las arquitecturas convencionales constaban de capas sucesivas donde la dimensión de las capas de activación disminuye a medida que avanzaba la red.

La idea central de esta arquitectura es generar una estructura simétrica que permita captar características de los datos de entrada a distintas escalas. Con este fin, en lugar de tener un modelo convencional, se propone la estructura de la figura 1.8. Esta consiste en un camino que se va contrayendo, seguido de un camino expansivo, obteniendo así la forma de 'U' que le da nombre a la red.

El camino contractivo (también llamado *encoder*) consiste en aplicar sucesivas veces las siguientes operaciones: Dos convoluciones con kernel de tamaño 3×3 seguida de una activación ReLU (en el esquema de la figura 1.8 señalado como una flecha azul), y un *max pooling* de 2×2 que reduce la dimensión espacial del vector (flecha roja). Entre un bloque de *downsampling* y el siguiente, se duplica la cantidad de mapas de características de entrada.

El bloque más inferior de la figura 1.8, que tiene dos capas convolucionales de 1024 filtros cada una, y produce los *feature maps* de menor dimensión de toda la red, será llamado *espacio latente*.

El camino expansivo (llamando *decoder*) busca una simetría con el *encoder* utilizando una operación de *upsampling* en lugar de un *max pooling* (señalado con una flecha verde hacia arriba). Esta operación reduce la cantidad de mapas de activación a la mitad. Además, la entrada a un bloque de *upsampling* se concatena con el mapa de activación correspondiente del paso contractivo (marcado con una flecha gris en el esquema); esta conexión resulta en una mejor localización de la segmentación en la imagen de salida. El camino expansivo finaliza con una convolución con un *kernel* de tamaño 1×1 .

De esta red se obtienen segmentaciones de mayor precisión a nivel espacial, característica de especial interés en las imágenes biomédicas.

¹2015 IEEE ISBI (International Symposium on Biomedical Imaging)

1.5. Frameworks

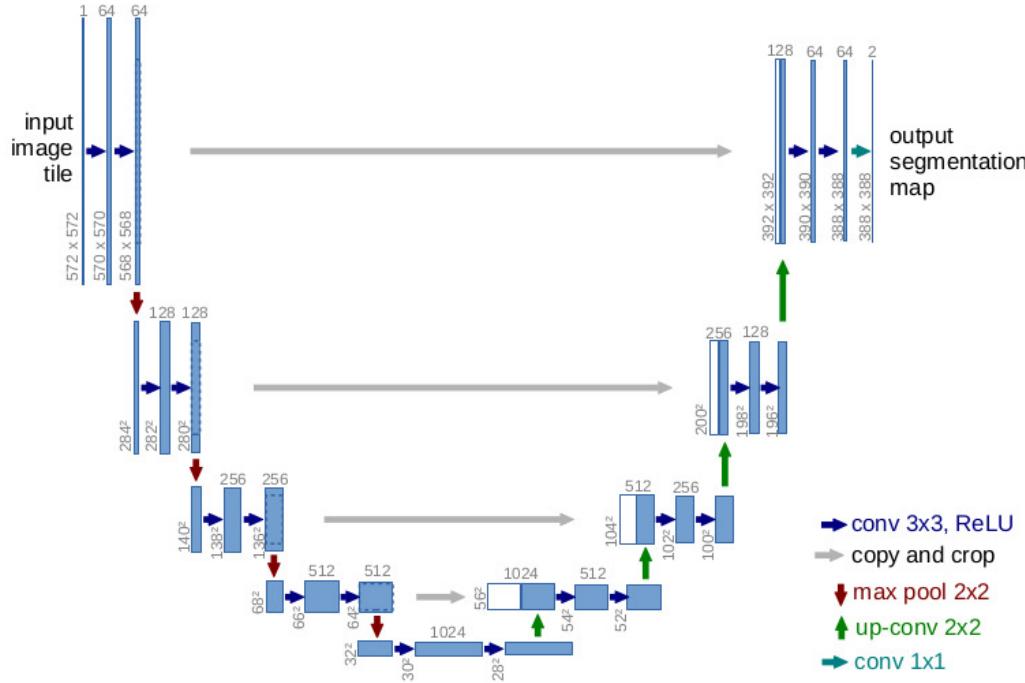


Figura 1.8: Arquitectura de la U-Net. Los 4 bloques de *downsampling* forman el camino contractivo. Le sigue el llamado espacio latente. Los 4 bloques de *upsampling* forman el camino expansivo. Adaptado de [15].

1.4.4. SegNet

En este trabajo [16] del año 2016 se desarrolla una FCN basada en una arquitectura *encoder-decoder* diseñada para lograr segmentación píxel-a-píxel eficientemente. Fue desarrollada para aplicaciones de comprensión de escenas de calles y carreteras, donde se requiere que la red sea capaz de comprender el contexto entre las diferentes clases, además de delinear los objetos detectados con precisión. Debido a que es una FCN, cuenta con menor cantidad de parámetros que redes que usan capas *fully connected*, lo que la hace más eficiente comparativamente, tanto en su entrenamiento como en inferencia, y permite trabajar con datos de entrada de tamaño variable.

1.5. Frameworks

1.5.1. Quick Annotator

Este proyecto tiene por objetivo acercar el aprendizaje automático a personas de áreas biomédicas que no se especializan en ello (por ejemplo, patólogos), facilitar el manejo y etiquetado de WSIs, agilizar el proceso de etiquetado, entrenar un modelo de segmentación, y hacer inferencia a partir del modelo [17].

Capítulo 1. Introducción

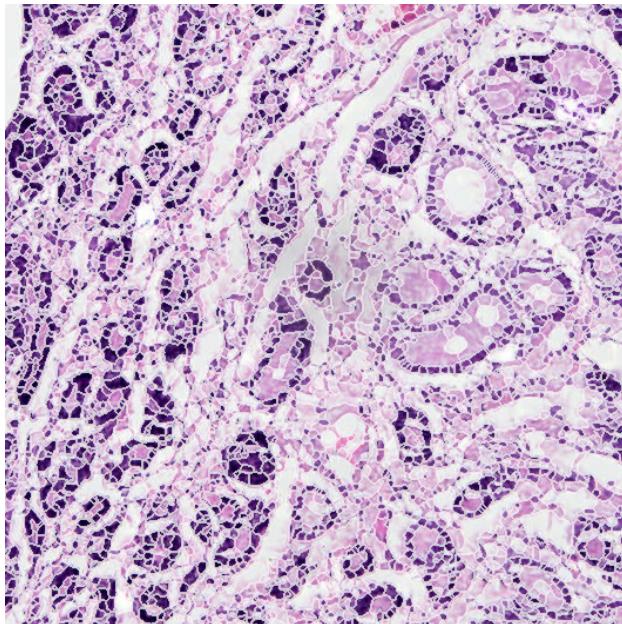


Figura 1.9: Parche de biopsia endometrial de yegua segmentada en superpíxeles generados por QuickAnnotator. Se agrupan los píxeles en regiones de color similar.

Consiste de una interfaz web en la que se puede armar una base de datos a partir de parches de WSIs, anotarlos, y usarlos para entrenar un modelo. Inicialmente, el usuario genera los parches utilizando una herramienta provista en el proyecto. El modelo a entrenar es una U-Net, y no está previsto entrenar con otras arquitecturas.

Con el fin de inicializar los parámetros del modelo, éste es entrenado de forma no-supervisada de la misma forma que se entrenaría un *auto-encoder*. En esta etapa el modelo es capaz de segmentar la imagen en *superpíxeles*, lo que facilita el proceso de anotación. En la figura 1.9 se ve un ejemplo de una imagen segmentada en superpíxeles.

Luego comienza un proceso de *active learning*, en el cual el usuario hace algunas anotaciones manuales sobre algunos parches (sin asistencia o aprovechando los superpíxeles generados) y entrena el modelo de forma supervisada. Luego, cuando se selecciona un nuevo parche, el modelo lo segmenta, y el usuario tiene la posibilidad de aceptar, rechazar o corregir las anotaciones generadas. Eventualmente se reentrena el modelo con las correcciones realizadas. El proceso termina cuando el usuario está conforme con el resultado, o cuando el mismo no presenta mejoría. En la figura 1.10 se ve un ejemplo de una segmentación generada por QuickAnnotator.

Al terminar con el proceso de *active learning*, queda generado un *dataset* con anotaciones de las estructuras biológicas de interés, que puede ser utilizado luego para entrenar otros modelos de aprendizaje automático y continuar con el análisis. Adicionalmente se cuenta con el modelo de segmentación entrenado.

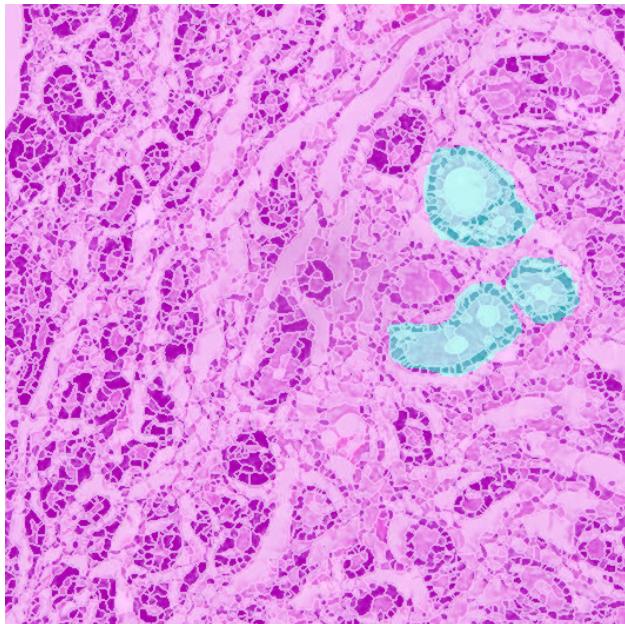


Figura 1.10: Parche de biopsia endometrial de yegua con glándula segmentada usando QuickAnnotator. El color celeste representa la clase positiva, mientras que la etiqueta rosada representa la clase negativa. Los superpíxeles ayudan al etiquetado ya que cada región se pinta del mismo color.

1.5.2. NoCodeSeg

NoCodeSeg [18] se trata de un proyecto que busca implementar un segmentador de imágenes biomédicas por medio de programas de código abierto donde no sea necesario programar. El objetivo del proyecto es que patólogos y otros científicos, sin conocimientos de programación, puedan entrenar un modelo de aprendizaje profundo que permita detectar clases de estructuras biológicas dentro de una WSI. Está implementado por medio de tres programas que permiten aplicar un proceso iterativo para entrenar el modelo deseado.

Uno de los programas utilizados se llama QuPath [19]. Este programa sirve para realizar anotaciones sobre WSIs y luego exportar parches de la misma, con su respectiva máscara de segmentación. Para implementar y entrenar un modelo de aprendizaje profundo se utiliza DeepMIB. Este programa cuenta con una interfaz gráfica donde es posible fijar ciertos parámetros característicos del modelo a entrenar. De momento cuenta con 3 modelos posibles, U-Net 2D y 3D, y SegNet. Una vez fijados los parámetros del modelo se puede realizar el entrenamiento desde la misma interfaz. Por último, el tercer programa se llama FastPathology [20], y sirve para realizar inferencia a partir de modelos de aprendizaje automático. El programa genera una máscara de segmentación de toda la WSI a partir del modelo entrenado y cuenta con una interfaz que permite visualizar el resultado. Esta herramienta está desarrollada de forma de optimizar recursos a la hora de hacer inferencia.

La figura 1.11 muestra el proceso iterativo propuesto en este proyecto, donde

Capítulo 1. Introducción

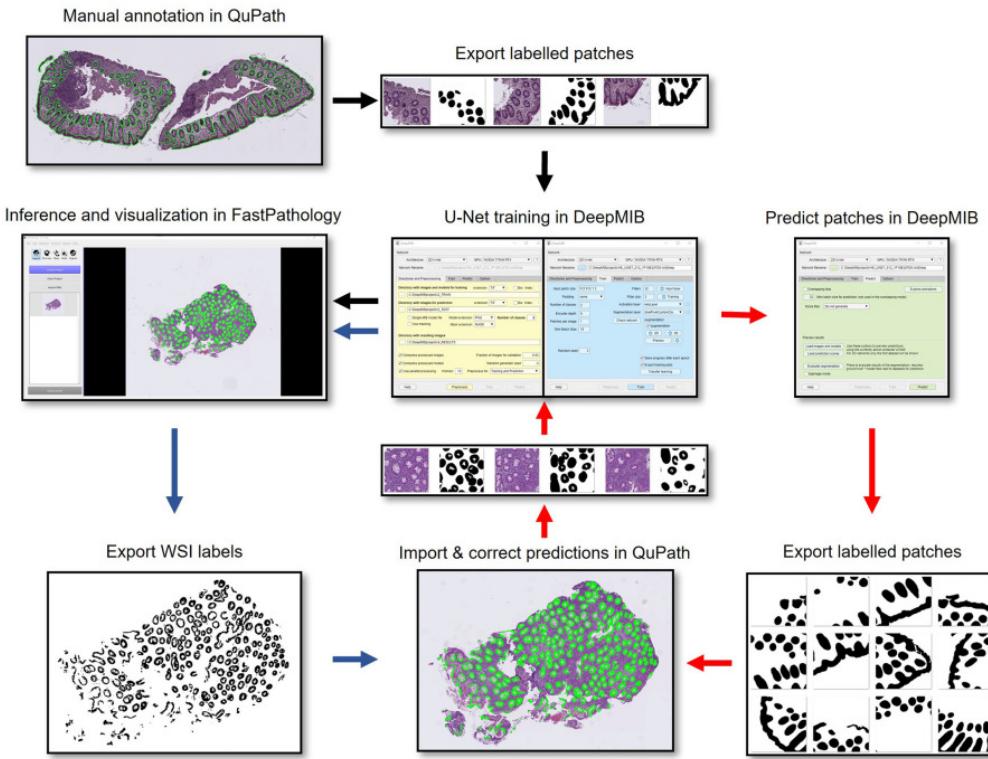


Figura 1.11: Flujo de trabajo propuesto en el proyecto NoCodeSeg [18]. Se comienza por abrir una WSI en QuPath y anotar estructuras de interés (imagen superior izquierda). La imagen y su máscara son recortados en parches, y con ellos se entrena un modelo segmentador, a través de DeepMIB. Luego se hace inferencia sobre nuevos parches de la WSI con el modelo entrenado, y se exportan a QuPath, donde se modifican a gusto del usuario, para volver a repetir este proceso.

a través de un método de *active learning* se logra no solo obtener el modelo de segmentación entrenado, sino que también se alcanza el etiquetado de una base de datos.

1.6. Organización del documento

Los capítulos que continúan este documento se organizan de la siguiente forma: El capítulo 2 presenta una descripción de las imágenes de trabajo y la base de datos generada. También expone acerca de una técnica de preprocessamiento de imágenes de distintos dominio para lidiar con la diversidad de tonalidades que se pueden presentar; el capítulo 3 se concentra en la búsqueda y evaluación de modelos de segmentación de estructuras, y a su vez describe el proceso de entrenamiento de un modelo de segmentación entrenado con la base de datos del proyecto. A continuación, el capítulo 4 propone un flujo de trabajo para implementar las soluciones halladas, y el capítulo 5 detalla los resultados obtenidos. Para finalizar,

1.6. Organización del documento

el capítulo 6 comprende las conclusiones finales del trabajo y la orientación que podría tener el trabajo a futuro.

Esta página ha sido intencionalmente dejada en blanco.

Capítulo 2

Imágenes y técnicas de preprocessamiento

Las muestras utilizadas para el desarrollo del proyecto fueron puestas a disposición por la tutora María José Estradé. Se tratan de WSIs de biopsias endometriales de yeguas con tinción H&E (hematoxilina y eosina). Las biopsias fueron extraídas y procesadas en distintas instancias y en distintos laboratorios. En este capítulo se describen las características principales de estas imágenes y se profundiza en una técnica de preprocessamiento denominada normalización de color.

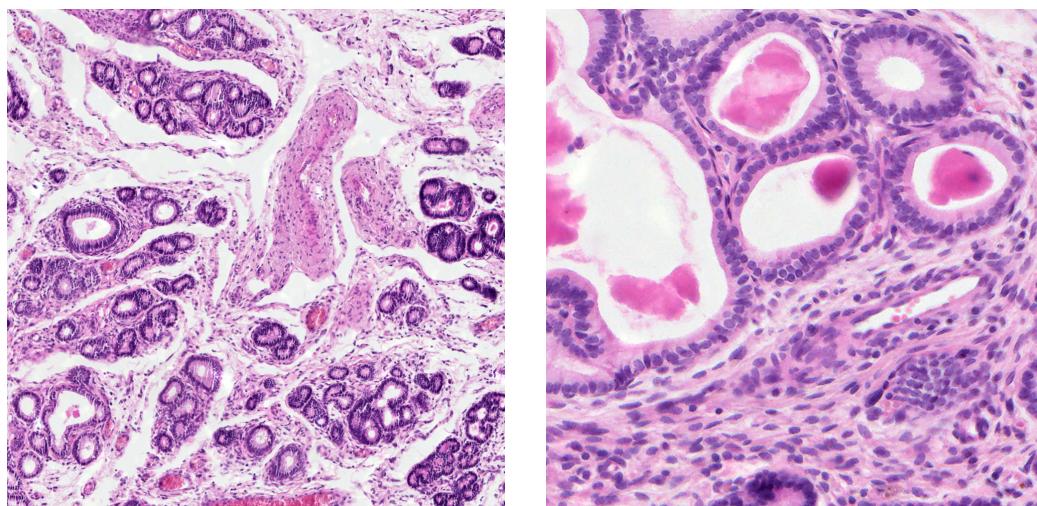
A partir del análisis bibliográfico realizado se constató que los procedimientos de normalización de color son una práctica frecuente en el procesamiento de imágenes histológicas debido a la diferencia de tonos que pueden presentar dos imágenes de muestras del mismo tejido. Estas diferencias son inherentes al proceso de generación de las muestras. Generalmente la normalización de color se aplica previo a entrenar modelos de aprendizaje automático, con el objetivo de reducir la variabilidad de color de un conjunto de imágenes. Mediante su implementación se pretende modificar el espacio de color de las imágenes, adaptando la distribución cromática a otra distribución conocida y/o deseada.

Durante la ejecución del proyecto se consideró relevante el estudio de estas técnicas para evaluar los resultados que podía brindar su aplicación previo al entrenamiento de los modelos de segmentación. Se estudiaron varios métodos y se probaron implementaciones disponibles sobre las imágenes de trabajo.

2.1. Descripción de los datos

Inicialmente se contaba con trece WSIs con tinción H&E y resolución en el entorno de $60\,000 \times 120\,000$ píxeles. El tamaño de cada uno de estos archivos WSI varía entre 213 MB y 1,2 GB, y presentan una magnificación máxima de $40\times$. Los microscopios presentan un lente fijo que aumenta $10\times$ la realidad, y lentes objetivos que tienen magnificaciones variables de $10\times$, $20\times$, o $40\times$. Es decir que cuando se habla de una magnificación máxima de $40\times$, se aumenta 400 veces la realidad. Las imágenes del tipo WSI almacenan la información en un formato *tiff*

Capítulo 2. Imágenes y técnicas de preprocesamiento



(a) Imagen de tamaño 1000×1000 píxeles a magnificación $10\times$. Se ven más estructuras con menor detalle.

(b) Imagen de tamaño 1000×1000 píxeles a magnificación $40\times$. Se ve menor cantidad de estructuras, aunque más detalladas

Figura 2.1: Ejemplos de imágenes de igual resolución a distinto nivel de magnificación.

piramidal; esto significa que se guardan varias imágenes de la muestra original, con distintos niveles de resolución, en el mismo archivo. Dependiendo el nivel de aumento que se quiera observar, qué nivel del archivo *tiff* será adecuado para ello. Por ejemplo, si se quiere observar la muestra completa será más adecuado el nivel de magnificación más bajo, en la figura 2.1a se observa una imagen a magnificación $10\times$ donde se puede apreciar mayormente el contexto de la muestra. De lo contrario, si se quiere observar una sección de la muestra con más detalle, por ejemplo los núcleos o glándulas, una mayor magnificación será la más conveniente. En la figura 2.1b se observa la misma muestra con una magnificación a $40\times$. Ambas imágenes en la figura 2.1 son del mismo tamaño (1000×1000), es decir tienen la misma resolución, pero tienen distinta magnificación.

De estas muestras, algunas presentan las patologías introducidas en la sección 1.2.2, con un grado de avance leve a moderado. Se ha de notar que muchas de las WSIs proporcionadas presentan algún tipo de desperfecto como por ejemplo: zonas borrosas, tejido desgarrado y pliegues superpuestos. Por este motivo, algunas de las muestras disponibles no pudieron ser consideradas para el trabajo.

Más avanzado el transcurso del proyecto se recibieron veintiún WSIs adicionales. Un subconjunto de estas muestras se obtuvo a partir de las mismas biopsias que la partida inicial. Fueron extraídas en el mismo laboratorio, por el mismo personal y con los mismos elementos que las primeras muestras. A pesar de ello, las imágenes obtenidas difieren ampliamente de las originales. En la figura 2.2 se puede observar la comparación entre ambas. Debido a que estas dos series de imágenes surgen de las mismas muestras, el grado de avance de las patologías es el mismo, de leve a moderado. Se observó que algunas de las WSIs nuevas presentan una coloración más acentuada. De acuerdo con la tutora, esto se debe a que los recortes de estas nuevas muestran son más anchos que los anteriores, o que

2.1. Descripción de los datos

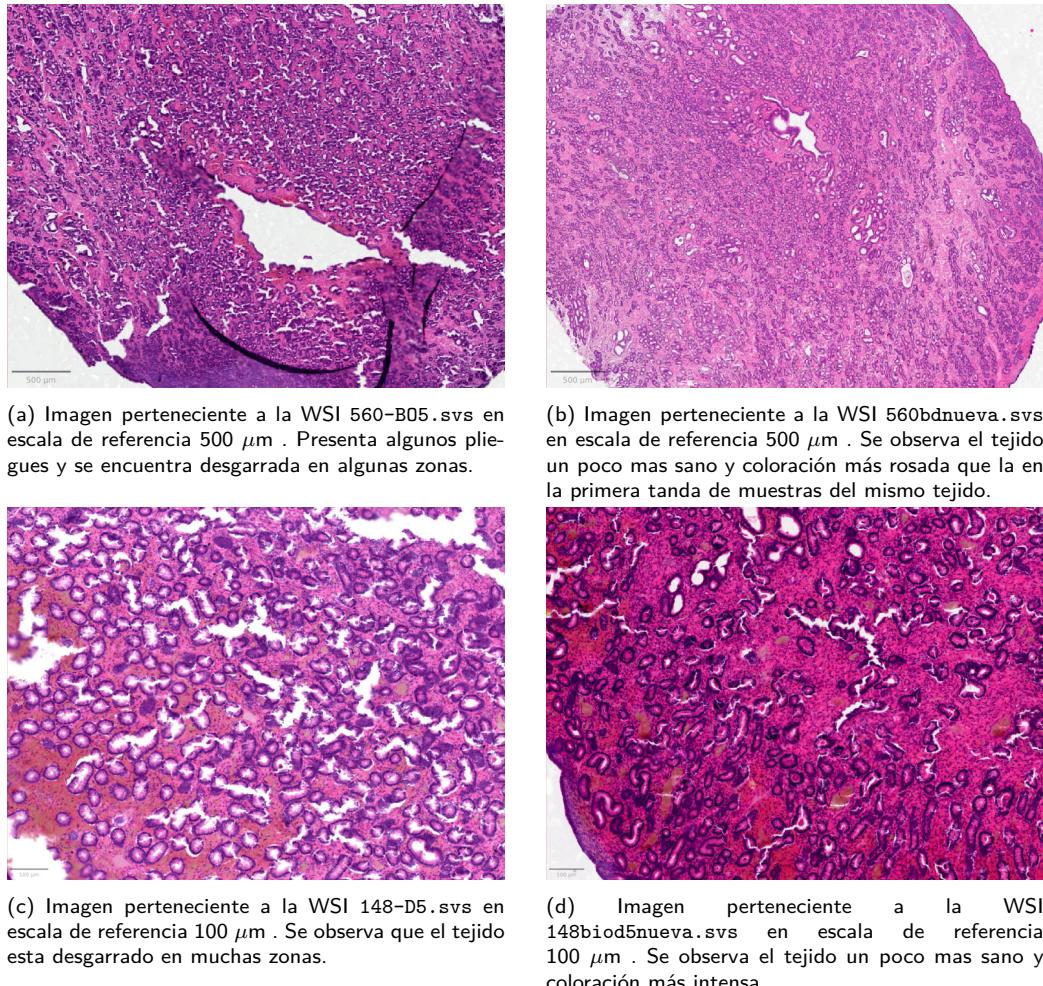


Figura 2.2: Comparación de las WSIs iniciales y la nueva partida. Las imágenes 2.2a y 2.2b corresponden a la yegua 560, mientras que las imágenes 2.2c y 2.2d corresponden a la yegua 148.

las mismas fueron expuestas por más tiempo a la tinción. Esto demuestra la gran variabilidad de las muestras generadas, aún para un mismo tejido, mediante este tipo de técnica.

Dentro de las nuevas WSIs obtenidas, también se encontraban algunas muestras proporcionadas por la doctora en veterinaria Sandra Fiala, profesora de la Universidad Federal de Pelotas. Estos archivos fueron obtenidos a través de la tutora María José Estradé. La particularidad de esta serie de WSIs es que presentan patologías con un mayor nivel de gravedad, como por ejemplo, glándulas dilatadas y dispuestas en forma de nido (signo de fibrosis periglandular), y edemas, entre otras alteraciones. En la figura 2.3 se pueden ver algunos ejemplos. Obtener muestras con patologías con un amplio espectro de gravedad es necesario para tener un conjunto de imágenes variado.

Esta serie de WSIs se particulariza por tener un bajo nivel de hematoxilina en

Capítulo 2. Imágenes y técnicas de preprocesamiento

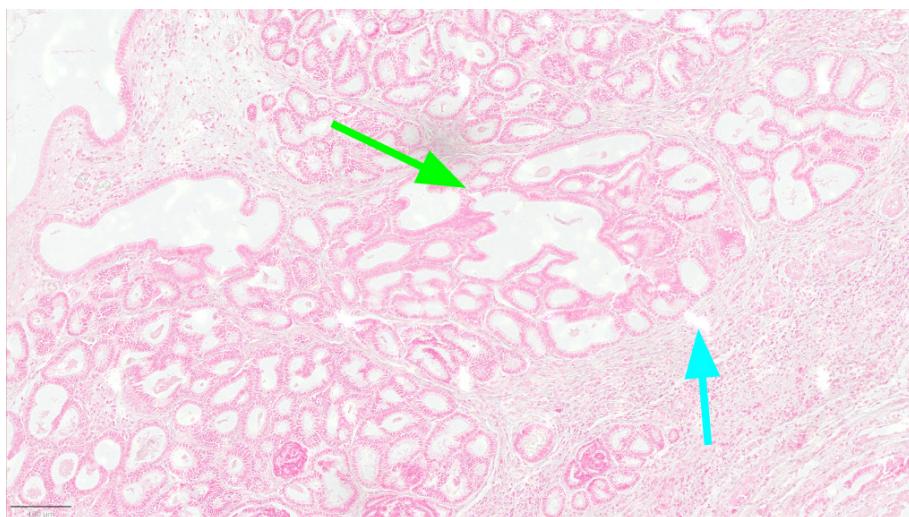


Figura 2.3: WSI del dataset - archivo *sandram6.svs*. La flecha verde señala glándulas dilatadas y dispuestas en forma de nido. La flecha celeste marca las fibras de colágeno que envuelven al nido de glándulas, generando el efecto de la patología llamada fibrosis. El tamaño puede compararse con las estructuras de las figuras 2.2c y 2.2d, cuya escala también es de $100\mu\text{m}$.

comparación con el primer grupo de WSIs adquirido. Este desperfecto en la tinción provoca que las WSIs sean más rosadas. A pesar de ello se optó por incluirlas dentro del conjunto de imágenes de trabajo, ya que presentaban características patológicas y estructurales de interés para el estudio.

2.2. Normalización de color

El caso de estudio cuenta con tres dominios claramente diferenciados: uno asociado a la primer partida de WSIs (13 archivos); otro correspondiente a la segunda partida, que se trata de las mismas muestras que las WSIs originales pero procesadas en otra oportunidad (12 archivos); y un tercer grupo conformado por muestras que presentan patologías más severas (9 archivos). Para simplificar la referencia hacia los dominios reconocidos, estos fueron denominados dominio violeta, rojo, y rosado respectivamente, por ser estos los tonos de color predominantes de cada grupo.

La figura 2.4 presenta una imagen de referencia de cada uno de los dominios. Por ejemplo, la imagen 2.4a corresponde al subconjunto violeta. El dominio violeta se caracteriza por presentar ambas tinciones, hematoxilina y eosina, sin que predomine una por sobre la otra. La imagen 2.4b pertenece al dominio rojo, en el cual la eosina toma un color más intenso, rosado oscuro o rojo, en comparación con las muestras originales. Por último la imagen 2.4c se trata de un representante del dominio rosado, que se destaca por la predominancia de la eosina, característica por su coloración rosada.

Considerando la totalidad de las imágenes disponibles en el estudio, se identificaron dos situaciones que merecía la pena analizar previo al procesamiento de las

2.2. Normalización de color

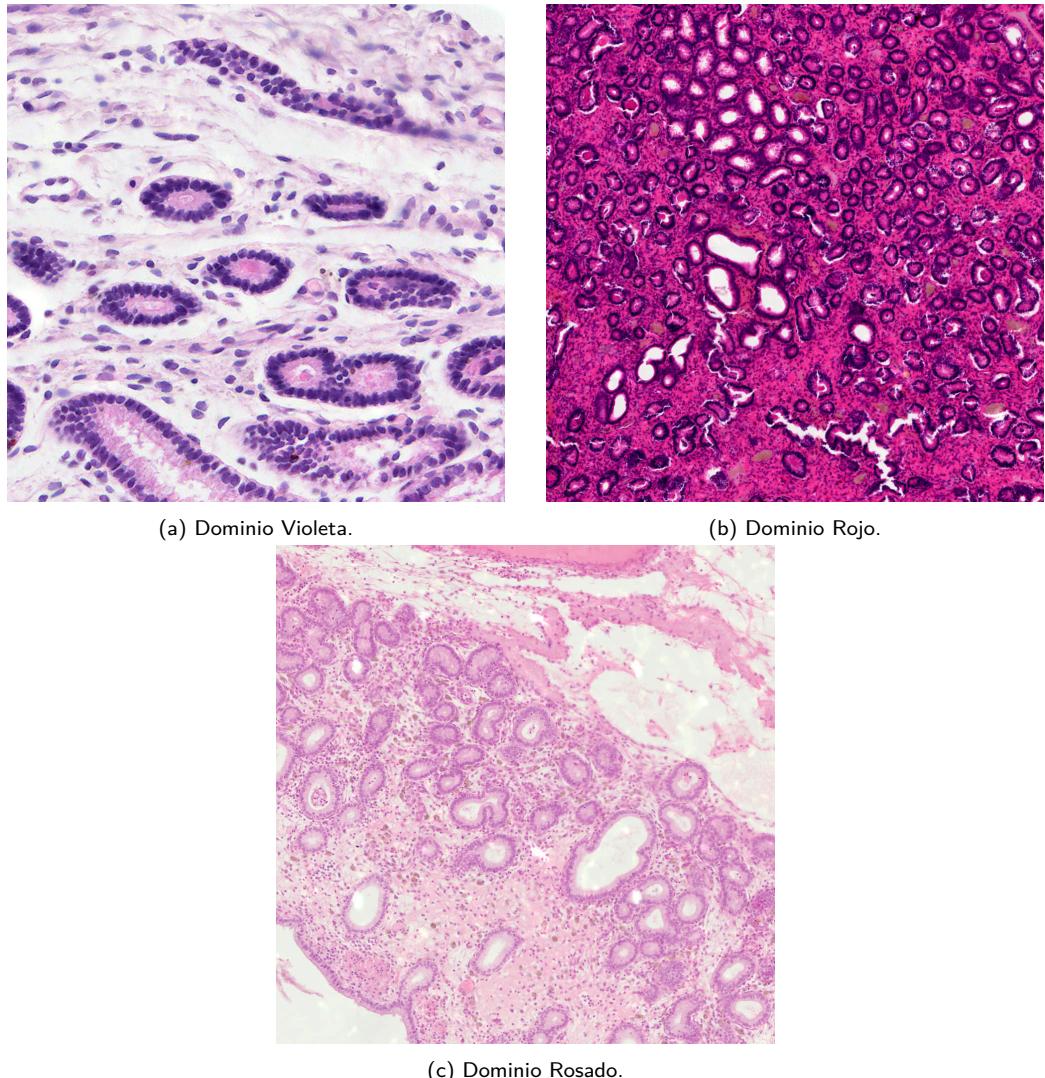


Figura 2.4: Recortes de WSIs que muestran los tres dominios presentes en el conjunto de imágenes.

mismas. La primera consiste en la presencia de un dominio no representativo dentro del contexto del problema, y la segunda se trata del desbalance entre imágenes de distintos dominios. Ambas serán explicadas a continuación.

En cuanto a la primer situación, el dominio rosado no conforma la normalidad de las imágenes que se pretende estudiar, debido al desperfecto en la técnica de coloración. Por este motivo se buscó una técnica de normalización de color para poder incluirlas sin afectar la distribución normal de los datos.

Por otro lado, si se consideraran las imágenes más típicas del problema a tratar, es decir las pertenecientes al dominio rojo o violeta, aún así se observan diferencias en la coloración de las estructuras a detectar. El desbalance de datos de uno u otro dominio podría confundir al modelo y/o conducirlo hacia una solución equivocada,

Capítulo 2. Imágenes y técnicas de preprocesamiento

por ejemplo asociando ciertos tonos a patologías particulares.

A pesar de que los problemas mencionados surgen por causas distintas, la estrategia propuesta para mitigar sus efectos es la misma: aplicar una técnica de normalización de color. La técnica puede ser aplicada de diferentes formas dependiendo si el objetivo es aumentar la cantidad de datos de algún dominio o llevar todas las imágenes a un dominio común. Previo a su implementación se estudiaron varios algoritmos para seleccionar aquel que mejor se adaptara a las WSIs disponibles.

2.2.1. Investigación de algoritmos

Se estudiaron algunos de los algoritmos disponibles que implementan métodos de normalización de color y presentan código abierto. En total cuatro algoritmos fueron puestos en práctica; tres de ellos aplican métodos tradicionales de procesamiento de imágenes y el restante aplica aprendizaje automático. A continuación se presenta un breve resumen de cada uno de estos.

Algoritmo de Macenko et al. (2009) [21]

Este método parte de una imagen de referencia y el objetivo es hallar lo que denominan los vectores de tinción. Estos vectores capturan la intensidad de cada tipo de tinte expresada en función de los canales rojo, verde y azul (RGB). Posteriormente, se realiza una deconvolución entre estos vectores y una imagen que se busca cambiar de dominio. Como resultado se obtiene una imagen por cada tinción, donde cada una representa el aporte de la tinción analizada al resultado global. Para obtener el resultado global se deben sumar todas las imágenes.

La Ley de Beer-Lambert [22] establece una relación entre la intensidad de luz que atraviesa un material y las propiedades del mismo. Resulta oportuno recordar que para observar una muestra en el microscopio, la muestra se encuentra en el portaobjetos y la iluminación se realiza de abajo hacia arriba. Es decir, la luz atraviesa la muestra, y cuanto más tejido con coloración haya, más oscura se percibirá la zona observada.

Si se denomina I a la intensidad de luz detectada luego de atravesar la muestra, I_0 a la intensidad de luz que alcanza la muestra, A a la cantidad de tinción administrada y c al factor de absorción, la relación de Beer-Lambert se puede escribir de la siguiente forma:

$$I_c = I_{0,c} e^{-Ac_c}.$$

El subíndice c indica que esta relación se aplica a cada canal de la imagen. A partir de la fórmula se hace evidente que existe una relación no lineal entre la intensidad de luz percibida por canal y la concentración de tinciones. Por este motivo, el algoritmo de Macenko et al. propone aplicar una transformación para trabajar en un nuevo espacio de color. La transformación aplicada consiste en calcular la Densidad Óptica (OD) haciendo uso de la siguiente fórmula:

2.2. Normalización de color

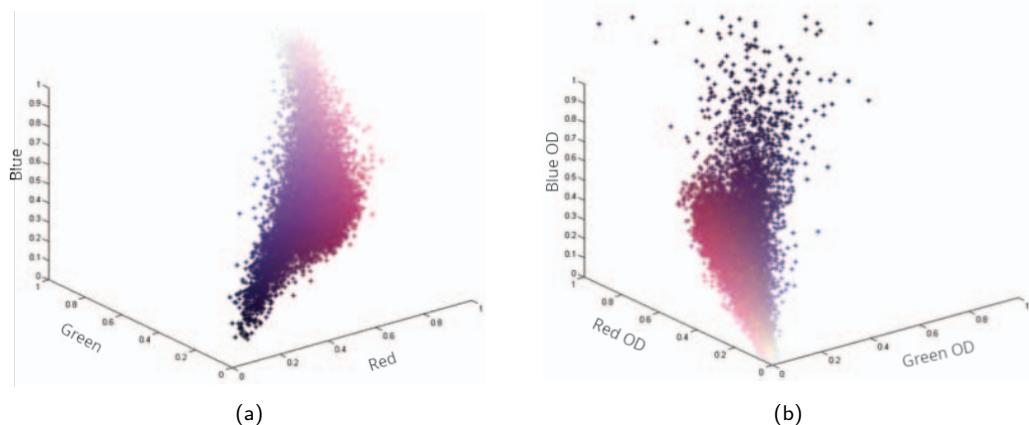


Figura 2.5: A la izquierda se visualiza una gráfica de una imagen del dominio violeta en el espacio RGB y a la derecha en el espacio OD. Se observa que en el espacio OD los píxeles violetas son linealmente separables de los rosados. Imagen adaptada de [21].

$$I_{OD} = -\log_{10}(I/I_0),$$

donde I e I_0 son las mismas que se definieron previamente, y el logaritmo se aplica sobre cada canal del vector resultante. La densidad óptica, también conocida como absorbancia, se puede interpretar como la capacidad del tejido de bloquear la luz.

Luego de aplicar la transformación se continua trabajando en un espacio de tres dimensiones. La principal ventaja que presenta el espacio OD es que existe una relación lineal entre los canales RGB definidos en este espacio y la concentración de tinciones, como se prueba en el trabajo [23]. Esto permite separar linealmente cada tinte. La figura 2.5 presenta una imagen representada en ambos sistemas de coordenadas.

Como paso siguiente se procede a filtrar los píxeles en I_{OD} que presentan valores muy bajos, que corresponden al fondo. Estas regiones dejan atravesar gran parte de la luz que reciben, por lo que presentan valores altos en el vector RGB y se corresponden con valores cercanos a cero de OD. Se suele poner un umbral y filtrar estos píxeles ya que no son de interés para el análisis.

A continuación, se procede a encontrar el plano 2D que presenta la mayor varianza. Para ello se utiliza la descomposición SVD y se toman los dos primeros vectores propios. Luego se proyectan todos los píxeles sobre este plano y se normalizan los valores en el rango $[0, 1]$. Este procedimiento es posible ya que como la imagen presenta dos tinciones principales, no se pierde demasiada información al descartar el tercer vector propio obtenido de la descomposición.

El paso siguiente consiste en hallar el ángulo entre todos los puntos de la imagen en el plano 2D y el primer vector propio indicado por la descomposición SVD. A partir de los ángulos hallados se determina la dirección de los vectores de tinción. Se seleccionan dentro de todos los ángulos obtenidos, aquellos que abarquen del percentil α al $(100 - \alpha)$, con $\alpha \approx 2$, para sortear valores atípicos y hacer más robusto el algoritmo.

Capítulo 2. Imágenes y técnicas de preprocesamiento

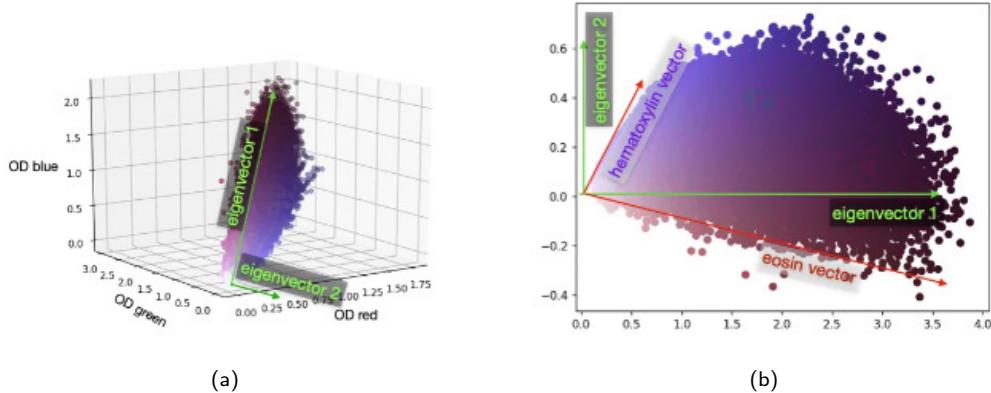


Figura 2.6: En la gráfica de la izquierda en el espacio OD se señalan los 2 vectores propios principales. En la gráfica 2D de la derecha se muestra la proyección de todos los píxeles sobre el plano formado por los 2 vectores propios principales. También se señalan en la imagen los vectores de tinción. Figura adaptada de [24].

Los vectores de tinción reflejan las concentraciones más puras de H y E presentes en la imagen. Los otros tonos que se pueden apreciar se tratan de una combinación lineal de estos vectores, y por lo tanto, todos los píxeles de la imagen quedan comprendidos en el primer cuadrante del sistema de coordenadas conformado por los vectores de tinción. Se puede observar en la figura 2.6 el plano determinado por los dos vectores propios principales, y también se señalan los vectores de tinción hallados.

Por su parte, los vectores de tinción quedan expresados como una combinación lineal de los canales RGB en el espacio OD. Para transformar una imagen de dominio se debe agregar un tercer componente ortogonal a esta base de vectores. Esta componente tendrá la misma dirección que el tercer vector propio obtenido a partir de la descomposición SVD. Los 3 vectores conforman la matriz cuadrada V , que luego se invierte y se multiplica por la imagen a transformar. La ecuación que expresa este resultado es la siguiente:

$$I_{OD} = VS \rightarrow S = V^{-1} I_{OD},$$

donde I_{OD} representa una imagen en el espacio con el mismo nombre, V la matriz conformada por los vectores propios y la tercera componente ortogonal, y S la intensidad de cada tinción. Por último, para obtener la imagen con el aporte de una de las tinciones, basta con multiplicar la componente de S que representa la saturación de dicha tinción, por el vector de tinción correspondiente. Se repite el procedimiento con la otra tinción, y si se desea se pueden sumar ambas imágenes para obtener la transformación de dominio.

Algoritmo de Reinhard et al. (2001) [25].

Este trabajo propone modificar propiedades estadísticas de una imagen por las mismas propiedades obtenidas de una imagen de referencia. La implementación se

2.2. Normalización de color

lleva a cabo en un espacio de color adecuado para aplicar las transformaciones.

Se comienza a partir de una imagen de referencia I_t (*target*), y una imagen a la cual se le va a cambiar el espacio de color para que se parezca al de la de referencia, llamada I_s (*source*). Lo primero que hace es transformar los tres canales RGB de ambas imágenes a un nuevo espacio de color. De acuerdo a los autores existe una fuerte correlación entre canales en el espacio RGB. Es decir, si un píxel tiene un nivel alto en el canal rojo y en el verde, probablemente tenga un nivel alto en el canal azul también. De este modo, se buscó trabajar en un espacio de color donde la correlación entre los canales fuera mínima. Esta característica está presente en el espacio de color $l\alpha\beta$ desarrollada en [26]. La ventaja de que los ejes presenten muy poca correlación entre sí, hace posible que se apliquen distintas operaciones a cada canal sin correr el riesgo de que los otros canales se vean afectados por esto.

La transformación que lleva del espacio RGB al espacio $l\alpha\beta$ requiere de varios pasos intermedios. De todo este proceso vale la pena mencionar que las transformaciones son invertibles, por lo que es posible pasar de un espacio a otro y luego aplicar la transformación inversa para volver al espacio de color inicial.

La técnica de transferencia de color de una imagen a otra tiene lugar en el espacio $l\alpha\beta$. Las transformaciones propuestas imponen la media y la desviación estándar de I_t sobre los píxeles de I_s , para cada canal. Estas operaciones están expresadas por las ecuaciones que se presentan a continuación:

$$\begin{aligned} l &= \frac{\sigma_t^l}{\sigma_s^l}(l_s - \mu_s^l) + \mu_t^l, \\ \alpha &= \frac{\sigma_t^\alpha}{\sigma_s^\alpha}(\alpha_s - \mu_s^\alpha) + \mu_t^\alpha, \\ \beta &= \frac{\sigma_t^\beta}{\sigma_s^\beta}(\beta_s - \mu_s^\beta) + \mu_t^\beta. \end{aligned}$$

Algoritmo de Vahadane et al. (2016) [27]

El objetivo de este trabajo es modelar la imagen histológicas por medio de dos matrices, una que captura la apariencia de color, y otra que contiene la información geométrica de las estructuras presentes en la misma. De esta forma, para una imagen dada es posible modificar su matriz de color por la de otra imagen de referencia, alterando únicamente el color mientras la morfología de las estructuras no se ve afectada.

El procedimiento de normalización de color consiste en dos etapas. La primera se trata de un problema de representación de tinciones y densidades de tinciones a nivel de píxel, que se obtiene mediante la técnica de *factorización en matrices dispersas no negativas* [28]. La segunda consiste en una *normalización de color con preservación de estructuras*, y es cuando se transfieren las propiedades de color de una imagen a otra.

El algoritmo se basa en la Ley de Beer-Lambert y aplica la definición de densidad óptica, ambos conceptos fueron introducidos previamente. Antes de continuar con las ecuaciones que formulan el problema se presentan las variables en juego. Se

Capítulo 2. Imágenes y técnicas de preprocesamiento

considera $I \in R^{m \times n}$ como la matriz RGB de intensidades de una imagen, donde $m = 3$ corresponde a los canales RGB, y n corresponde al número de píxeles. I_0 es la intensidad de luz que ilumina la muestra. Sea $W \in R^{m \times r}$ la matriz de apariencia de color, cuyas columnas representan la base de color de cada tinción, por lo que r corresponde a la cantidad de tinciones (en este caso, $r = 2$). $H \in R^{r \times n}$ que corresponde al mapa de densidad de tinciones donde las filas de la matriz representan la concentración de cada tinción.

Haciendo uso de estas variables, la Ley de Beer-Lambert establece que una imagen I puede escribirse de la siguiente forma:

$$I = I_0 \exp(-WH),$$

donde la exponencial se aplica elemento a elemento.

Si tomamos V como la densidad óptica tenemos:

$$V = \log \frac{I_0}{I},$$

donde el logaritmo también se aplica elemento a elemento.

Por lo que combinando las dos ecuaciones anteriores obtenemos:

$$V = WH.$$

La primera etapa del método es entonces la *factorización en matrices dispersas no negativas*. Este procedimiento se basa en los principios de no-negatividad y dispersión para modelar las matrices que representan las estructuras, llamadas *mapas de densidad de tinción*. Estos principios son definidos a continuación:

- No-negatividad: la densidad de tinción puede tomar valores positivos o cero para un píxel dado, pero no puede ser negativa. Un valor de densidad de tinción negativo significaría que la muestra emite luz.
- Dispersión: a cada estructura biológica, como los núcleos o el citoplasma, se le adhieren ambas tinciones que componen la tinta H&E, en mayor o menor medida. Se asume que las proporciones relativas de tinción son fijas para cada estructura para una imagen, y se le denomina tinción efectiva a la combinación de tinciones presente en estas estructuras. De esta forma, cada píxel de la imagen, que representa un tipo de material biológico, se caracteriza por un tono de tinción efectivo.

La factorización se logra por medio de un proceso de optimización, donde a partir de la imagen I se calcula la densidad óptica correspondiente V , y el objetivo es hallar la matriz de apariencia de color W y el mapa de densidades de tinción H . Esta etapa tiene en cuenta los principios vistos con anterioridad para delimitar el rango de las posibles soluciones. Tanto W como H tienen todos sus elementos positivos o nulos, y la dispersión está dada por la discretización de las estructuras biológicas. Es decir, cada estructura tiene una extensión espacial finita, y se caracteriza por una tinción efectiva de acuerdo al material biológico que se trate.

2.2. Normalización de color

La estimación conjunta de ambas matrices se trata de un problema de optimización no convexo, por lo que puede converger a un mínimo local y no así al mínimo global. Este problema se resuelve alternando entre ambas matrices dejando los parámetros de una fija mientras se optimiza la otra.

La siguiente etapa es la *normalización de color con preservación de estructuras*. En esta etapa se busca normalizar la matriz de apariencia de color de una imagen I_s con respecto a la de una imagen de referencia I_t . Se aplica el paso anterior y se descompone V_s en $W_s H_s$ y V_t en $W_t H_t$. A continuación una versión escalada de H_s se combina con W_t , y a partir de ello se logra preservar la estructura contemplada en la matriz H_s modificando la apariencia de color de la imagen I_s por medio de W_t .

HistAuGAN (Wagner et al. 2021) [29]

Se trata de un algoritmo de normalización de color para imágenes histológicas que reconoce dos aspectos relevantes de las imágenes para tratarlos de forma diferenciada. Por un lado trabaja con un codificador de contenido (E_c), que es invariante ante los distintos dominios y codifica información acerca de la estructura de los tejidos, como por ejemplo los tamaños y las formas. Por otro lado utiliza un codificador de atributos (E_a), que es distinto para cada dominio y aprende las características específicas de color.

Para lograr comprender la lógica del algoritmo es necesario mencionar los conceptos y métodos introducidos en los trabajos anteriores. El primero de estos, y donde se sientan las bases de los modelos generativos que se utilizarán, es el proyecto donde se desarrollan las redes generativas adversarias (GAN) [30]. Le sigue el trabajo de las Cycle-GAN [31] que utiliza dos GANs para obtener un mapeo entre dos dominios. Por último, DRIT++ [32] propone un modelo para generar un mapeo entre dominios, donde a partir de una imagen de entrada perteneciente a uno de los dominios, se pueden obtener múltiples imágenes correspondientes en el otro dominio.

Las GANs fueron introducidas en 2014 y desde entonces han sido un recurso muy frecuente en el área del aprendizaje automático. Consisten en un método de aprendizaje no supervisado que cuenta con dos subredes: un generador y un discriminador. El generador, también llamado modelo generativo, recibe como entrada un vector aleatorio de largo fijo y como salida produce una imagen que trata de imitar a las imágenes con las que fue entrenado. En otras palabras, aprende la distribución de probabilidad de los datos y logra establecer una correspondencia entre un vector que contiene ruido y el dominio de las imágenes de entrenamiento. Por otro lado, el discriminador actúa como un clasificador. El objetivo de este modelo es determinar si la imagen que se le presenta como entrada fue producida por el generador, o si se trata de una imagen real de la base de datos. Esta subred solo se utiliza en la fase de entrenamiento para evaluar que tan bueno es el generador. Después de la etapa de entrenamiento, el discriminador se descarta y el modelo generativo se convierte en el enfoque principal, recreando la relación subyacente entre los vectores del espacio latente y las imágenes del dominio de trabajo.

Posteriormente surgen las GAN condicionales, que se caracterizan por imponer

Capítulo 2. Imágenes y técnicas de preprocesamiento

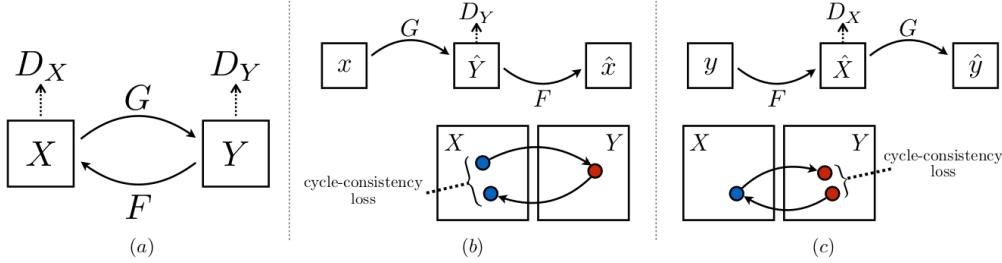


Figura 2.7: Imagen obtenida del informe de Cycle-GAN [31] que muestra los componentes del modelo y explica gráficamente la función de pérdida de consistencia de ciclo. La entrada $x \in X$ se convierte al dominio Y a través de G , para luego ser devuelta a X por la función F , obteniendo \hat{x} . La función de consistencia de ciclo es calculada entre x y \hat{x} . El proceso es análogo para la entrada $y \in Y$.

una condición sobre la imagen a generar. Tanto el generador como el discriminador reciben la condición como un parámetro de entrada adicional. La condición puede ser una etiqueta o una imagen y sirve para dirigir la salida hacia un subdominio del conjunto de datos. A modo de ejemplo, en el trabajo donde se introducen estas redes [33], se entrena una GAN sobre la base de datos de dígitos MNIST [34] y como condición de entrada se proporciona la etiqueta correspondiente a la imagen del número que se quiere generar.

Continuando con la idea del condicionamiento para la generación de imágenes, en 2018 se publica *Image-to-image translation with conditional adversarial networks* [35]. Este trabajo realiza transferencia entre pares de imágenes a partir de GANs condicionadas. El concepto de transferencia entre imágenes se refiere a la capacidad de aplicar el conocimiento adquirido de un dominio de imágenes a otro dominio de imágenes relacionado. En el trabajo se mencionan algunos ejemplos de esta tarea, como puede ser generar imágenes a partir de mapas de bordes, o sketches, o la coloración de imágenes en blanco y negro.

Incrementando la complejidad del problema, en 2020 se introducen las Cycle-GAN para realizar tareas de transferencias de imagen a imagen, entre imágenes sin correspondencia. El objetivo de este trabajo consiste en aprender una relación recíproca entre dos dominios X e Y , dados datos de entrenamiento $x_i \in X$ e $y_i \in Y$. Esto implica dos funciones de mapeo $G : X \rightarrow Y$ y $F : Y \rightarrow X$, y también son necesarios dos discriminadores D_x y D_y . D_x busca diferenciar entre imágenes x e imágenes generadas $F(y)$, de la misma forma que D_y busca diferenciar entre imágenes y y generadas $G(x)$. En la figura 2.7 (a) se muestran gráficamente como interactúan estos componentes.

Además de las funciones de costo características de las GANs, que sirven para penalizar cuando las imágenes producidas por un generador difieren de las imágenes del dominio que están tratando de imitar, este trabajo introduce las funciones de pérdida de consistencia de ciclo. Estas funciones buscan que, para cada imagen x del dominio X , el ciclo de transferencia entre imágenes sea capaz de mapear a x consigo misma. En la figura 2.7 (b) se muestra el proceso completo donde al

2.2. Normalización de color

aplicar el generador G a la imagen x el resultado es \hat{y} , que imita a los elementos del dominio Y . Si luego aplicamos el otro modelo generador F sobre la imagen \hat{y} obtenemos un elemento del dominio X , que deberá parecerse lo más posible a la imagen original x . Lo mismo aplica para las imágenes y del dominio Y , ver figura 2.7 (c). Minimizando estas funciones de costo se logra el cometido del trabajo de generar el mapeo entre los dos dominios principales.

La tercera fuente de referencia es DRIT++; HistAuGAN se basa fuertemente en este trabajo. DRIT++ propone hacer transferencia entre imágenes que no presentan una correspondencia y además plantea la posibilidad de que la salida del algoritmo sean varias imágenes, todas igualmente válidas y pertenecientes al mismo dominio.

Para conseguir este objetivo, este trabajo propone desagregar la información de las imágenes en dos espacios: un espacio común a ambos dominios de entrada, denominado espacio de contenido, y un espacio específico para cada dominio, denominado espacio de atributos. El trabajo cuenta con varias funciones de costo apropiadas para lograr la desagregación de los dos espacios: una correspondiente al espacio de contenido, que evita que los vectores incluyan información propia de los dominios; y otra referida al espacio de atributos, que complementa la tarea del discriminador característico y penaliza las imágenes generadas.

Para comprender la metodología de trabajo propuesta en DRIT++ se puede observar la figura 2.8 que expone una vista general del método. Se parte de dos dominios $\mathcal{X} \subset \mathbb{R}^{H \times W \times 3}$ e $\mathcal{Y} \subset \mathbb{R}^{H \times W \times 3}$, sin correspondencia entre ellos. Se cuenta con dos codificadores de contenido $E_{\mathcal{X}}^c, E_{\mathcal{Y}}^c$, dos codificadores de atributos $E_{\mathcal{X}}^a, E_{\mathcal{Y}}^a$, dos modelos generadores $G_{\mathcal{X}}, G_{\mathcal{Y}}$, y tres discriminadores, uno para cada dominio $D_{\mathcal{X}}, D_{\mathcal{Y}}$, y otro asociado al espacio de contenido D_{adv}^c .

Tomando el dominio \mathcal{X} como ejemplo, el codificador de contenido genera un vector en el espacio de contenido, que es invariante ante dominios: $E_{\mathcal{X}}^c : \mathcal{X} \rightarrow \mathcal{C}$. Por otro lado, el codificador de atributos genera un vector con información propia del dominio $E_{\mathcal{X}}^a : \mathcal{X} \rightarrow \mathcal{A}_{\mathcal{X}}$. Luego el generador $G_{\mathcal{X}}$ toma ambos vectores como entrada y genera una imagen que trata de imitar al dominio \mathcal{X} . A continuación se aplica el discriminador $D_{\mathcal{X}}$ que debe reconocer si la imagen es real o fue generada. A su vez, también participa el discriminador de contenido D_{adv}^c que debe reconocer de qué dominio proviene el vector de contenido generado por el codificador.

La representación desagregada de la información de las imágenes permite realizar transferencias entre estas, combinando el vector de contenido de una de ellas con el vector de atributos de la otra. Para lograr este cometido el trabajo de referencia propuso una función de costo que se denomina consistencia de ciclo cruzado. El primer paso del entrenamiento consiste en tomar dos imágenes, una del dominio \mathcal{X} y otra del dominio \mathcal{Y} , y obtener los respectivos vectores de los espacios de contenido y atributos. Luego, como el espacio de contenido es común a ambos dominios, se intercambian ambas representaciones. Se utilizan ambos modelos generadores para producir nuevas imágenes del dominio \mathcal{X} e \mathcal{Y} , con los vectores de contenido intercambiados. El siguiente paso consiste en volver a utilizar los codificadores y producir nuevos vectores en los espacios de contenidos y atributos, sobre las imágenes generadas. Por último, se vuelven a intercambiar los vectores de contenidos, y

Capítulo 2. Imágenes y técnicas de preprocesamiento

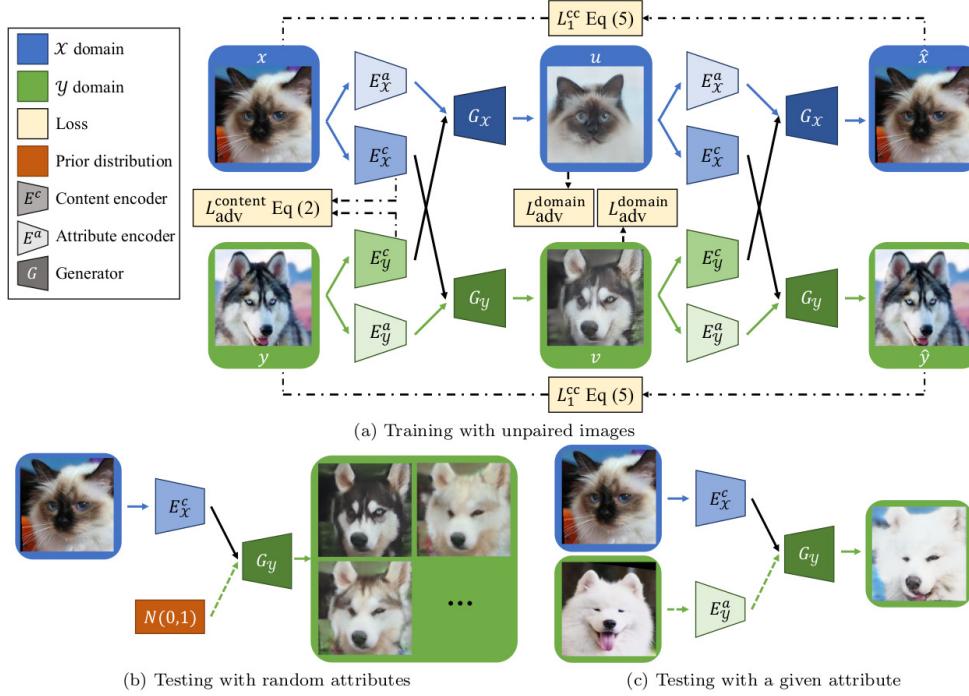


Figura 2.8: Imagen obtenida de [32]. En la sección (a) se expone la metodología propuesta para el entrenamiento. Cada dominio (X, Y) cuenta con un codificador de contenido (E_X^c, E_Y^c) y otro de atributos (E_X^a, E_Y^a). A cada entrada (x, y) se le calcula los vectores de contenido y atributos utilizando los codificadores de su dominio. Luego, el generador de cada dominio (G_X, G_Y) toma como entrada los atributos de su dominio y el contenido del otro, generando imágenes intermedias. En este paso se calculan las funciones de costo de los adversarios de cada dominio (L_{adv}^{domain}). Luego se calculan los vectores de cada imagen generada y se repite el ciclo anterior, donde G_X acepta el vector proveniente de E_X^a , y el vector E_Y^c . Lo análogo se hace para G_Y . Cada generador genera una imagen (\hat{x}, \hat{y}) que, de ser perfecto, sería idéntica a la original. Las secciones (b) y (c) representan distintas formas de utilizar el modelo para hacer inferencia. En (b) se genera una imagen del dominio Y a partir de un vector de contenidos del dominio X , y un vector de atributos aleatorio. Esto da como salida una imagen del dominio Y con el contenido de la entrada. En (c) se genera una imagen del dominio Y a partir del vector de contenido de una imagen del dominio X y el vector de atributos de una imagen del dominio Y .

se generan otros representantes del dominio \mathcal{X} e \mathcal{Y} . El objetivo de esta práctica es ser capaz de generar resultados lo más similares posibles a las imágenes originales de cada dominio. En la figura 2.8 (a) se puede observar la representación esquemática del flujo de trabajo recién descrito.

En la misma figura 2.8 (b) y (c) se presentan dos formas posibles de utilizar el modelo para hacer inferencia. Una opción es partir de una imagen de entrada, de la cual se obtiene el vector de contenido, y sortear de una distribución normal un vector de atributos. La otra forma es utilizar una imagen de cada dominio y extraer de una de ellas la representación de contenido y de la otra el vector de atributos. Como resultado en ambos casos se obtienen imágenes generadas del dominio al

2.2. Normalización de color

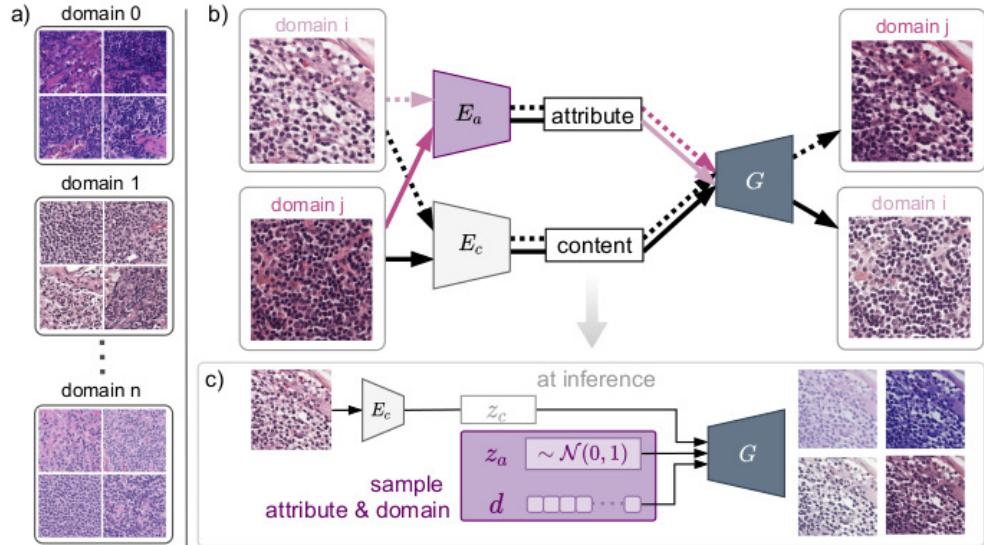


Figura 2.9: Figura obtenida del informe de HistAuGAN [29]. En la sección (a) se muestran ejemplos de imágenes de histopatología con tinción H&E. En la sección (b) se presenta el flujo de trabajo para entrenar el modelo, y en la sección (c) para hacer inferencia.

que pertenece el vector de atributos.

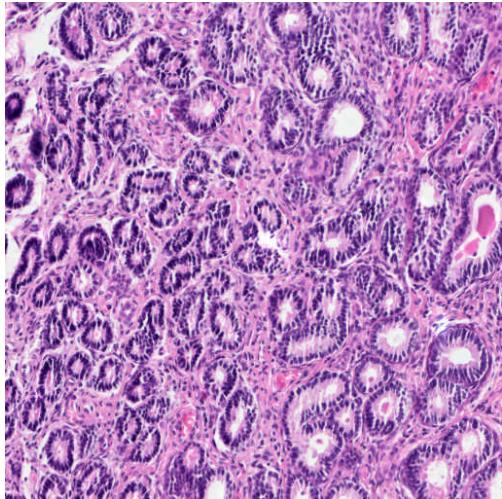
Por último HistAuGAN [29], el algoritmo probado para la tarea de normalización de color, aplica el mismo procedimiento de trabajo introducido por DRIT++ específicamente a imágenes histológicas. Este trabajo propone el flujo que se observa en la figura 2.9. La subfigura (a) muestra los distintos dominios que pueden componer un problema en el área de la histopatología. Luego la subfigura (b) muestra parte del proceso de entrenamiento, y cómo el modelo generador recibe vectores de contenido y atributos de distintos dominios y los combina para producir nuevas imágenes. Finalmente, la subfigura (c) representa el proceso de inferencia. Se obtiene la representación en el espacio de contenido de la imagen de entrada, se muestrea un vector de atributos a partir de la distribución aprendida, y se crea un vector d que determina el dominio de transferencia. Los tres vectores se toman como entrada del modelo generador, que es el encargado de producir una imagen con la morfología de la imagen de entrada y las características de color determinadas por el vector de atributos, de acuerdo al dominio d .

2.2.2. Pruebas realizadas

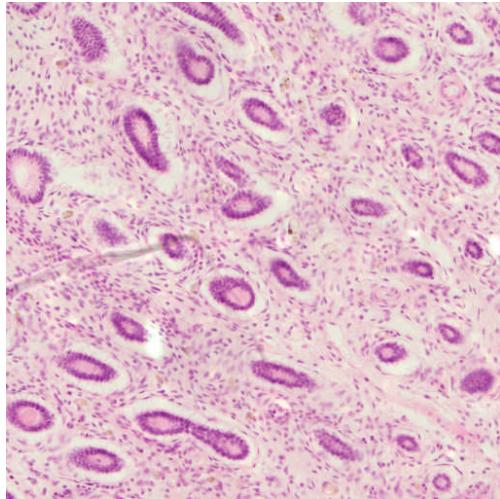
Luego de estudiar en profundidad los algoritmos presentados en la sección anterior, se continuó con las pruebas prácticas. Las pruebas fueron llevadas a cabo con imágenes de la base de datos recopilada para el proyecto. La evaluación de los resultados consistió en la valoración visual de las imágenes obtenidas.

Para los algoritmos de Macenko et al. y Reinhard et al. se decidió emplear el

Capítulo 2. Imágenes y técnicas de preprocesamiento



(a) Imagen de referencia perteneciente al dominio violeta.



(b) Imagen a modificar perteneciente al dominio rosado.

Figura 2.10: Imágenes obtenidas a partir de WSIs recopiladas para el proyecto. Estas imágenes fueron utilizadas para probar los algoritmos de Macenko et al. y Reinhard et al.

código disponible en el repositorio torchstain [36], accesible a través de Github. Como referencia de color se utilizó la imagen 2.10a del dominio violeta, mientras que como imagen a modificar se tomó la imagen 2.10b.

Los resultados obtenidos aplicando Macenko et al. y Reinhard et al. se pueden observar en las figuras 2.11 y 2.12 respectivamente. Es oportuno mencionar que la implementación de Macenko et al. permite desagregar ambas tinciones y observar los aportes de cada una por separado. Con ninguno de estos dos algoritmos se obtuvieron resultados satisfactorios, ya que ninguno alcanzó a replicar los tonos violeta y rosado que predominan en la imagen fuente. Ambos resultados denotan cierta uniformidad o continuidad en la coloración de las imágenes obtenidas, que no se corresponde con la variabilidad en los tonos del dominio de destino.

En cuanto al algoritmo de Vahadane et al., el mismo fue evaluado con la implementación disponible en el repositorio fast-stain-normalization [37], disponible a través de Github. La imagen de destino utilizada, sobre la cual se quería aplicar la normalización, fue la misma que para las pruebas de Macenko et al. y Reinhard et al. Sin embargo, como imagen de referencia se utilizó 2.13a. Esta elección difirió de los casos anteriores, ya que se obtuvieron mejores resultados al utilizar la imagen mencionada en lugar de otras pertenecientes al dominio violeta. El resultado se puede observar en la figura 2.13c.

La implementación de Vahadane et al. testeada parece ser mejor que los métodos anteriores. Los tonos rosado y violeta de la imagen obtenida se aproximan más a los de la imagen de referencia. No obstante, en este ejemplo se pudo apreciar una de las principales carencias de este y los anteriores procedimientos: todos ellos toman una sola imagen como referencia. Esto hace que los algoritmos no logren replicar la diversidad de tonos presentes en una WSI completa, y queden sujetos solo al rango de colores provisto en la referencia.

2.2. Normalización de color

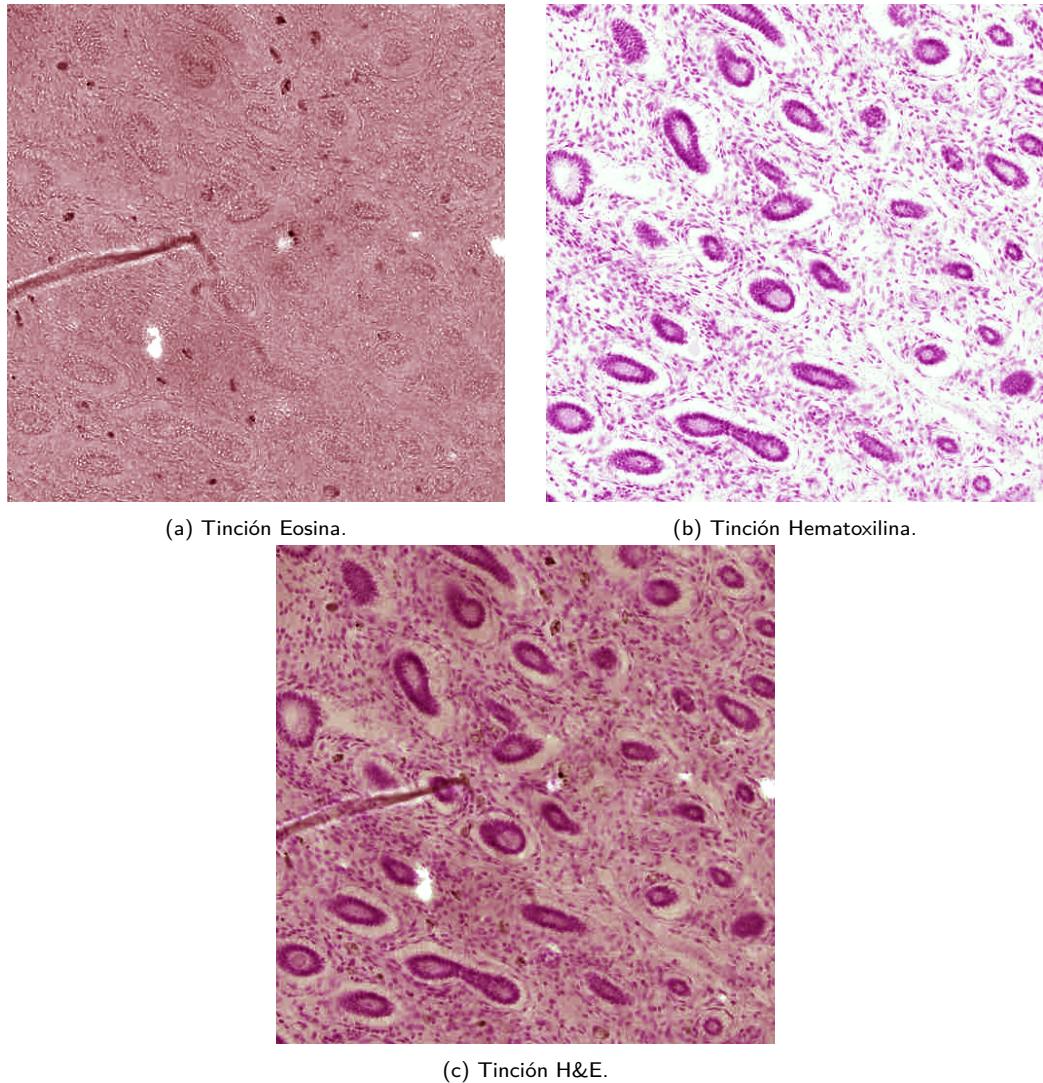


Figura 2.11: Imágenes obtenidas con el algoritmo de Macenko et al. En las figuras (a) y (b) se muestran cada tinción por separado, y la figura (c) es una combinación de las anteriores.

La cuarta implementación evaluada es la presentada por el proyecto HistAuGAN. Este método se diferencia del resto por ser el único algoritmo implementado con aprendizaje automático. La cantidad de imágenes que se utiliza para aprender el espacio de colores hace que el método sea más robusto y tenga mayor diversidad en las tonalidades resultantes.

Otro aspecto que diferencia el modelo de HistAuGAN del resto de los algoritmos es que la aplicación es multi-dominio, lo que implica que a partir de una imagen de entrada se puede obtener la normalización de color para cualquiera de los dominios de trabajo.

Su implementación se puede encontrar en el siguiente repositorio de Github [38], y tiene disponible un modelo pre-entrenado con la base de datos de Ca-

Capítulo 2. Imágenes y técnicas de preprocesamiento

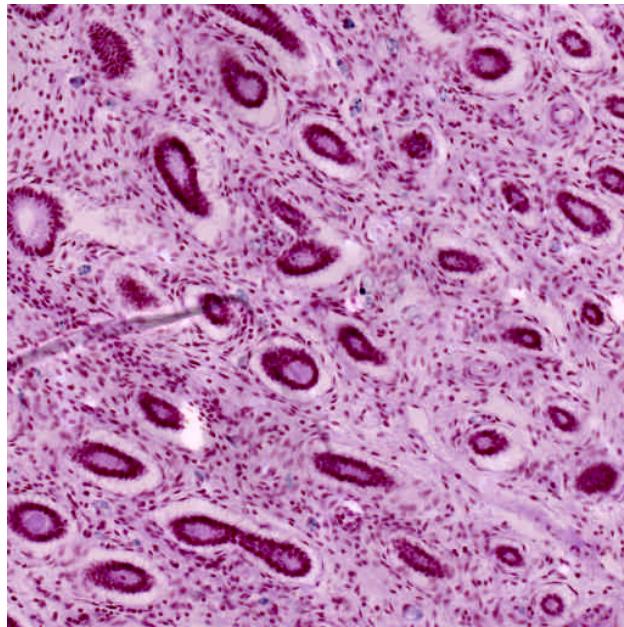


Figura 2.12: Resultado obtenido con el algoritmo de Reinhard et al. Imagen llevada del dominio original (rosado) al dominio de referencia (violeta).

melyon17 [39]. Esta base de datos surge de un desafío que propuso detectar y clasificar metástasis de cáncer de mama en imágenes histológicas de secciones de nodos linfáticos. Las imágenes provienen de cinco centros médicos, por lo que el trabajo cuenta con un total de cinco dominios distintos.

También es posible entrenar un modelo con datos propios. En el caso de estudio se entrenó un modelo por 59 épocas, tomando imágenes de dos dominios de referencia, el dominio rosado y el violeta. Se utilizaron 997 y 994 imágenes de cada dominio respectivamente. En la figura 2.14 se puede ver el resultado de inferencia del modelo entrenado.

Sin dudas el algoritmo de HistAuGAN es el que presenta mejores resultados. Las imágenes obtenidas con este método replican satisfactoriamente el espacio de color del dominio que se quiere imitar y genera salidas realistas. A modo de resumen se incluye la figura 2.15 donde se compara la salida de los cuatro algoritmos. En la sección 5.1 se comentarán los detalles de un nuevo entrenamiento realizado, teniendo en cuenta los tres dominios de trabajo, y cómo se integró la técnica de normalización de color al entrenamiento de los modelos de segmentación.

2.2. Normalización de color

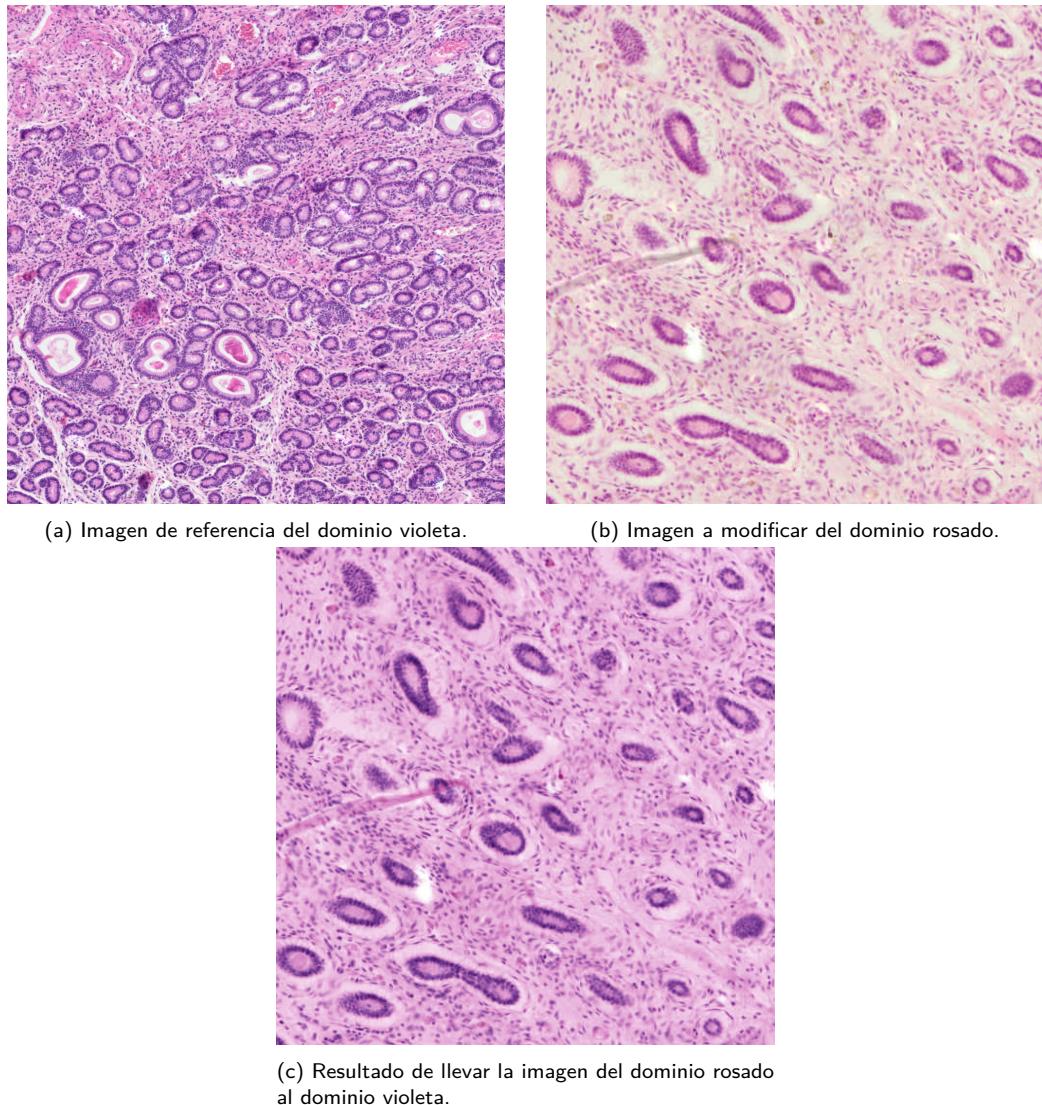
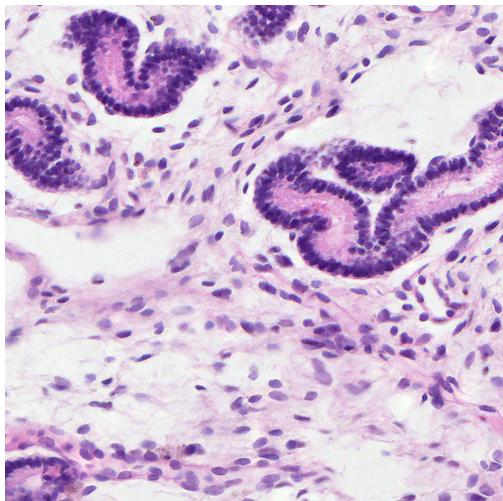
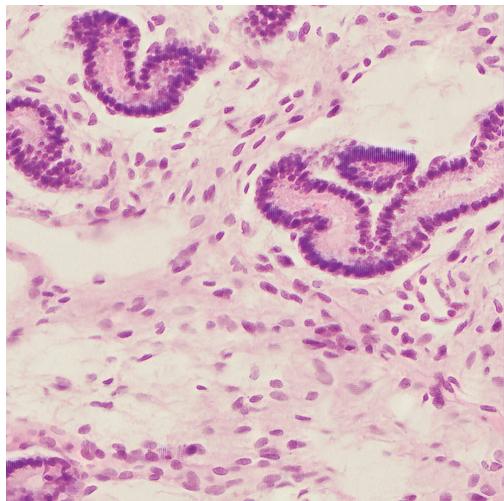


Figura 2.13: Las figuras (a) y (b) se tratan de imágenes obtenidas a partir de WSIs recopiladas para el proyecto. La figura (c) es resultado alcanzado con el algoritmo de Vahadane et al.

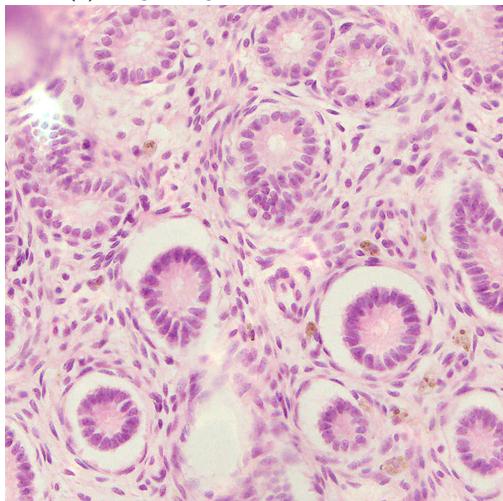
Capítulo 2. Imágenes y técnicas de preprocesamiento



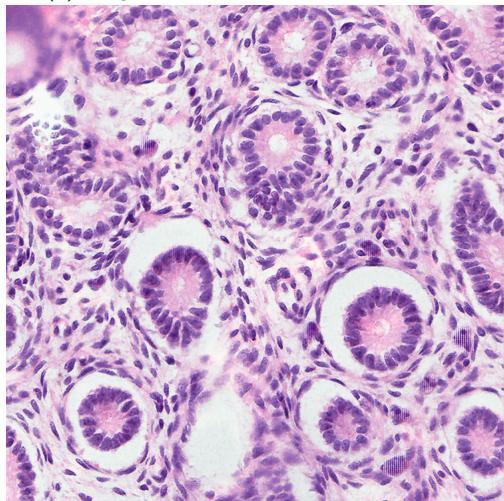
(a) Imagen original del dominio violeta.



(b) Imagen transformada al dominio rosado.



(c) Imagen original del dominio rosado.



(d) Imagen transformada al dominio violeta.

Figura 2.14: Inferencia usando HistAuGAN sobre imágenes de la base de datos.

2.2. Normalización de color

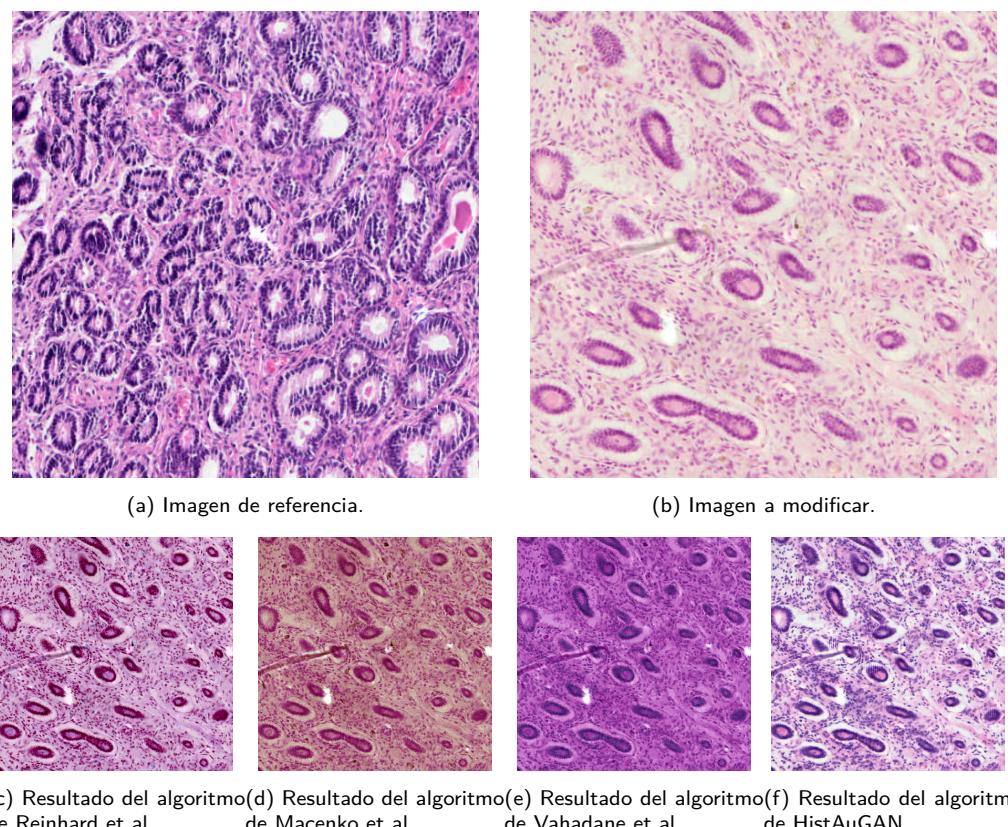


Figura 2.15: Imágenes utilizadas para probar los algoritmos de normalización de color y resultados de cada uno. HistAuGAN obtiene los mejores resultados.

Esta página ha sido intencionalmente dejada en blanco.

Capítulo 3

Segmentación de estructuras

Dentro de todas las estructuras biológicas que quedan comprendidas en una muestra de endometrio de yegua, el trabajo se concentró en tres categorías de interés: núcleos, glándulas y fibrosis. Estas clases fueron elegidas porque permiten identificar en una muestra la presencia de las patologías descritas en la escala de *Kenney-Doig*. Este capítulo describe los modelos de segmentación evaluados para identificar las clases mencionadas. En particular, para la segmentación de fibrosis, como no existía un modelo disponible que segmentara esta patología, se entrenó un modelo con las imágenes disponibles y etiquetas anotadas por el equipo de trabajo.

Tal como se describió en la sección 1.2.2, la endometritis se caracteriza por la acumulación de células del sistema inmunológico. La segmentación y clasificación de núcleos, y un estudio posterior de densidad, permite estudiar si se trata de una acumulación o simplemente una distribución normal.

Por otra parte, la endometrosis se caracteriza por la fibrosis periglandular, los nidos glandulares y la deformación/dilatación de las glándulas. La figura 1.4c muestra estas características. En este caso, tanto la segmentación de fibrosis como la de glándulas adquieren relevancia, porque hacen posible la identificación de nidos o dilataciones glandulares en cada caso.

En lo que respecta a la detección de lagunas linfáticas, otra de las patologías presentes en la clasificación de *Kenny-Doig*, no fue posible generar un modelo para su segmentación, dado que los datos disponibles no contaban con suficientes muestras de dicha anomalía.

3.1. Segmentación de glándulas

Las glándulas presentan gran variabilidad de formas, ya que no solo dependen del corte al momento de tomar la muestra sino que también varían su forma dependiendo del período estral en el que se encuentre la yegua. Como se describió en la sección 1.2.1, en el estro las glándulas se presentan más elongadas y la luz de la glándula es mas reducida, mientras que en el diestro son más circulares, aumenta la luz y sus núcleos son más circulares (ver figura 1.3).

Con el fin de obtener una segmentación de glándulas se probaron dos modelos

Capítulo 3. Segmentación de estructuras

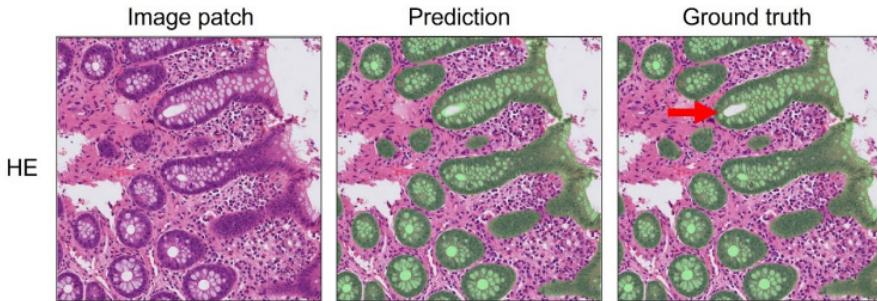


Figura 3.1: **Imágenes de entrenamiento del modelo de epitelio de colon:** A la izquierda, la imagen original recortada. En el centro la predicción del modelo reportado y a la derecha las anotaciones. La flecha roja muestra los límites aproximados para el tamaño mínimo del orificio del túbulo (lleno o no lleno) utilizado durante las anotaciones.

diferentes. Inicialmente la búsqueda de este modelo se basó en encontrar soluciones que resolvieran inferencias en toda una WSI y no únicamente en imágenes. Es importante resaltar que todos los modelos resuelven la inferencia a nivel de imagen, sin embargo no todos están integrados a herramientas para reconstruir la inferencia en una WSI. El tener la inferencia para toda la WSI permite reconstruir un archivo de tipo *.tiff*, que no solo sirve para una visualización global de toda la muestra sino que también da la posibilidad de recortar la segmentación a distintos tamaños. Por ejemplo, hacer inferencia con un modelo con cierta magnificación, ensamblar todas las inferencias en un archivo *.tiff*, y luego recortar el archivo a otro nivel de magnificación.

El primer modelo considerado fue el segmentador de epitelio en muestras mucosa de colon desarrollado en el trabajo [40], obtenido del repositorio de modelos de FastPathology [20]. Estas muestras fueron extraídas en el Hospital Universitario St. Olavs (Noruega) a pacientes con enfermedades inflamatorias intestinales confirmadas o simplemente pacientes con controles sanos. La base de datos completa consistía en 251 WSIs escaneadas a una magnificación de $40\times$. Las imágenes extraídas se recortaron en parches de 2048×2048 píxeles y fueron reescaladas a 512×512 píxeles, es decir que la magnificación resultante de estas imágenes es equivalente a una magnificación de $10\times$. La arquitectura utilizada para este modelo fue una U-Net. En la figura 3.1 se aprecian las anotaciones del modelo junto con la inferencia. Este modelo fue entrenado para segmentar epitelio; para el caso de las imágenes de trabajo esto comprende el epitelio glandular y también el epitelio luminal. Cuenta con la ventaja de estar integrado en FastPathology; es decir, resuelve la inferencia en una WSI completa, y los resultados se pueden visualizar sobre la misma, todo en un solo paso.

Sin embargo, para poder utilizar este modelo fue necesario realizar cierto procesamiento a la segmentación. Observando las inferencias se destacan dos problemas: el primero consiste en llenar la luz en las glándulas más dilatadas y el segundo en eliminar las inferencias en los contornos del tejido.

El primer filtrado de relleno de máscaras se basó en dos pasos. El primero fue una operación morfológica de *closing* que consiste en una dilatación seguido de una

3.1. Segmentación de glándulas

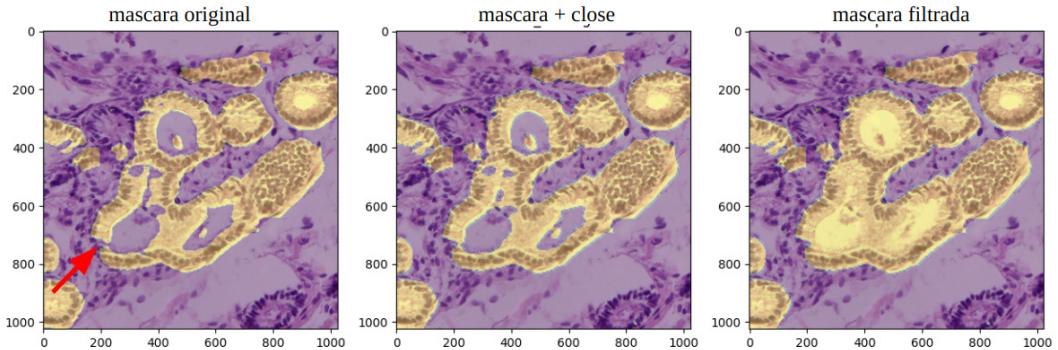


Figura 3.2: Filtro de relleno de huecos: En la imagen de la izquierda se tiene la inferencia cruda del modelo. La flecha roja señala la discontinuidad en el borde de la glándula. En la imagen del centro se tiene la máscara luego del *closing* con los bordes de las glándulas unidos. Mientras que en la imagen de la derecha se tienen las máscaras luego de aplicar el algoritmo de *Flood Fill*.

erosión. Esto ayuda a llenar pequeños agujeros sobre las máscaras, pero su objetivo principal fue conectar bordes de glándulas que no habían sido segmentados. En la figura 3.2 se presenta un ejemplo de ello. Esto es necesario para el siguiente paso donde se aplica el algoritmo de *FloodFill* implementado por OpenCV. Dado un punto en la máscara (en este caso un punto correspondiente al fondo), el algoritmo busca todas las componentes conexas de ese punto llenando el fondo. Una vez obtenida la máscara del fondo se invierte para obtener la máscara de las glándulas.

El segundo filtrado consistió en eliminar las inferencias de epitelio correspondientes al borde del tejido de la muestra. En la Figura 3.3 se representa el flujo de trabajo para este filtrado. Inicialmente fue necesario reconocer en qué parches se encontraban los bordes de la muestra. Umbralizando la imagen original se obtiene una máscara del tejido, y si el porcentaje de fondo es mayor a cierto umbral, se considera que la imagen corresponde a un borde. Posteriormente se llenaron los huecos de la máscara de tejido para obtener una máscara conexa y se aplica una operación morfológica de erosión para reducir los bordes de la máscara de tejido. La máscara erosionada se compara con la máscara de epitelio original como se muestra en la figura 3.3. Por último se aplica un algoritmo de separación de instancias en la máscara de epitelio y se compara con la máscara de tejido erosionada. Se eliminan las instancias de epitelio que no coincide con la máscara de tejido, considerándolas que corresponden a bordes.

A pesar de invertir este esfuerzo en posprocesar las máscaras, las inferencias resultantes no fueron completamente satisfactorias. En la figura 3.4a se observa uno de los problemas del algoritmo de *Flood Fill*, el cual presenta limitaciones al momento de llenar agujeros cuando están sobre el borde la imagen. Esto compromete la caracterización de la forma de las glándulas. Adicionalmente, haciendo una evaluación visual se observa que muchas de las glándulas no son detectadas por el modelo. Sin embargo el problema más significativo radica en la separación

Capítulo 3. Segmentación de estructuras

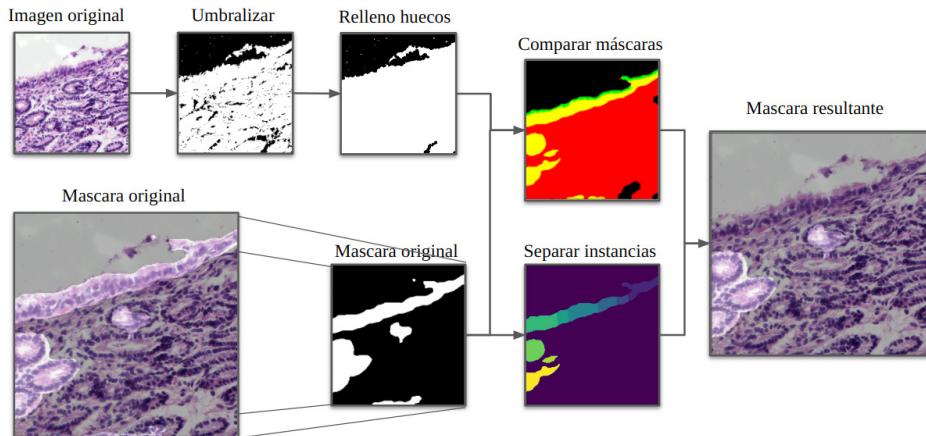


Figura 3.3: **Filtro de eliminación de epitelio luminal:** al comparar las máscaras binarias se coloca el tejido en el canal rojo y el epitelio en el canal verde. Para la visualización se observa que los píxeles donde coinciden ambas máscaras poseen color amarillo (coinciden los canales rojo y verde), mientras que las instancias que no coinciden (debido a la erosión) tienen color verde. Se desea eliminar las instancias de la máscara de epitelio que contienen píxeles de color verde.

en instancias de la máscara. Esto se ejemplifica en la figura 3.4b donde la máscara no es lo suficientemente precisa para determinar correctamente los bordes de las glándulas. Asimismo, al utilizar un algoritmo de *watershed* la separación de instancias es insatisfactoria.

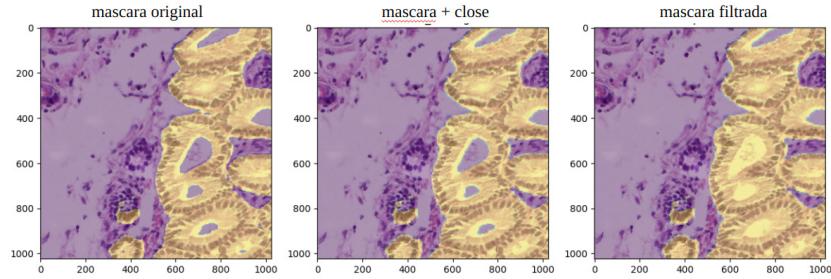
Estos resultados dieron pie a la búsqueda de un modelo específico de glándulas e indagar la posibilidad de introducirlo manualmente en una interfaz que resolviera el manejo de WSIs. En el año 2015 MICCAI¹ lanzó un desafío de segmentación de glándulas en imágenes de histología de colon (GlaS). En el resumen [41] se presenta una descripción de la base de datos junto con los resultados de los mejores equipos. Los datos liberados consisten en 165 imágenes de tinción H&E a una magnificación de 20×. A pesar de que se cuenta con la descripción de los trabajos con mejor desempeño en la competencia, no se tienen implementaciones oficiales de los ganadores del concurso. Sin embargo, se encontró el repositorio [42] con una implementación del desafío, el cual desarrolla una solución utilizando una arquitectura U-Net en PyTorch. En la figura 3.5 se ilustra una comparación entre la inferencia del modelo con las imágenes de GlaS y las imágenes de estudio.

Como este modelo no fue entrenado para ver toda una WSI, se observó que también tenía problemas con imágenes correspondientes al borde del tejido. Es por esto que se decidió mantener el mismo posprocesamiento de las máscaras, tanto para el modelo de epitelio como para el modelo de glándulas.

Con el fin de estudiar ventajas y desventajas de cada modelo, en el capítulo 5 se estudia en más profundidad la comparación de cada uno de estos modelos. Quedan ambos disponibles en la implementación del flujo de trabajo para que el usuario pueda elegir cual utilizar dependiendo de su finalidad.

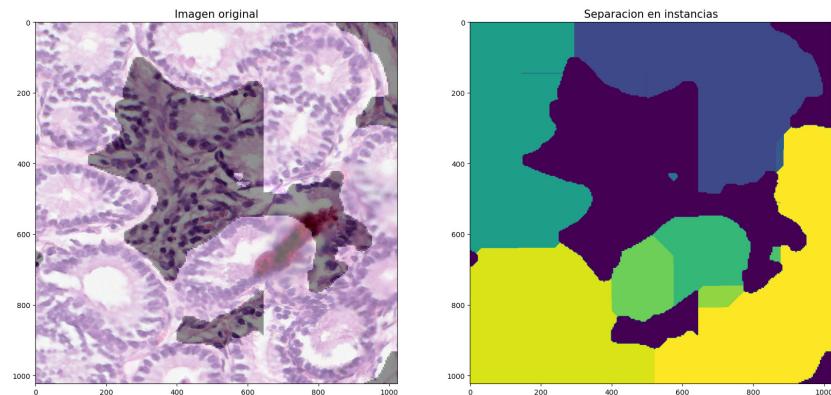
¹Medical Image Computing and Computer Assisted Intervention Society

3.2. Segmentación de núcleos



(a) Errores en el filtro de relleno de huecos. Las máscaras sobre los bordes de la imagen no están cerradas, por lo tanto el modelo no lo detecta.

Separacion en instancias - Mascaras de glandulas



(b) Errores al separar en instancias. Varias glándulas quedan bajo la misma segmentación

Figura 3.4: Resultados no satisfactorios del modelo de segmentación de epitelio de colon.

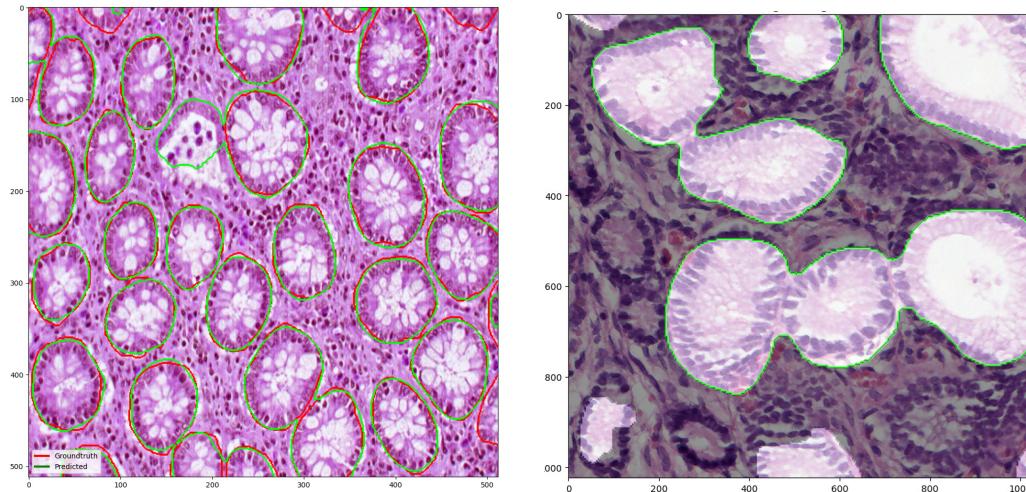
3.2. Segmentación de núcleos

Tal como se comentó previamente, el objetivo de utilizar un modelo de segmentación de núcleos consiste en detectar núcleos de células del sistema inmune. Estas células están presentes en mayor cantidad cuando la muestra tiene infección, lo cual es un indicio de endometritis. Dependiendo del tipo de célula detectada, es posible obtener un indicio del grado de la inflamación. En línea con lo mencionado en la sección 1.2.2, las infecciones pueden ser clasificadas como agudas o crónicas, y las células del sistema inmune predominantes en cada caso son neutrófilos polimorfonucleares o linfocitos, respectivamente.

Los neutrófilos polimorfonucleares se caracterizan por aparecer tener muchos núcleos con formas irregulares, cuando en realidad contiene un único núcleo lobulado conectado por delgados puentes. Por otro lado, los linfocitos se diferencian por presentar un núcleo bien redondo y oscuro. Además, debido a que el núcleo abarca casi la totalidad de la célula, presentan textura uniforme. En la figura 3.6 se puede ver una imagen de referencia de cada una de estas células.

Luego de analizar las imágenes de trabajo, se optó por identificar únicamente linfocitos. Se consideró que el nivel de magnificación no era suficiente para observar con detalle características de los núcleos, y se prefirió observar rasgos más generales

Capítulo 3. Segmentación de estructuras



(a) Inferencia en las imágenes de colon de GlaS.

(b) Inferencia en las imágenes de útero de yegua.

Figura 3.5: Inferencias con el modelo de segmentación de glándulas. A pesar de que las imágenes de entrenamiento de este modelo provienen de tejidos distintos al caso de estudio, se observa la similitud entre ambos.



(a) Neutrófilos Polimorfonucleares. Se caracterizan por la forma de su núcleo, que parece estar dividido en varias regiones.



(b) Linfocitos. Se caracterizan por su núcleo redondo, de color oscuro y uniforme, que ocupa gran parte de la célula.

Figura 3.6: Células del sistema inmune en muestras con tinción H&E. Adaptado de [43].

como la forma y el color.

En una primera instancia la búsqueda de modelos se concentró en aquellos que permitieran identificar linfocitos directamente. Entre los modelos vistos que cumplían este requisito, vale la pena mencionar el trabajo de Shvetsov et al [44]. Este modelo usa la arquitectura propuesta por Graham et al [45], denominada HoVer-Net. Se trata de una red convolucional que segmenta y clasifica núcleos. Comienza con un *backbone* de extracción de características que se inspira en la red Preact-ResNet50 [46] y le siguen tres ramas que cumplen las tareas de determinar

3.2. Segmentación de núcleos

si un píxel corresponde al fondo o un núcleo, separar instancias de núcleos que se encuentren pegadas, y clasificarlos.

El trabajo [44] toma la arquitectura de HoVer-Net y replica sus resultados. También lo entrena con otras bases de datos distintas, y realiza algunas pruebas con datos de distintos conjuntos para ver qué tan bien generaliza el modelo. Además, se evalúa la variación del desempeño ante diferentes técnicas de aumento de datos. Sin embargo, comprueban que si la base de datos de entrenamiento es lo suficientemente grande y los tejidos que la conforman son variados, las técnicas de aumentado de datos no tienen incidencia en los resultados.

Entre las bases de datos probadas se encuentra PanNuke. Se trata de una base de datos creada por Gamper et al [47] que contiene máscaras de segmentación y clasificación de núcleos de múltiples tipos de cáncer, para 19 tipos de tejidos distintos. Provienen de aproximadamente 27 000 WSIs distintas, con diferentes niveles de magnificación, obtenidas de diversas instituciones, y cuentan con un total de unos 216 000 núcleos etiquetados. Las categorías propuestas para la clasificación de los núcleos son las siguientes cinco: células inflamatorias, tejido conectivo, neoplásico, epitelial, y núcleos muertos.

El modelo propuesto por Shvetsov et al. se encuentra disponible en el repositorio de FastPathology. Esto facilita su uso ya que además de contar con los pesos del modelo pre-entrenado, al descargar el mismo también se accede al flujo de trabajo necesario para correr un modelo en este ambiente, donde se definen los parámetros del problema.

La figura 3.7 muestra un ejemplo de la inferencia del modelo sobre una imagen de la base de datos del presente proyecto. Los linfocitos se clasifican dentro de la categoría inflamatorio, conforme a las clases introducidas por la base de datos de PanNuke. De acuerdo con la evaluación de la doctora en veterinaria María José Estradé, y tal como se puede ver en la figura 3.7, la inferencia del modelo no muestra resultados satisfactorios. Los linfocitos no son siempre clasificados como inflamatorios, además de que otros núcleos no relevantes para los objetivos deseados son incluidos dentro de esta clase.

La búsqueda de modelos destinados a la segmentación y clasificación de núcleos no arrojó resultados prometedores. En consecuencia, se prosiguió con la exploración de modelos para segmentación de núcleos únicamente. En este escenario, la estrategia consistió en implementar un procesamiento clásico posterior, destinado a diferenciar los núcleos de los linfocitos respecto a los demás núcleos detectados. La elección de un procesamiento clásico se planteó con el propósito de evitar el etiquetado manual de datos.

El modelo de segmentación de núcleos seleccionado también se encuentra disponible en el repositorio de FastPathology. Se trata de un modelo con arquitectura Lightweight U-Net presentada en el trabajo [48], entrenado con la base de datos de PanNuke, utilizando solamente las máscaras de segmentación. Cuenta con ~ 7.87 millones de parámetros, y su entrada es de 256×256 píxeles y una resolución de $20\times$. El modelo fue implementado por los creadores de FastPathology [20].

En la figura 3.8 se observa una inferencia realizada con el modelo sobre una sección de una WSI de la base de datos recopilada para el proyecto. El modelo pa-

Capítulo 3. Segmentación de estructuras

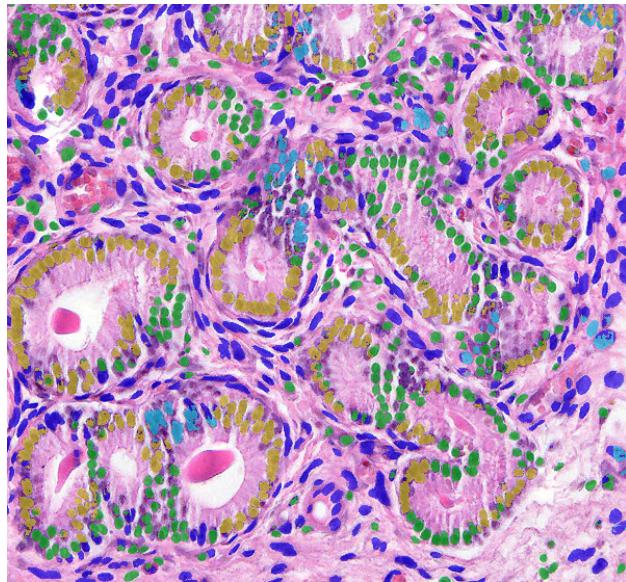


Figura 3.7: Inferencia del modelo de Shvetsov et al. [44] sobre imagen del proyecto. Los núcleos pintados de amarillo se clasifican como células muertas, los azules como células inflamatorias, los verdes como neoplásicos, y los turquesa como epitelial no neoplásicos. Hay una categoría adicional que no aparece en la imagen que son los núcleos del tejido conectivo.

reco detectar un alto porcentaje de los núcleos presentes en la imagen. En términos generales, se concluyó que el modelo tenía un rendimiento aceptable para la tarea propuesta. Como paso siguiente se procuró posprocesar las máscaras obtenidas para reducir la cantidad de núcleos a clasificar.

Para realizar el posprocesamiento se trabajó a nivel de parche y se superpusieron las máscaras de segmentación del modelo de glándulas y del modelo de núcleos. El primer paso consistió en descartar los núcleos de las células que conforman el epitelio de las glándulas. Para ello se superponen ambas máscaras y se descartan los núcleos que queden dentro de una segmentación de glándula.

Luego, como se observó que las máscaras de glándulas no incluían un alto porcentaje de las glándulas presentes en las muestras, y se quería continuar descartando núcleos, se llevó a cabo una segunda etapa de posprocesamiento que realiza lo siguiente: se dilatan los núcleos y se toman las componentes conexas cuya área es mayor a un umbral (se considera que éstas se tratan de núcleos de células que conforman el epitelio de glándulas que no fueron captadas por el modelo). Por este motivo, se descartan los núcleos que quedan comprendidos dentro de las regiones conexas seleccionadas. En la figura 3.9 se pueden ver algunas imágenes ilustrativas que muestran las distintas instancias del posprocesamiento.

3.3. Segmentación de fibrosis

En cuanto a la segmentación de fibrosis, no se encontraron modelos disponibles que realizaran dicha tarea en imágenes con tinción H&E. Es por esto que se optó

3.3. Segmentación de fibrosis

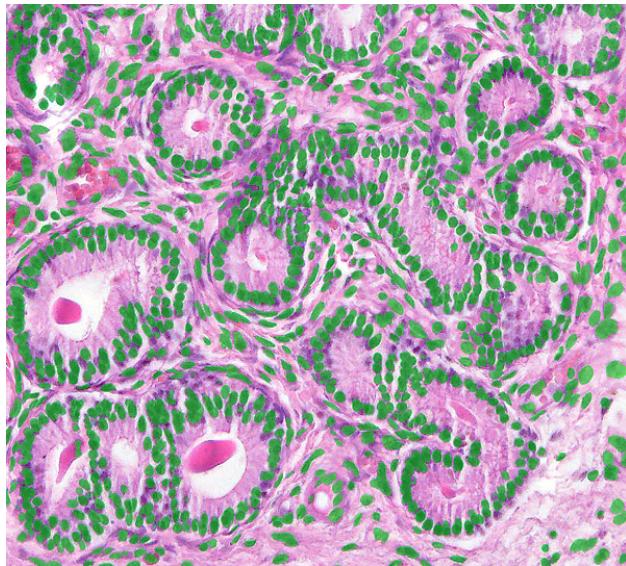


Figura 3.8: Inferencia del modelo de segmentación de núcleos con arquitectura Lightweight U-Net y entrenado con la base de datos de PanNuke, sobre imagen del proyecto.

por entrenar un modelo propio a partir de las WSIs recolectadas, con la ayuda de la herramienta *Quick Annotator* (QA), presentada en la sección 1.5.1.

El *framework* de NoCodeSeg, presentado en la sección 1.5.2, no fue utilizado para entrenar ningún modelo, debido a que la exportación de una máscara de segmentación desde QuPath a otra aplicación requería demasiado tiempo.

Luego de inspeccionar las WSIs disponibles en busca de la anomalía en cuestión, 10 de ellas fueron elegidas para ser etiquetadas.

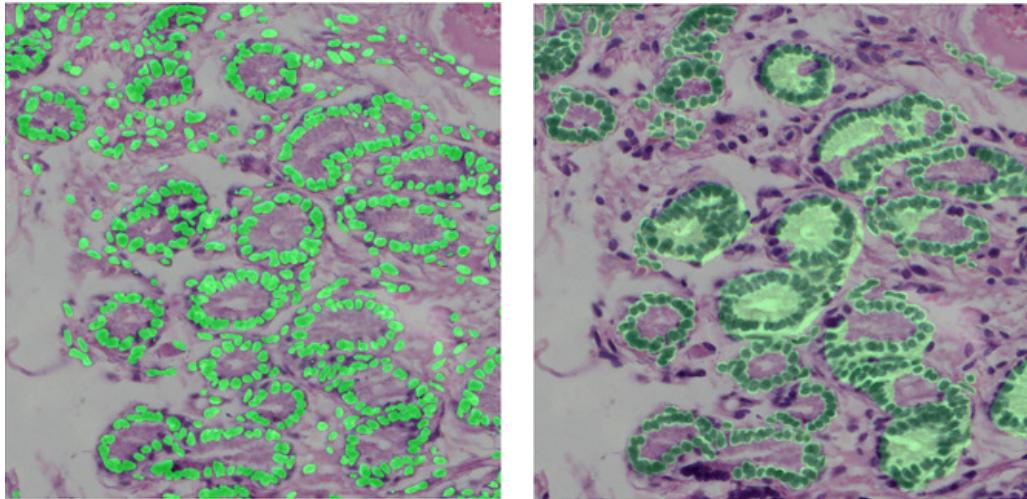
Como se menciona en la sección 1.5.1, el primer paso para usar QA es cortar las WSIs en parches de tamaño y magnificación adecuados, y para ello hay que determinar dichos parámetros. Siguiendo la guía [49] se eligió un tamaño de parche de 1000×1000 píxeles a una magnificación de $10\times$. Esta configuración es tal que permite apreciar nidos de glándulas dilatadas en un solo parche, al tiempo que se distinguen las fibras de colágeno alrededor de las mismas, como se aprecia en la figura 3.10. Estas son características histológicas importantes de la fibrosis, como se vio en la sección 1.4.

Usando la herramienta se obtuvieron parches con las características mencionadas, de las 10 WSIs seleccionadas. Se utilizó además una funcionalidad de QA que descarta los parches en los que no hay tejido; esto baja la cantidad de parches significativamente.

3.3.1. Primer acercamiento

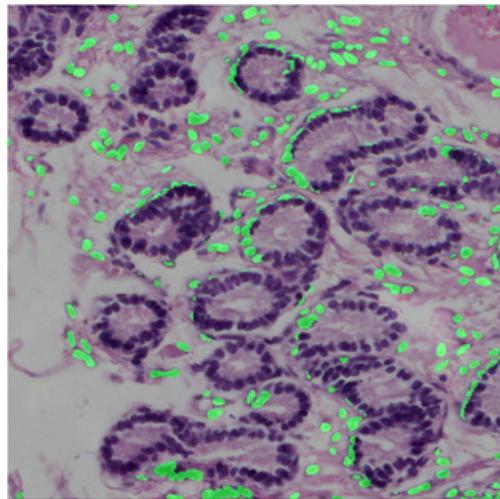
Como se explica en la sección 1.5.1, trabajar con QA es un proceso iterativo, donde se anotan algunas imágenes, se entrena el modelo, y se vuelve a anotar de forma sucesiva. La interfaz permite seleccionar regiones de interés de los parches sobre la cual se genera una máscara de segmentación; a estas secciones las llama-

Capítulo 3. Segmentación de estructuras



(a) Máscara generada por el modelo segmentador de núcleos.

(b) Máscara de glándulas y regiones conexas grandes.



(c) Máscara de segmentación de núcleos posprocesada.

Figura 3.9: Imágenes del proceso de posprocesamiento realizado sobre las máscaras de núcleos. Se eliminan las instancias de núcleos que coinciden con las de glándulas. Por otro lado se dilatan las glándulas y se eliminan las instancias con área mayor a cierto umbral, asumiendo que corresponden a glándulas no detectadas.

remos ROIs (por *region of interest*) siguiendo la nomenclatura de la herramienta. QA guarda una máscara para cada parche combinando las máscaras generadas para cada ROI del mismo. Se utilizaron ROIs de tamaño 512×512 píxeles, pudiendo etiquetar tres clases: positivo, negativo, y *desconocido*. En la figura 3.11 se observa el flujo de QA, donde las etiquetas en color celeste son positivas y las rosadas negativas. Para un mismo parche pueden seleccionarse varias ROIs, incluso ROIs superpuestas.

En el proceso de *active learning* se refinó seis veces el modelo de segmentación, que corresponde a una U-Net [15]. En total se etiquetaron 500 parches, generando

3.3. Segmentación de fibrosis

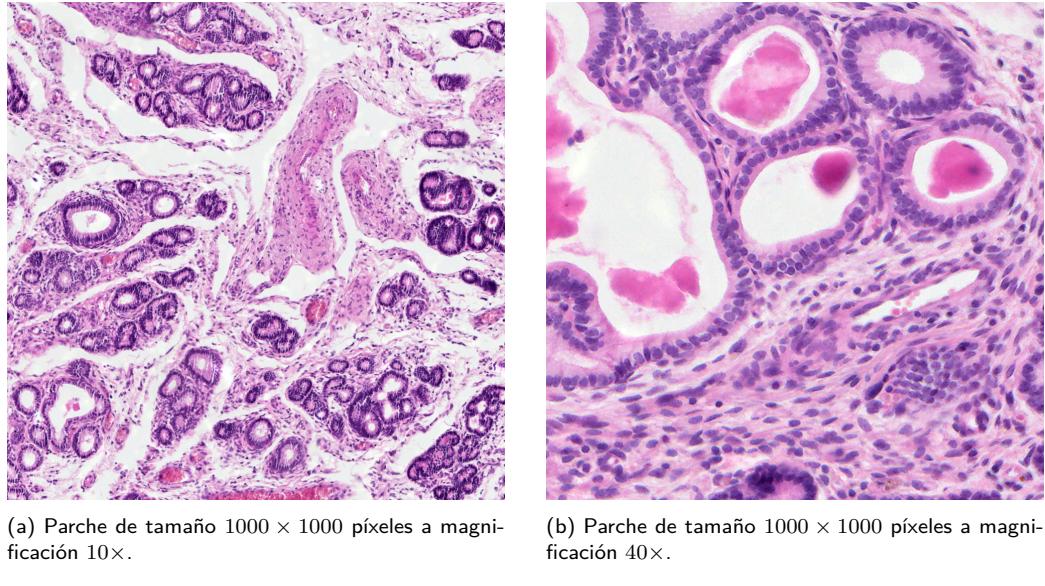


Figura 3.10: Ejemplos de parches de igual tamaño a distintos niveles de magnificación.

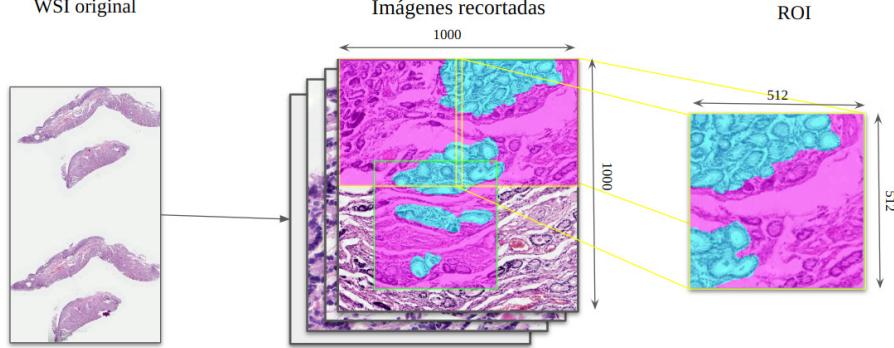


Figura 3.11: Flujo de trabajo para el etiquetado de Fibrosis. A la izquierda se tiene la WSI que es recortada en imágenes de 1000×1000 . En el centro se observa una imagen que tiene tres ROIs etiquetadas, y a la derecha se tiene la ROI seleccionada de tamaño 512×512 .

1023 ROIs, de las cuales 62 % integran el conjunto de entrenamiento y 38 % el conjunto de validación. Estos conjuntos están balanceados en las clases; tienen la misma proporción de píxeles positivos y negativos. La función de costo optimizada durante el entrenamiento es la entropía cruzada ponderada, definida por la ecuación:

$$H(p, q) = \frac{1}{MN} \sum_{m=1}^M \sum_{n=1}^N \sum_{i=0}^1 -\omega_i p_i^{n,m} \log q_i^{n,m}, \quad (3.1)$$

donde, dado el píxel en la posición (n, m) de la imagen, $q_i^{n,m}$ representa la probabilidad inferida por el modelo de que pertenezca a una clase i , y $p_i^{n,m}$ la probabilidad real de que pertenezca a esa clase (obtenida de la etiqueta). M y N son las dimensiones de la entrada. En la función de costo se ignoran los píxeles de la clase

Capítulo 3. Segmentación de estructuras

desconocido, por lo que este es un problema binario donde tenemos dos clases complementarias: la clase positiva (presenta fibrosis), y la clase negativa (no presenta fibrosis).

Como la fibrosis no está presente en todas las imágenes del *dataset*, existe un gran desbalance entre la clase positiva y la negativa. Es por esto que QA calcula pesos para cada clase, de forma de balancear su aporte a la función de costo. Este parámetro se actualiza en cada iteración del proceso de *active learning* según las fórmulas siguientes:

$$\begin{aligned}\omega_1 &= 1 - \frac{\text{Ppix}}{\text{Ppix} + \text{Npix}} = \frac{\text{Npix}}{\text{Ppix} + \text{Npix}}, \\ \omega_0 &= 1 - \omega_1,\end{aligned}\tag{3.2}$$

donde Ppix corresponde a la cantidad de píxeles positivos en todo el *dataset* y Npix a la cantidad de píxeles negativos. Nótese que a lo largo de las iteraciones estos valores van cambiando a medida que se agregan nuevas etiquetas. Finalizado el etiquetado los pesos utilizados fueron $\omega_1 = 0.940$ y $\omega_0 = 0.060$.

Por defecto, QA aplica las siguientes transformaciones a los datos de entrada: rotaciones, escalados, cambios en el contraste, cambios en los niveles de intensidad, *blurring* (filtrado pasabajos), y adición de ruido. Se aplican a cada ROI antes de entrar a la red, con cierta probabilidad de ser aplicadas. Además, cada ROI se submuestra a tamaño 256×256 píxeles al entrar a la red. Algunos ejemplos de aumentaciones de datos se muestran en la figura 3.12.

En la figura 3.13 se observa, para cada iteración, la evolución a lo largo de las épocas de las funciones de costo en entrenamiento, junto con la métrica de *accuracy*. Es interesante observar cómo luego de las primeras dos iteraciones el modelo tiende a estancarse y no tener grandes mejorías. Más aún, la función de costo de la última iteración (6_Apr28_03-04-44), es superior a las iteraciones anteriores. Ésto se debe a que, para esta ultima iteración se introdujeron imágenes de la segunda serie de WSIs (imágenes con distinta coloración a las originales).

Para evaluar el modelo generado se reservaron 57 parches (11 % de todo el *dataset*) y se reportaron las métricas más utilizadas para evaluaciones de modelos de segmentación. Estas métricas son la IOU (Intersection Over Union y el coeficiente de Sorenson-Dice:

$$IOU = \frac{TP}{TP + FP + FN}\tag{3.3}$$

$$DC = \frac{2TP}{2TP + FP + FN},\tag{3.4}$$

donde *true positives* (TP) se refiere a los píxeles acertados en la predicción, *false positives* (FP) son los píxeles erróneamente etiquetados como positivos, y *false negatives* (FN) los erróneamente etiquetados como negativos. Ambas métricas están comprendidas entre 0 y 1, y un valor más cercano a 1 significa que la predicción de la clase positiva se asemeja más a la etiqueta. En el Anexo A.1 se detalla más información sobre estos entrenamientos, junto con otras métricas de evaluación.

3.3. Segmentación de fibrosis

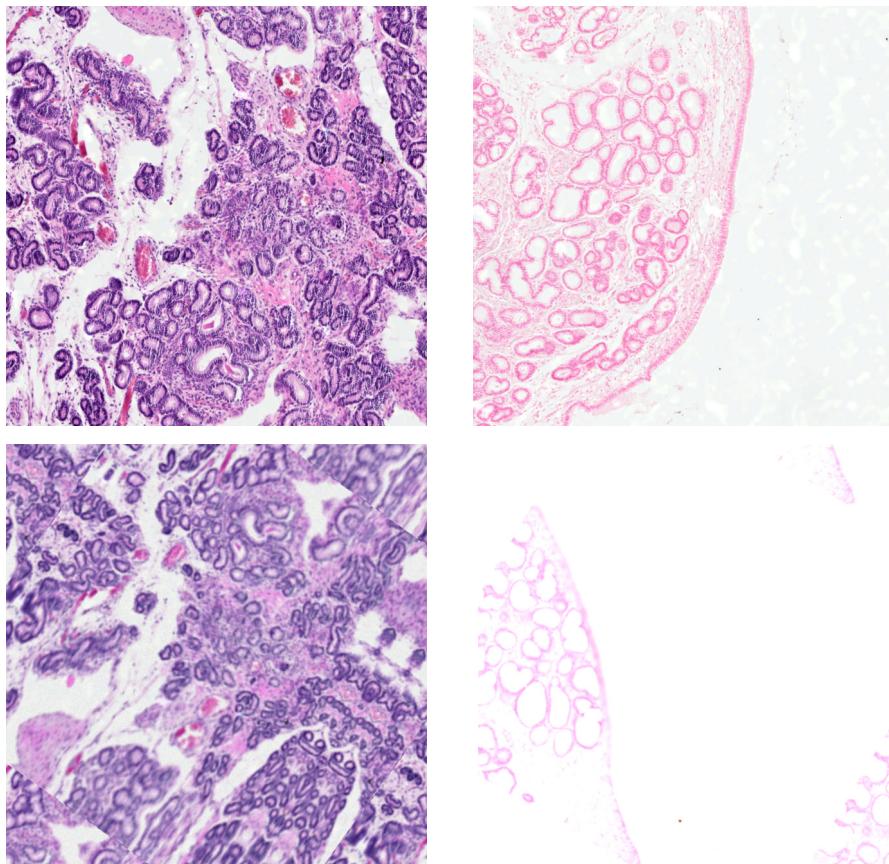


Figura 3.12: Ejemplos de las aumentaciones de datos de QuickAnnotator. En la fila de arriba se observan las imágenes originales. En la de abajo, las transformadas. Se puede ver como la técnica de manejo de bordes, el *blur*, y las modificaciones sobre el color de las imágenes hacen que se pierda detalle de las estructuras, valioso a la hora de entrenar.

Para todas las iteraciones del modelo se reportan las métricas mencionadas para el conjunto de test en la tabla 3.1. Se observa que para las últimas iteraciones el modelo mejora de forma moderada, sin embargo siguen siendo muy bajas.

Iteración	IOU	Coeficiente de Sorenson-Dice
1	0.080	0.133
2	0.161	0.241
3	0.194	0.277
4	0.175	0.261
5	0.202	0.288
6	0.217	0.311

Tabla 3.1: Métricas de iteraciones del modelo de segmentación de fibrosis con el *framework* de QA al momento de generar el *dataset*.

Capítulo 3. Segmentación de estructuras

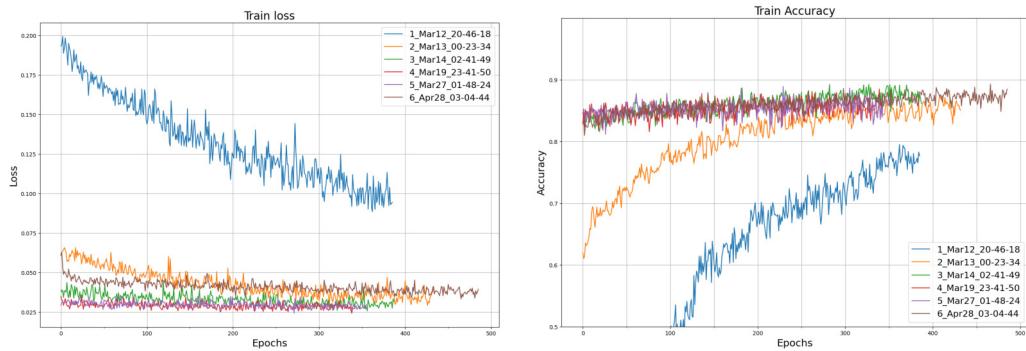


Figura 3.13: Métricas de los entrenamientos correspondientes al proceso de *active learning* realizados con QuickAnnotator. A la izquierda se gráfica la entropía cruzada sobre el conjunto de entrenamiento. A la derecha se presenta el *accuracy* sobre el mismo conjunto.

3.3.2. Problemas con Quick Annotator

Los resultados expuestos en la sección anterior indican que el modelo de segmentación de QA, entrenado de esa manera, no logra resolver la tarea de manera satisfactoria. Fue necesario indagar en la implementación del proyecto y hacer modificaciones para intentar mejorar los resultados. En esta sección se presentan las medidas adoptadas para resolver esta situación.

A partir de este punto ya no se etiquetaron parches adicionales; la base de datos de fibrosis se conforma por las imágenes etiquetadas hasta el momento. Además se dejó de lado el *framework* (interfaz gráfica, *backend*, base de datos) y se entrenó únicamente con el código de entrenamiento de QA.

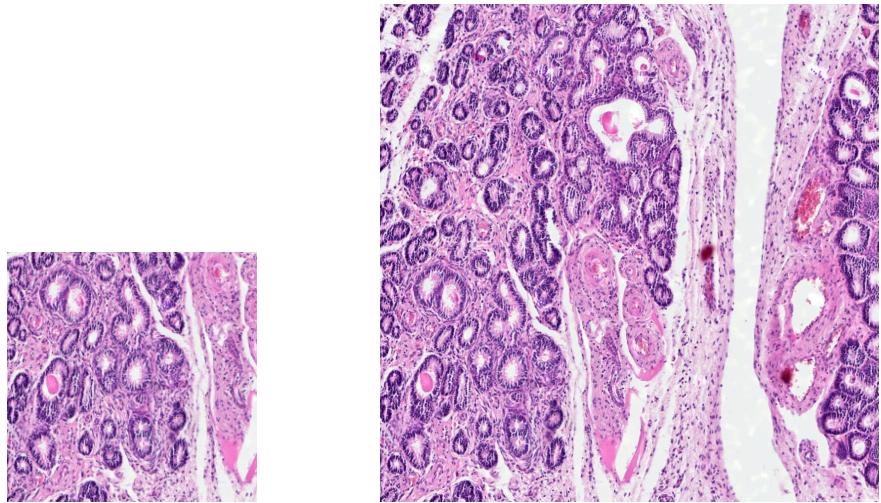
Al modificar el entrenamiento, los requerimientos computacionales para llevarlo a cabo incrementaron, y los recursos computacionales personales ya no eran suficientes. Fue necesario ejecutar los entrenamientos en el Centro Nacional de Supercomputación (ClusterUY) [50]. En el Anexo B se detalla la configuración del ambiente necesario para realizar entrenamientos en ClusterUY.

Tamaño de entrada al modelo

El primer cambio realizado tuvo que ver con el tamaño de las imágenes de entrenamiento. Tal como se desarrolla en la sección 3.3.1, el tamaño de las imágenes de entrada al modelo es de 256×256 píxeles. Ésto se debe a que, por defecto, cada ROI es submuestreada a este tamaño antes de entrar a la red. En base a los criterios mencionados para elegir el tamaño del parche (ver sección 3.3), un nuevo proyecto de QA fue creado para entrenar con los parches de 1000×1000 píxeles. En el anexo A.2 se describen las modificaciones necesarias para ejecutar estos entrenamientos. En la figura 3.3.2 se comparan los distintos tamaños de entrada.

Siguiendo los pasos de QA, primero fue entrenado el *autoencoder* (modelo base, ver sección 1.5.1). La figura 3.15 muestra el error cuadrático medio entre las imágenes del conjunto de entrenamiento y las imágenes generadas por el modelo, promediado por época. Al ser un entrenamiento no-supervisado, no existe conjunto

3.3. Segmentación de fibrosis



(a) Ejemplo de entrada a la red con tamaño por defecto (256×256). (b) Ejemplo de entrada a la red con nuevo tamaño (1000×1000).

Figura 3.14: Comparación entre tamaños de imágenes de entrenamiento de la red de QuickAnnotator.

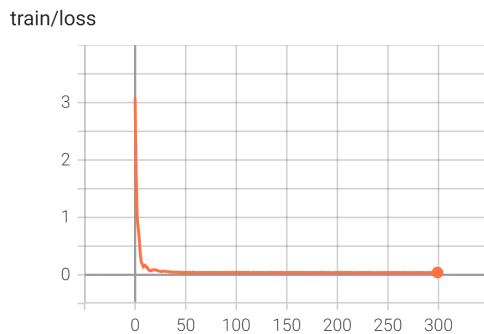


Figura 3.15: **Experimento de tamaño de imagen:** Error cuadrático medio entre las imágenes de entrenamiento del *autoencoder*, promediado por época.

de validación ni de prueba. Se puede ver que en pocas épocas el modelo aprende a reconstruir las imágenes de entrada de manera adecuada.

Luego fue entrenado el modelo de segmentación, inicializando sus pesos con los del modelo base. Las métricas de este experimento, reportadas en la figura 3.16, muestran que mientras el modelo tiene buenos resultados sobre el conjunto de entrenamiento, su desempeño es pobre sobre el conjunto de validación.

En la figura 3.16b se puede ver que la función de costo en validación diverge alrededor de la época 25. Además se ve que el *True Positive Rate* (TPR), es decir, la cantidad de píxeles verdaderos positivos en relación al total de píxeles, tiene gran variación entre iteraciones.

La divergencia de la entropía cruzada puede deberse a varios factores, por ejemplo: el gran desbalance de clases presente en el *dataset*, el aumentado de datos, o la inicialización de los pesos de la red. Por este motivo se optó por simplificar el en-

Capítulo 3. Segmentación de estructuras

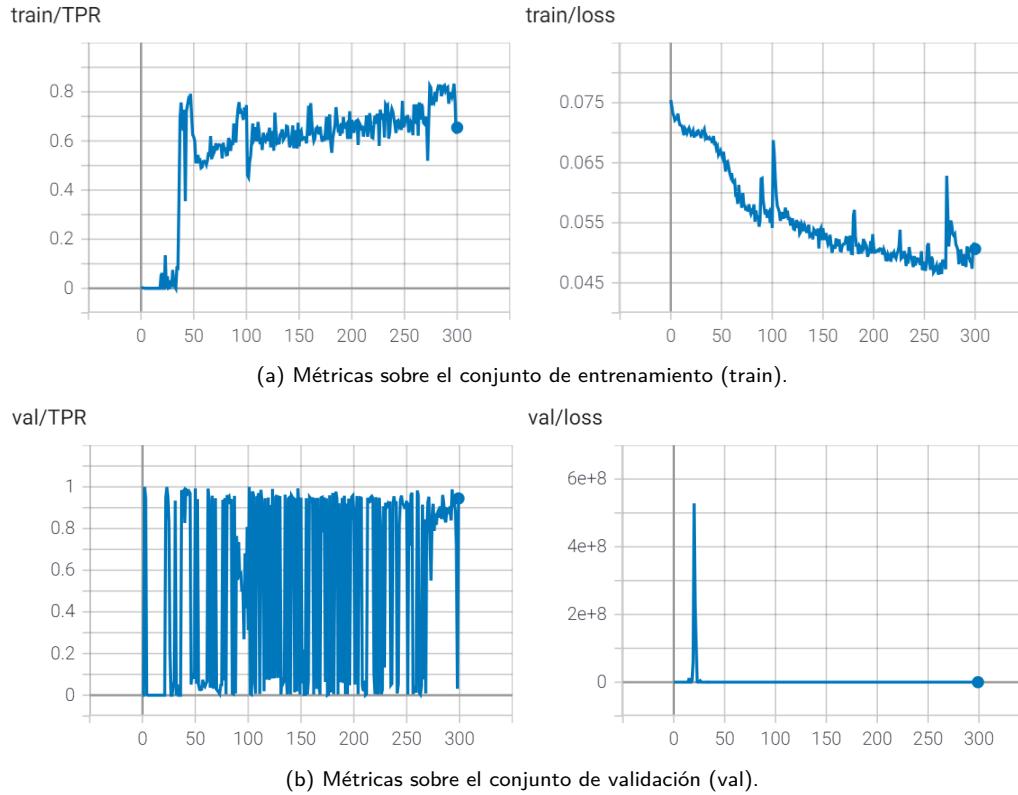


Figura 3.16: **Experimento de tamaño de imagen:** Entropía cruzada ponderada (*loss*) y relación de verdaderos positivos (TPR), en función del numero de épocas. La gran variación del TPR entre épocas, junto con la divergencia de la *loss* en validación, que se ven en la figura 3.16b, revelan un problema con el entrenamiento.

trenamiento y afectar de a una estas áreas, haciendo experimentos y respondiendo en base a los resultados.

Durante los experimentos subsecuentes será entrenada únicamente la red de segmentación, tomando como inicialización el modelo base obtenido. Es decir, todos los experimentos a continuación comparten los pesos iniciales, a no ser que se explice lo contrario. El tamaño de los datos de entrenamiento se mantuvo de 1000×1000 píxeles.

Función de costo

El siguiente de los cambios realizados fue la función de costo. El gran desbalance de clases del *dataset* hace que sea favorable para la red predecir gran parte de la salida como clase negativa. Es por esto que fue cambiada la función de costo del modelo por el coeficiente de Sorenson-Dice (DC), definido en la ecuación 3.4. Este coeficiente se calcula en base a la clase positiva exclusivamente, por lo que es favorable en este contexto. Esta función de costo corresponde a $1 - DC$ ya que el objetivo del entrenamiento es minimizarla, a diferencia de lo que se buscaría usando el coeficiente de Dice.

3.3. Segmentación de fibrosis

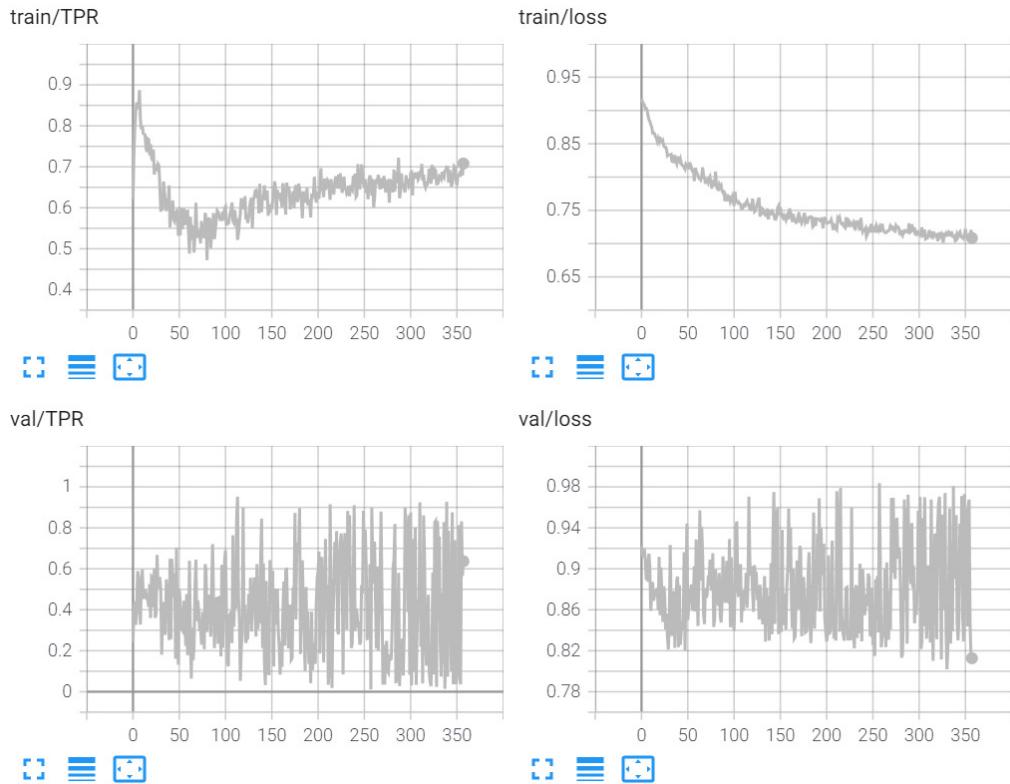


Figura 3.17: **Experimento de función de costo:** Función de costo de Dice (loss) y TPR sobre conjuntos de entrenamiento (train) y validación (val), en función de las épocas.

En las métricas de la figura 3.17 de este experimento se ve que la función de costo sobre el conjunto de entrenamiento (train/loss) no presenta grandes mejorías respecto al valor inicial, por más que la tendencia es decreciente. Sobre el conjunto de validación se observa gran variabilidad (val/loss), al igual que utilizando la función de entropía cruzada. Como este cambio no introdujo mejoras en los resultados, fue descartado.

La variabilidad en la función de costo sugiere que el modelo no logra ajustarse a los datos, lo que dio lugar a revisar las transformaciones de datos aplicadas por QA.

Aumentado de datos

Por defecto, QA aplica las transformaciones mencionadas en la sección 3.3.1. En la figura 3.12 se pueden ver ejemplos de cómo estas transformaciones introducen artefactos que no son naturales del problema, como bordes negros en medio del tejido, o pérdida de detalle de las estructuras. Para evitar este tipo de alteraciones se optó por utilizar únicamente transformaciones de rotación múltiplo de 90 grados.

Capítulo 3. Segmentación de estructuras

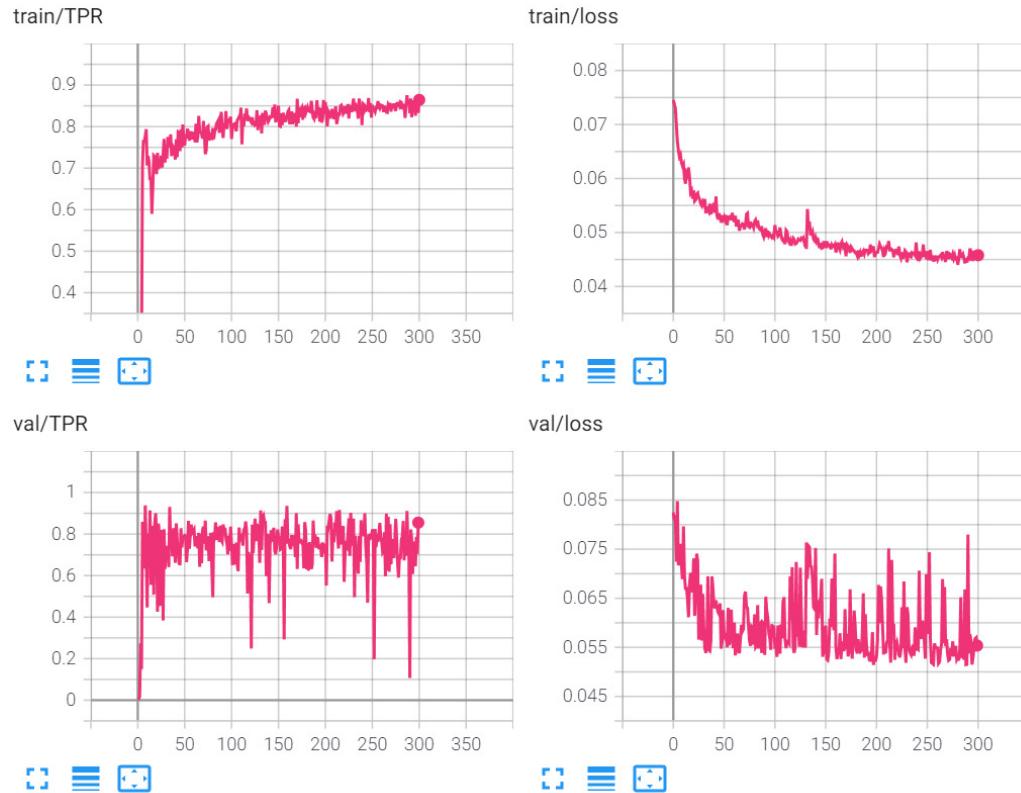


Figura 3.18: **Experimento de aumentado de datos:** Entropía Cruzada y TPR sobre conjuntos de entrenamiento y validación, en función de las épocas. Las métricas sobre el conjunto de entrenamiento (train) muestran un comportamiento normal. El TPR sobre el conjunto de validación (val) muestra gran variabilidad a lo largo de las épocas, y la loss no muestra una disminución respecto a su valor original.

Como se puede ver en la figura 3.18, el comportamiento es similar al observado entrenando con todas las transformaciones de datos, por lo que no se le puede atribuir el mal desempeño del modelo a las mismas.

Cambio en la inicialización de los pesos

Originalmente, para inicializar los parámetros del segmentador, QA entrena una U-Net de forma no supervisada utilizando todas las imágenes de la base de datos de fibrosis. A diferencia del segmentador, este modelo es un *autoencoder*; su objetivo durante el entrenamiento es aprender un espacio de dimensión reducida que permita codificar la entrada, para luego reconstruirla. Ambos modelos difieren en su arquitectura únicamente en la última capa; mientras la salida del *autoencoder* es de 3 canales (RGB), la del segmentador es de 2 canales (clase positiva, clase negativa). Esto permite inicializar todos los pesos del segmentador con los del *autoencoder*, salvo la capa de salida.

3.3. Segmentación de fibrosis

Habiendo experimentado con los parámetros del entrenamiento mencionados anteriormente, y observando los resultados, se optó por aplicar otra técnica de inicialización de los parámetros de la red. Específicamente, se hizo *transfer learning* de un modelo que resuelve una tarea relacionada: segmentación de núcleos en tinciones H&E.

Base de datos de segmentación de núcleos Las bases de datos de segmentación de tinciones H&E más grandes, y disponibles públicamente, son de segmentación de núcleos. Una de ellas fue la desarrollada en el trabajo [51], donde se generaron (de manera semi-automática) segmentaciones de núcleos de cáncer para más de 5 000 WSIs, correspondientes a tinciones H&E de diversos pacientes, laboratorios, y tipos de tejidos. Los datos segmentados pertenecen al repositorio de muestras de The Cancer Genome Atlas Program (TCGA). Cada píxel es clasificado como positivo (canceroso) o negativo (no canceroso).

Esta base de datos presenta la ventaja (respecto a otras como PanNuke, presentada en el trabajo [47]) de que tanto las imágenes como las segmentaciones están en formato *.tiff*. Esto permitió generar los parches y máscaras de entrenamiento con el tamaño y magnificación más adecuado para el caso de estudio.

A modo de probar el método, fueron elegidas 10 WSIs de cáncer urotelial, de las cuales fueron generados 2 161 parches de tamaño 1000×1000 píxeles a magnificación $10\times$. Estos se separaron en conjuntos de entrenamiento y validación, de 1 728 y 433 parches respectivamente. Esta base de datos será referida como BLCA-DB (Bladder Cancer Database, siguiendo la nomenclatura de los autores de la base de datos). El tamaño y magnificación de los parches son iguales que los de la base de datos de fibrosis (F-DB).

Luego fue entrenado el segmentador de QA con el 10% de la BLCA-DB. El objetivo de este primer entrenamiento es sobreajustar el modelo a los datos, para evaluar si la red tiene capacidad suficiente. Los pesos y sesgos fueron inicializados mediante la opción por defecto, que corresponde a una distribución uniforme; las características de la misma pueden verse en la documentación de PyTorch [52]. Los resultados de este entrenamiento no fueron buenos. De manera similar a los entrenamientos anteriores, la función de costo presenta gran varianza a lo largo de las épocas, tanto para el conjunto de entrenamiento como para el de validación. En una siguiente iteración se entrenó inicializando con el método de Xavier (desarrollado en el trabajo [53]), pero los resultados fueron los mismos.

Conclusión

Al no lograr entrenar el segmentador de QA de manera satisfactoria, incluso luego de las modificaciones desarrolladas en esta sección, y ni siquiera al usar una base de datos externa, fue elegida otra implementación de U-Net en PyTorch para ser entrenada; precisamente, un proyecto llamado Pytorch-UNet².

²PyTorch-UNet fue desarrollado para competir en el desafío CARAVANA de Kaggle.

Capítulo 3. Segmentación de estructuras

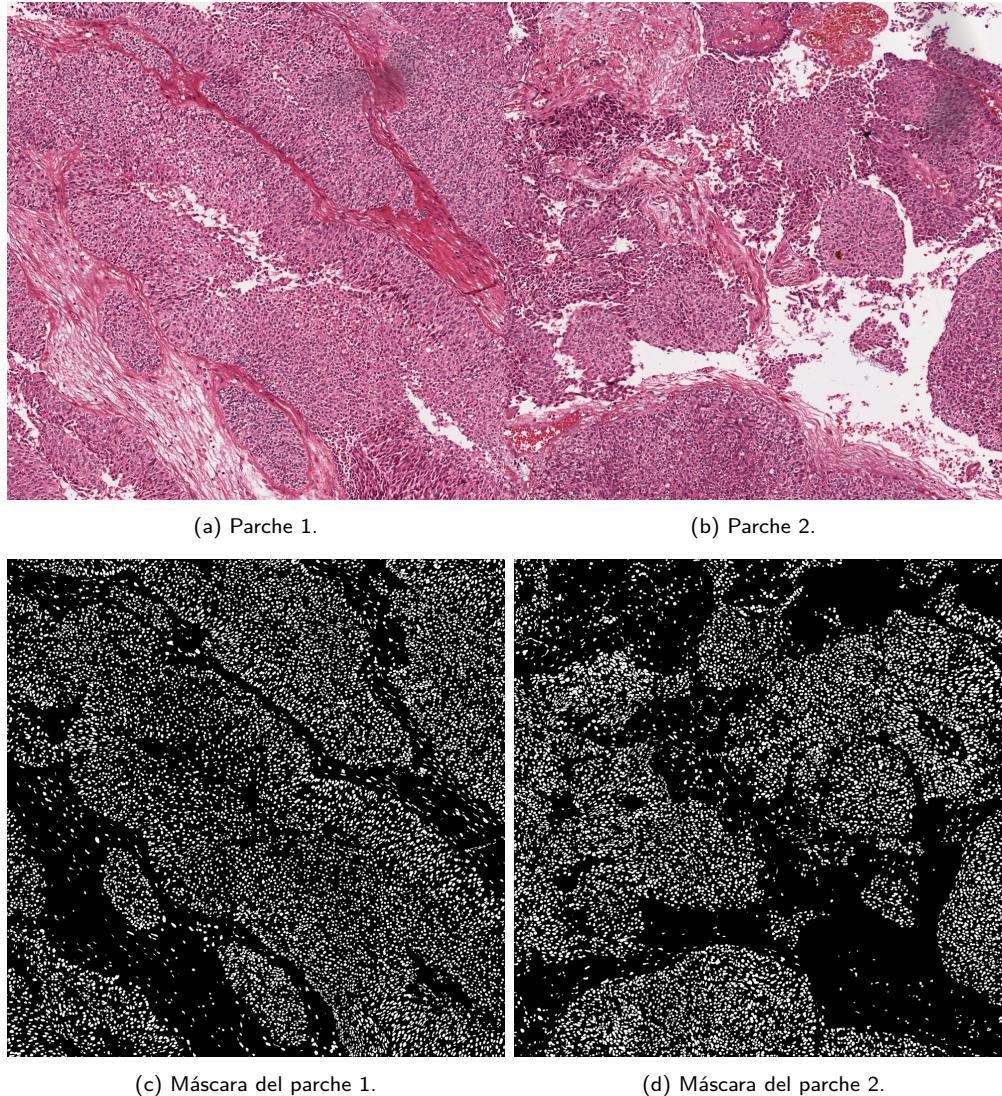


Figura 3.19: Ejemplos de parches de BLCA-DB (figuras (a) y (b)) junto a sus máscaras (figuras (c) y (d)), originarios del trabajo [51]. Cada parche es de tamaño 1000×1000 píxeles. Los píxeles de color blanco de las máscaras pertenecen a clase positiva (núcleo canceroso).

3.3.3. Cambio de implementación: Pytorch-UUnet

Como se menciona en la sección anterior, este proyecto fue elegido como alternativa al flujo de entrenamiento provisto por QA. El repositorio contiene la implementación de una U-Net y su entrenamiento, usando la librería PyTorch. Con el objetivo de segmentar fibrosis, este modelo fue entrenado primero con la base de datos BLCA-DB a modo de inicialización, y luego fue reentrenado con F-DB. A continuación se describe la arquitectura utilizada y las características de los entrenamientos. Luego se presentan resultados de los mismos.

3.3. Segmentación de fibrosis

Descripción del modelo

La arquitectura implementada es exactamente la que se ve en la figura 1.8, a diferencia del tamaño de la entrada; es decir, la cantidad de bloques de *downsampling* y *upsampling* es la misma, así como la cantidad de filtros en cada bloque.

Descripción del entrenamiento

La función de costo optimizada durante el entrenamiento fue

$$L = \left(-\sum_{i=0}^1 \omega_i p_i \log q_i \right) + (1 - DC), \quad (3.5)$$

es decir, la suma de la entropía cruzada (ecuación 3.1) y la función de costo de Dice (ecuación 3.4).

El optimizador utilizado es RMSProp. Se caracteriza por mantener un *learning rate* adaptativo, que se ajusta según una media móvil de la magnitud de los gradientes en un *batch*, para cada peso de la red.

La inicialización de los pesos de la red es la usada por defecto en PyTorch; muestras de una distribución uniforme.

El entrenamiento se hace por épocas, donde en cada época se itera sobre todos los *batches* del conjunto de entrenamiento. El tamaño de un *batch* es de 8 imágenes. Cada 5 *batches* se hace una ronda de evaluación, en la que se reporta el coeficiente de Dice sobre el conjunto de validación.

Resultados

Entrenamiento con BLCA-DB A modo de prueba de concepto se entrenó la U-Net con 10 % de la base de datos BLCA-DB. Para agilizar el entrenamiento, las imágenes son submuestreadas a tamaño 500×500 píxeles antes de entrar a la red.

En la figura 3.20 se pueden ver las métricas de entrenamiento y validación para este experimento, donde se aprecia que el modelo se ajusta correctamente a los datos y logra un buen desempeño en el conjunto de validación.

Entrenamiento con F-DB El modelo mencionado anteriormente se reentrenó usando la base de datos de fibrosis. Un trabajo anterior [54] reportó que los resultados de reentrenar una U-Net fijando los pesos del espacio latente son prácticamente iguales a reentrenar toda la red, por lo que durante este entrenamiento se congelaron dichos pesos. Esta sección concentra aproximadamente la mitad de los parámetros de la red, por lo que es una optimización significativa a la hora de hacer *fine tuning*.

Para este entrenamiento se utilizaron todos los datos de la base de datos de fibrosis, 90 % para el conjunto de entrenamiento y 10 % para validación. A diferencia del entrenamiento sobre BLCA-DB, se mantuvo fijo el valor de *learning rate* en 1×10^{-5} . Los datos son submuestreados a tamaño 500×500 píxeles antes de entrar a la red. Se entrenó el modelo durante 10 épocas.

Capítulo 3. Segmentación de estructuras

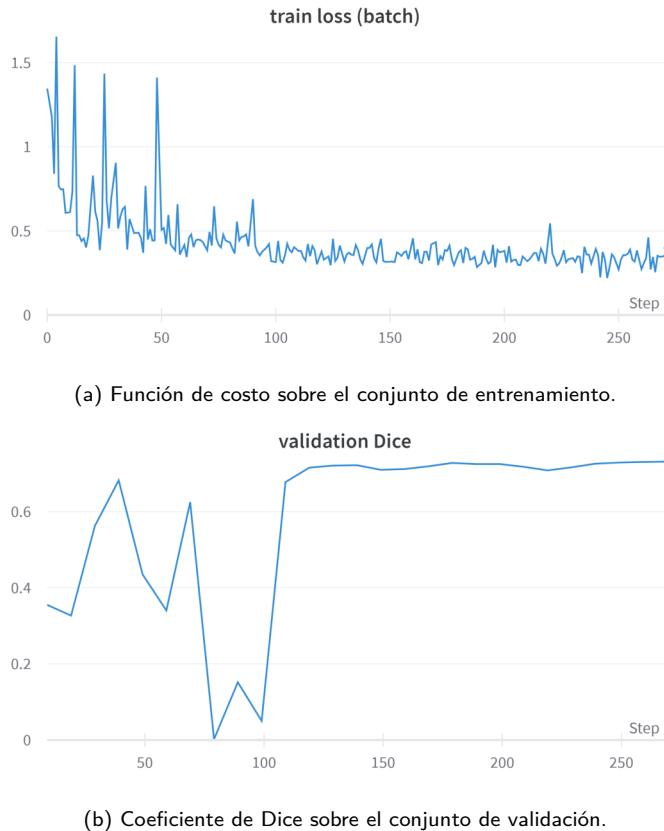


Figura 3.20: Métricas de entrenamiento de PyTorch-UNet sobre BLCA-DB, en función del *batch* (5 épocas en total).

Las métricas del entrenamiento reflejan un comportamiento extraño, como se puede ver en la figura 3.21. Durante los primeros 140 *batches*, la función de costo de entrenamiento tiene una tendencia decreciente, al igual que el coeficiente de Dice en el conjunto de validación. Luego, a partir del *batch* 140, el costo de entrenamiento tiende a estancarse, mientras que el DC en validación incrementa repentinamente y se mantiene en 0.75 para el resto del entrenamiento, lo que supone un buen desempeño sobre este conjunto. Tomando muestras de la inferencia en validación, se ve a partir del *batch* 140 la red predice máscaras completamente negativas, tanto para parches que no tienen fibrosis como para parches que sí. Esta comportamiento llama a hacer un análisis conceptual sobre el cálculo del DC en el contexto de esta base de datos.

Debido a la naturaleza de la fibrosis, la F-DB consta principalmente de imágenes con etiquetas completamente negativas; es decir, sin fibrosis. Cuando se calcula el coeficiente de Dice sobre una máscara completamente negativa, el conjunto *true positives* es vacío. ¿Qué ocurre con el DC cuando $TP = 0$, y la predicción de la red es completamente negativa? La implementación del DC usada en este entre-

3.3. Segmentación de fibrosis

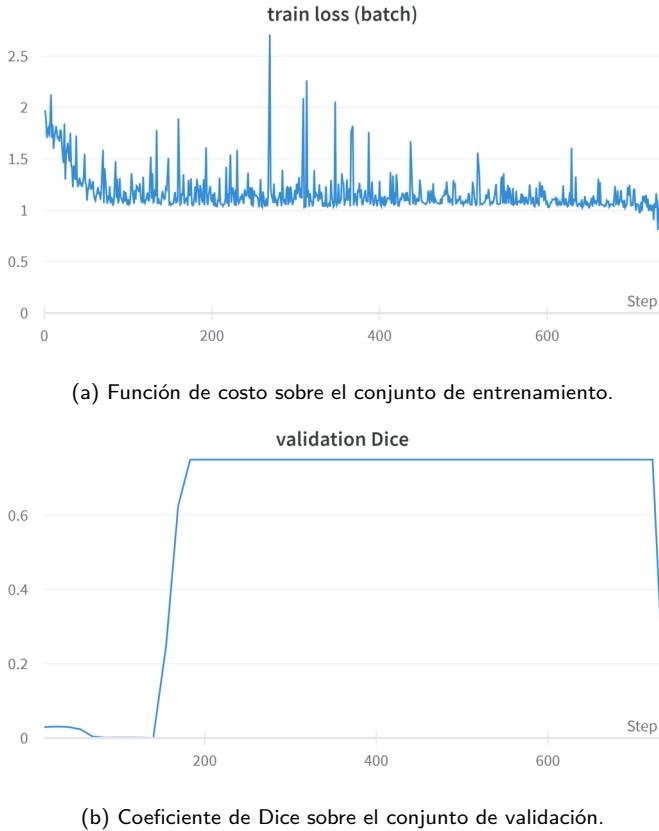


Figura 3.21: Métricas de entrenamiento de PyTorch-UNet sobre base de datos de fibrosis, en función del *batch* (10 épocas en total).

namiento lo calcula de la siguiente manera

$$DC = \frac{2TP + \epsilon}{TP + FP + FN + \epsilon}, \quad (3.6)$$

donde $\epsilon = 1 \times 10^{-6}$ se utiliza para dar estabilidad numérica. En este caso, las cantidades *false positive* (FP) y *false negative* (FN) también son 0, ya que la predicción y el *ground truth* son iguales. Según la definición 3.4 el DC presenta una indeterminación 0/0, y la implementación utilizada (ecuación 3.6) la resuelve definiendo $DC = 1$ arbitrariamente.

La gran cantidad de parches completamente negativos (63 % del total), junto con la participación del DC en la función de costo, incentivan a la red a predecir máscaras completamente negativas ante cualquier entrada. Para resolver esta situación se removieron de F-DB todos los parches con etiqueta completamente negativa.

Entrenamiento con F-DB sin datos completamente negativos Se repitió el entrenamiento con la base de datos de fibrosis, esta vez excluyendo los datos mencionados. Las métricas del entrenamiento, reportadas en la figura 3.22 muestran

Capítulo 3. Segmentación de estructuras

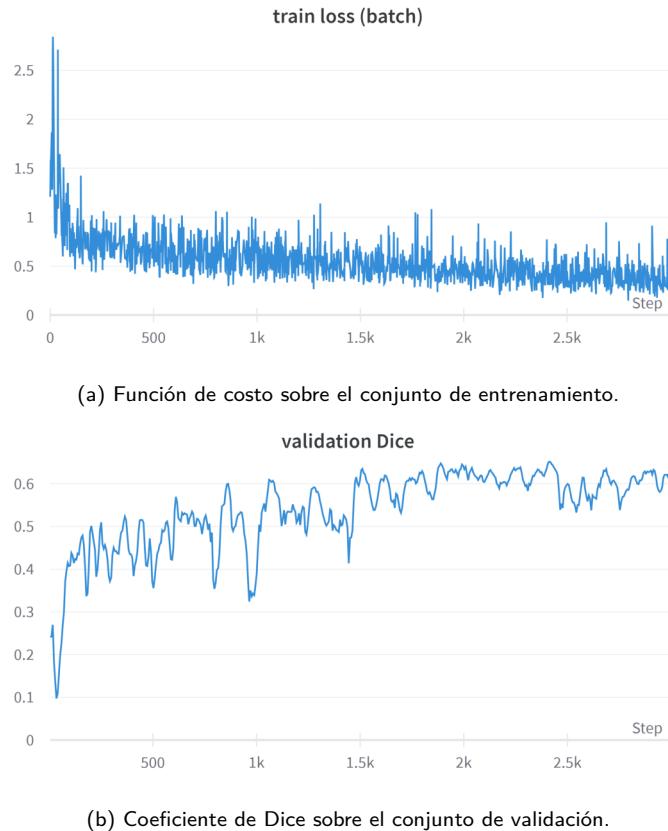


Figura 3.22: Métricas de entrenamiento de PyTorch-UNet sobre base de datos de fibrosis, en función del *batch* (100 épocas en total).

que el modelo logra ajustarse a los datos.

Conclusión

Cuando el DC forma parte de la función de costo, se debe prestar especial atención a la presencia de datos con etiqueta completamente negativa. El estudio realizado no permite concluir nada sobre el *transfer learning* realizado, más que no daña al entrenamiento. En una sección posterior se presentarán resultados de un entrenamiento final y comparaciones con otros entrenamientos.

3.3.4. Modificaciones a F-DB

Dados los resultados anteriores, fueron removidos de F-DB los parches cuyas etiquetas fueran completamente negativas.

Naturalmente, fue entrenado el modelo segmentador de QA con la base de datos modificada. Como muestran los resultados del experimento, en la figura 3.23, el entrenamiento fue exitoso en cuanto a la estabilidad de la función de costo sobre el conjunto de validación. Además, el resto de las métricas (*accuracy*, TPR, *True*

3.3. Segmentación de fibrosis

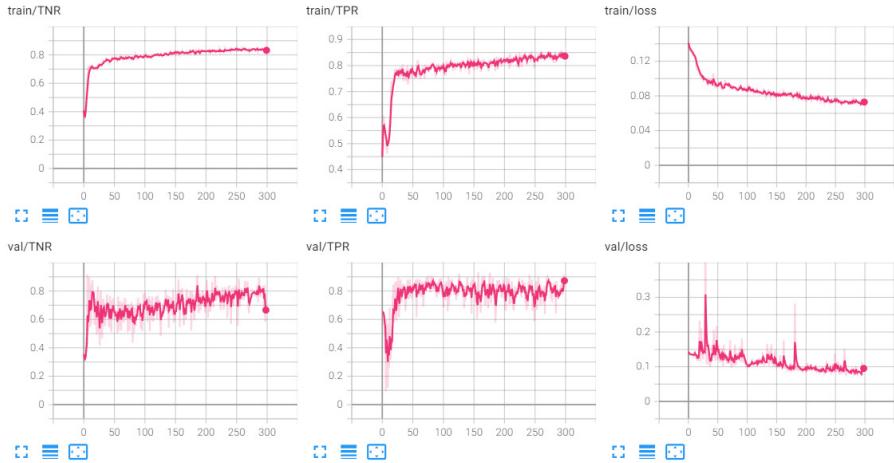


Figura 3.23: Métricas del entrenamiento de QA sobre base de datos de fibrosis en función de las épocas, sin datos completamente negativos.

Negative Rate (TNR)) mejoran a medida que avanza el entrenamiento, tanto sobre el conjunto de entrenamiento como en el de validación.

Entropía cruzada y predicciones negativas

Dado que el entrenamiento con la F-DB original incentiva a la red a predecir segmentaciones negativas ante cualquier entrada, existe la posibilidad de que el valor de la entropía cruzada tome valores muy grandes; por ejemplo, como se ve en la figura 3.16, el promedio en una época se eleva a 5×10^8 . La expresión de la función de costo del entrenamiento es la ecuación 3.1.

En el caso que la función se evalúa para un píxel perteneciente a la clase positiva, se tiene $p_1^{n,m} = 1$ y $p_0^{n,m} = 0$, por lo que los términos de la ecuación 3.1 asociados a la clase negativa valen 0. Dado los valores que alcanza la función de costo, y que la red sobrepredice la clase negativa, el valor $q_1^{n,m}$ tiene que ser muy cercano a cero.

Al tener máscaras completamente negativas, el modelo se ajusta rápidamente a esos datos, ya que al dar una salida completamente negativa el valor de $H(p, q)$ es cero para estas imágenes. Los casos que sí tienen píxeles positivos son tan escasos que, o bien no son suficientes para balancear el valor de $H(p, q)$, o hacen que alcance valores extremadamente grandes, lo que hace imposible el aprendizaje.

Luego de que la red aprende a predecir segmentaciones completamente negativas ante cualquier entrada, el desbalance de clases junto al gran porcentaje de datos completamente negativos, recompensan a la red por este comportamiento, haciendo que llegar a otra solución sea inconveniente. Es importante destacar la diferencia entre tener un desbalance de clases, a tener máscaras completamente negativas. En el primer caso $H(p, q)$ no es cero ya que hay píxeles de clase positiva, y la situación puede balancearse mediante los pesos de las clases.

Capítulo 3. Segmentación de estructuras

3.3.5. Conclusiones

Con el propósito de generar un modelo segmentador de fibrosis fue que se llevaron a cabo los experimentos expuestos en esta sección. A continuación se resumen las principales conclusiones de los mismos.

QuickAnnotator fue útil para la generación y el manejo de la base de datos. Sin embargo, no cumplió con el objetivo de acelerar el etiquetado de datos mediante su modelo segmentador, que no logró desempeñarse bien con los datos generados por la plataforma misma. Su uso en una computadora personal es inviable, ya que cada reentrenamiento requiere de horas de ejecución y recursos computacionales que no están disponibles en este tipo de dispositivos.

Como ya fue mencionado, la fibrosis es un fenómeno que escasea en la mayoría de las WSIs con las que se cuenta. Al usar QuickAnnotator, es importante tener en cuenta la cantidad de parches completamente negativos que conforman la base de datos final.

Por otro lado, el modelo del proyecto PyTorch-UNet muestra buen comportamiento durante el entrenamiento y será utilizado como parte de la solución implementada, que se describe en el siguiente capítulo 5.

Por último, aplicar *transfer learning* entre la segmentación de núcleos y segmentación de fibrosis mostró ser una buena técnica de inicialización de la red para entrenar con F-DB, y ayudó a combatir la escasez de imágenes etiquetadas.

Capítulo 4

Flujo de trabajo

Hasta el momento se describieron los datos de trabajo, la normalización de color como técnica de preprocesamiento, los modelos de segmentación de las estructuras de interés, y el posprocesamiento de las máscaras generadas. En este capítulo se desarrolla la integración de las tareas anteriores en un único flujo de trabajo. El objetivo de esta práctica consiste en contar con un procesamiento completo (punta a punta) de toda una WSI, combinando los *frameworks*, modelos de aprendizaje automático, librerías o todas aquellas técnicas necesarias, en un mismo entorno. Además, combinar la información extraída de las WSIs para realizar un análisis cuantitativo de la muestra.

Los datos cuantitativos son de gran interés para los patólogos; esta información ayudaría determinar el grado de las anomalías presentes en una muestra biológica de forma objetiva. Actualmente la escala de *Kenney-Doig* utiliza criterios abiertos para determinar la categoría en que se encuentra una yegua 1.1, lo que hace que la discusión entre distintas interpretaciones continúe vigente [55].

Más aún, contar con datos cuantitativos habilita la realización de análisis estadísticos, la comparación cuantitativa entre WSIs, o de regiones distintas de una misma WSI. En particular, para este último caso, a partir del flujo de trabajo completo es posible volver a la WSI para señalar la o las regiones que presentan valores por fuera de los rangos medios. Es decir, regiones donde sería aconsejable prestar mayor atención a la hora del análisis clínico.

A pesar de las bondades de la extracción de datos cuantitativos, es importante señalar que el foco del flujo de trabajo propuesto no se centra en esta finalidad. El énfasis de la investigación se encuentra en la unificación de las herramientas de trabajo en un mismo entorno, y en el análisis punta a punta de una WSI completa.

La implementación fue desarrollada en Python y hace uso de la librería FAST [56, 57]. Esta herramienta es de código abierto, desarrollada en *C++*, y fue diseñada para realizar procesamiento de alto rendimiento, inferencia de modelos de aprendizaje automático y visualización de imágenes médicas. Ofrece la posibilidad de trabajar con varios tipos de procesadores (*multi-core CPUs*, GPU) y con diversos motores de inferencia (OpenVINO, TensorRT, Tensorflow), evitando que el usuario tenga que interactuar con las *APIs* de cada una de estas librerías directamente. Además, optimiza el manejo de memoria y acepta una amplia variedad

Capítulo 4. Flujo de trabajo

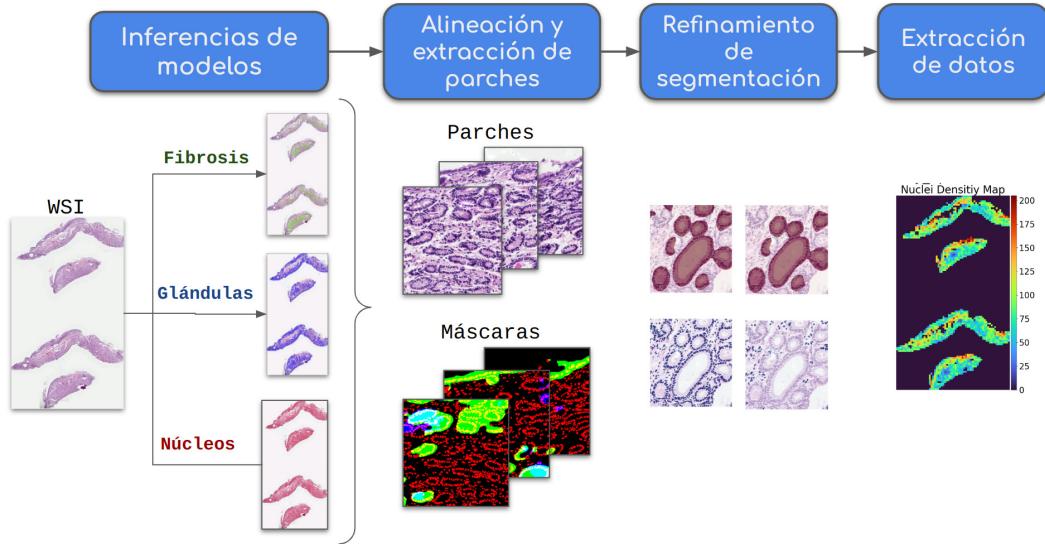


Figura 4.1: Flujo de trabajo propuesto, dividido en cuatro módulos principales: inferencia de modelos, alineación y extracción de parches, refinamiento de segmentación y extracción de datos.

de formatos para los datos de entrada; dos aspectos importantes en el área del procesamiento de imágenes médicas. Puntualmente para el caso de estudio, esto es fundamental para trabajar con WSIs. Resulta oportuno mencionar que FAST fue desarrollado por el mismo equipo de trabajo que FastPathology. En definitiva, FastPathology se trata de una interfaz gráfica que utiliza FAST de base.

El flujo de trabajo consiste en la ejecución secuencial de una serie de etapas, tal como se muestra en el diagrama de la figura 4.1. El primer paso consiste en realizar la inferencia sobre toda la WSI con cada modelo de segmentación, y en generar las máscaras correspondientes. A continuación se recorta cada máscara en parches de igual tamaño y magnificación. Se posprocesan las máscaras a nivel de parche, combinando la información de cada estructura, para mejorar los resultados. Por último, se calculan algunas características sobre toda la WSI y se señalan sobre ella regiones que requieren de mayor atención. En las siguientes secciones se describe en profundidad cada etapa del flujo de trabajo.

La implementación desarrollada se encuentra disponible en Github bajo el nombre del proyecto Mare FEst

4.1. Inferencia de los modelos de segmentación

Inicialmente se realiza inferencia con los modelos de segmentación. Para ello es necesario partir la WSI en parches y correr el modelo sobre cada parche. El tamaño de imagen y la magnificación dependen de los datos con los que haya sido entrenado cada modelo. Una vez realizada la inferencia sobre todos los parches, estos se vuelven a unir y se genera una máscara de toda la WSI en formato *tiff*. FAST proporciona funcionalidades para realizar estas tareas.

4.1. Inferencia de los modelos de segmentación

El procedimiento completo requiere de la configuración de varios parámetros, como el tamaño de las imágenes que entran a la red, el grado de solapamiento entre las mismas, y su magnificación (que determina las dimensiones de las estructuras biológicas que ve la red). Estos tres parámetros determinan la cantidad de parches resultantes de la WSI. A mayor solapamiento y magnificación, y menor tamaño de imagen, aumenta la cantidad de imágenes necesarias para cubrir la WSI. Como el modelo de núcleos requiere de mayor magnificación y menor tamaño de entrada que los modelos de glándulas y fibrosis, se le configuró un menor solapamiento, de modo de disminuir la cantidad de parches y así acelerar la inferencia sobre toda la WSI.

Otro parámetro importante en modelos de segmentación es el umbral de salida de la red. Como fue descrito en capítulos anteriores, varios modelos de segmentación tienen dos canales de salida: uno representa la probabilidad de cada píxel de pertenecer a la clase positiva, y el otro representa la probabilidad complementaria. Una práctica habitual es asignarle al píxel la clase de mayor probabilidad, para condensar la información en forma binaria en un solo canal. Sin embargo también es posible determinar un umbral e imponerlo en el canal de interés. Este fue el caso para el modelo de glándulas o el de epitelio, donde se toma un umbral de 0.2 para captar mayor cantidad de píxeles.

FAST cuenta con la posibilidad de utilizar un archivo escrito en pseudo-código para especificar estos atributos, junto con las tareas a ejecutar. En muchos casos este método de implementación puede ser de ayuda, ya que no es necesario contar con conocimientos de programación para ejecutarlo. En FastPathology esta es la única opción de configuración posible. En el anexo B.1.1 se incluye el archivo utilizado para generar inferencias con el modelo de núcleos, a modo de ejemplo.

Los algoritmos de segmentación seleccionados para llevar a cabo el análisis completo son: el modelo de glándulas (el usuario puede optar si utilizar el modelo de epitelio [40] o el de glándulas de GlaS [42]), el modelo de núcleos [20], y un modelo de segmentación de fibrosis entrenado con datos propios. Los dos primeros fueron introducidos en el capítulo 3, mientras que el modelo final de fibrosis es presentado en el capítulo 5.

Inicialmente se buscó unificar el módulo de inferencia de máscaras junto con el refinamiento de las mismas. Esto conllevaría una significativa mejora en el rendimiento, ya que, en inferencia se refinarián todas las imágenes de la WSI, eliminando la necesidad de volver a procesar cada parche individualmente. Para realizar eso, sería necesario una alineación de todas las máscaras de forma simultánea, utilizando un método de FAST llamado *Python Process Object*. Esto fue posible para el filtro de relleno de huecos (desarrollado en la sección 3.1), ya que este filtro es agnóstico a otras máscaras. Sin embargo, no es el caso del filtro de núcleos (el cual necesita las máscaras de glándulas) o el filtro de eliminación de epitelio (que necesita máscaras de tejido). Por lo tanto no fue posible aplicar estos filtros en esta etapa.

Una práctica habitual para optimizar el proceso de inferencia es umbralizar la WSI completa de forma de obtener una máscara de segmentación de tejido. FAST también proporciona una funcionalidad para esto, utilizando procesamiento

Capítulo 4. Flujo de trabajo

clásico de imágenes. Al fraccionar la WSI en parches también se divide la máscara de segmentación de tejido; luego, al momento de trabajar sobre cada parche se determina si el porcentaje de fondo (donde no hay tejido) es mayor a un umbral. En caso de que lo sea, se descarta el parche.

En cuanto al formato de almacenamiento de los pesos y arquitectura del modelo se trabajó siempre con la extensión *onnx*. Se trata de un formato abierto para modelos de aprendizaje automático que permite intercambiar desarrollos provenientes de distintos *frameworks*. En algunos casos, como los modelos de segmentación de glándulas o fibrosis, fue necesario convertirlos de *pth* (formato de PyTorch) a *onnx* para hacerlos compatibles con FAST.

4.2. Alineación y extracción de parches

Una vez que se cuenta con los archivos *tiff* que contienen las máscaras de segmentación de toda la WSI, una para cada modelo, se procede a alinear y recortar en parches. Para realizar esto se utiliza un archivo de configuración, donde se establecen todas las variables para determinar este recorte. Este archivo se guarda junto con los parches para mantener un registro de las mismas. Estas variables corresponden al tamaño de recorte de la imagen, nivel de magnificación, orden de los canales donde se guardan las máscaras, y opcionalmente el tamaño de escalado de la imagen al momento de realizar el refinado de las máscaras (para acelerar el procesamiento).

Como las inferencias de las segmentaciones son realizadas con niveles de magnificación y tamaños de imagen diferentes (adecuados a cada modelo), es necesario alinearlas para que coincidan entre sí. Para esto se toma la información del alto y largo de toda la WSI y también de los archivos *tiff*, y se escalan para que todas coincidan en un mismo punto. Luego, una funcionalidad de FAST permite extraer cada imagen y convertirlo a un tipo de dato adecuado al posprocesamiento. De modo de no perder la información de espacialidad, las coordenadas dentro de la WSI de donde es extraída la imagen son guardadas en el nombre del archivo (tanto en el parche como en la máscara). Cada máscara es guardada como una imagen RGB, donde cada canal representa una segmentación.

En esta oportunidad, las máscaras y la WSI son recortadas del mismo tamaño (1024×1024), a nivel 0 que corresponde a una magnificación de $40\times$. La elección del tamaño de parche y magnificación tenía que adecuarse simultáneamente a todos los modelos utilizados. De esta forma, la magnificación de $40\times$ permite apreciar los núcleos con suficiente detalle, mientras que el tamaño de parche de 1024×1024 capta una región suficientemente grande como para apreciar glándulas completas. En la figura 4.2 se ilustran estas características.

En esta etapa se genera un archivo de reporte de tipo *json* que contiene no solo información de la configuración sino que también guarda datos de la misma WSI, como sus dimensiones, nombre, etc. Posteriormente otros datos de relevancia son guardados en este archivo.

4.3. Refinamiento de las máscaras de segmentación

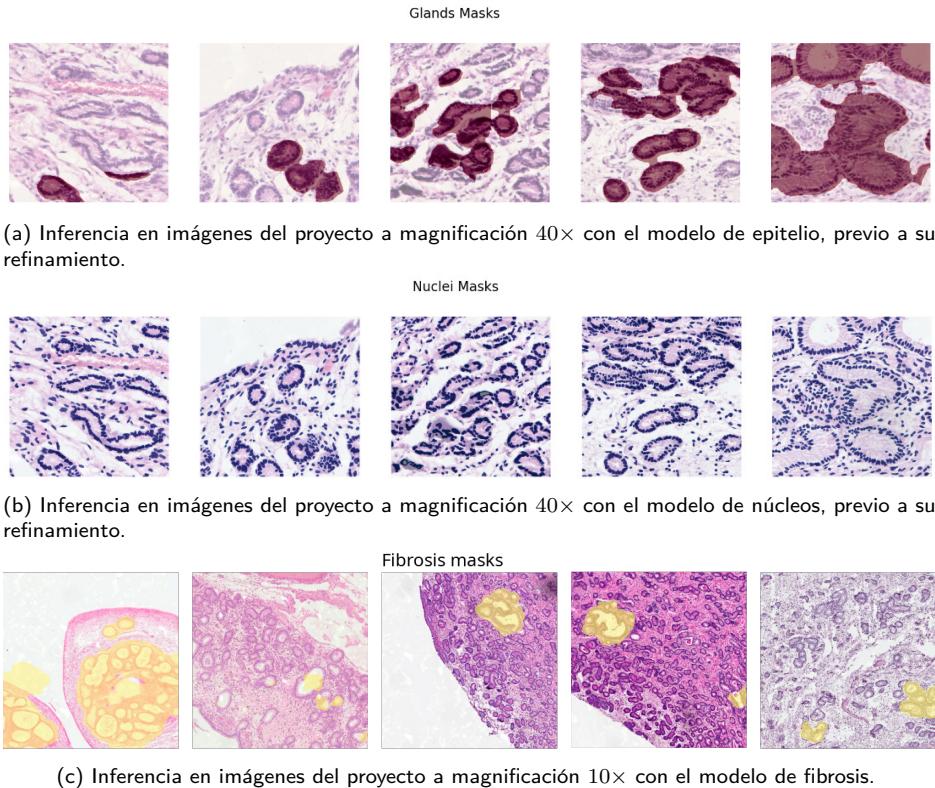


Figura 4.2: Etapa de extracción y alineación de máscaras. Inferencias de glándulas y núcleos previos al refinamiento. Inferencias del modelo de fibrosis. Se observa que la magnificación de 40× es adecuada para observar las primeras dos clases, no así la tercera.

4.3. Refinamiento de las máscaras de segmentación

El tratamiento posterior sobre las máscaras de glándulas y núcleos fue descrito en las secciones 3.1 y 3.2 respectivamente. En el primer caso el objetivo es mejorar la segmentación obtenida, rellenando la luz de las glándulas, y filtrar segmentaciones no deseadas, como el tejido luminal. Mientras que en el segundo, el propósito consiste en filtrar instancias que no forman parte del objeto de interés, es decir, descartar núcleos que se sabe de antemano que no son linfocitos.

De acuerdo con lo que se describió en la sección 3.2, para llevar adelante la segmentación de núcleos se utiliza la segmentación de glándulas. El argumento que avala esta práctica es que un núcleo que se encuentra dentro de una glándula no se trata de un linfocito. De este modo, remover estos núcleos de la máscara simplifica la tarea posterior de clasificación, donde las clases son linfocito / no linfocito.

Dentro de estos procesamientos se realizan operaciones morfológicas que tienden a ser computacionalmente muy costosas. Con el fin de acelerar este procedimiento, las máscaras son transformadas a tamaño 512×512 (o cualquier dimensión determinada en el archivo de configuración) durante el refinamiento y luego son devueltas a su tamaño original.

La ventaja de tener todas las máscaras en las mismas coordenadas es que per-

Capítulo 4. Flujo de trabajo

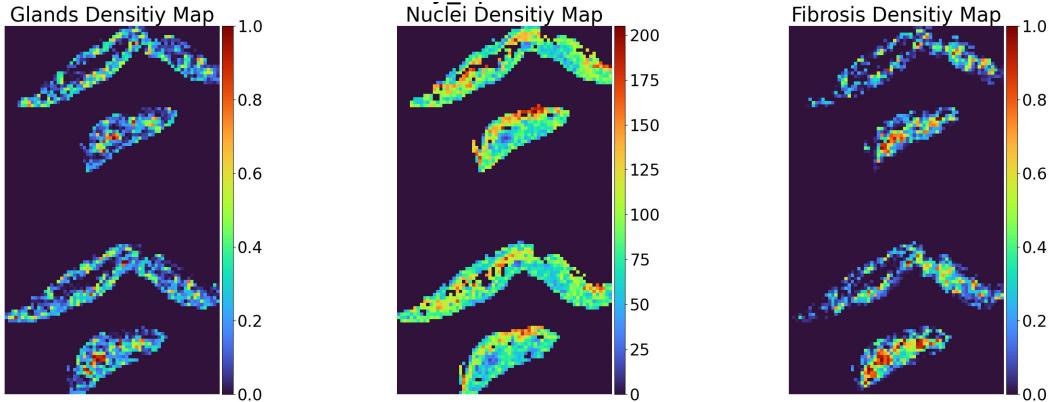


Figura 4.3: Mapas de calor de la densidad de glándulas, núcleos y fibrosis, de la WSI *Lady.svs*, generados utilizando los modelos de segmentación. La escala de los mapas de glándulas y fibrosis está normalizada, y la densidad en esos casos se mide como la cantidad de píxeles segmentados sobre el total de píxeles, por parche. En el caso del mapa de densidad de núcleos, la densidad se calcula como la cantidad de núcleos por parche.

mite cruzar información de las mismas. Como se mencionó en capítulos anteriores, se utiliza la máscara de glándulas para refinar núcleos, y los parches de la imagen para refinar las glándulas. También se podría haber cruzado información entre otras segmentaciones, como por ejemplo entre glándulas y fibrosis, dado que estas presentan varios puntos en común. Sin embargo durante el desarrollo del proyecto no se exploraron estas alternativas. En la sección *Trabajo a futuro* 6.2 se comentan algunas opciones que podrían ser interesantes para mejorar las segmentaciones de fibrosis.

En este módulo también se calculan los mapas de densidad para cada canal. Con la información de la ubicación de cada imagen dentro de la WSI se genera un mapa de calor donde se pueden observar regiones con acumulación de núcleos o alta densidad de glándulas. En el caso del mapa de fibrosis y de glándulas, se reporta para cada imagen la cantidad de píxeles con detección positiva sobre área total de la imagen. Mientras que para núcleos se reporta cantidad de instancias de núcleos por imagen. En la figura 4.3 se observan estos mapas. Esta información se almacena en un archivo de la librería Numpy `density_map.npy`, donde se tiene en cada canal las densidades de cada segmentación y un cuarto canal con información binaria de existencia o ausencia de tejido.

4.4. Extracción de datos cuantitativos

Finalmente, las máscaras de segmentación recortadas y procesadas son utilizadas para calcular características de las instancias detectadas. A partir de las máscaras de glándulas se estudia la distribución del área, y a partir de las máscaras de núcleos se busca clasificar los mismos.

4.4. Extracción de datos cuantitativos

4.4.1. Extracción de información de glándulas

En lo que refiere a las máscaras de glándulas se decidió mantener ambos modelos para que el usuario decida cuál utilizar según su propósito. En el caso del modelo entrenado con GlaS muchas glándulas no son detectadas; aquellas que sí lo son presentan una definición precisa de sus límites. Este tipo de inferencia es buena si se quiere realizar un análisis de la forma de las glándulas, sin embargo al no detectar gran cantidad de glándulas, no es bueno para un análisis de densidad. Por otra parte el modelo de segmentación de epitelio logra detectar conjuntos de glándulas, mientras que la forma no se caracteriza correctamente; es un mejor modelo si se quiere caracterizar densidad. Es por esto que el mapa de densidad se reporta en cantidad de píxeles sobre área total, en vez de reportar cantidad de glándulas por imagen.

En esta sección se busca caracterizar la densidad espacial de glándulas. Para ello, se identifica cada instancia de glándula en las máscaras y se calcula su área relativa a la imagen. Visualizando algunas inferencias se aplicó un filtrado para eliminar aquellas instancias con áreas menores a cierto umbral, considerándolas espurias. En la figura 4.4 se observan las inferencias para cada modelo y la comparación de ambos histogramas. Es interesante observar como el histograma para el modelo de epitelio posee más datos en áreas mayores a 0.8, señalando posibles nidos glandulares, mientras que el modelo de GlaS infiere áreas mucho menores.

Se genera un archivo *csv* que contiene la información de todas las glándulas de la WSI. Para cada una se guarda un identificador, su ubicación dentro de la WSI (coordenadas de la imagen donde fue extraída), un identificador de la glándula dentro de la imagen, y su área.

4.4.2. Extracción de información de núcleos

En cuanto a las máscaras de núcleos, como no se contaba con datos etiquetados para entrenar un modelo de clasificación, se optó por calcular características que permitan identificar los linfocitos de forma no supervisada. Debido a esto, el enfoque de esta sección fue exploratorio y deja abierta la puerta a varias líneas de investigaciones futuras.

Las características seleccionadas son: área, circularidad, nivel de intensidad, y textura. A continuación se detalla la razón por la que se seleccionaron estos atributos y se incluye la figura 4.5 como referencia para observar las características susodichas.

- Área: los linfocitos presentan núcleos con área ligeramente menor que otras células. El área se obtiene aplicando una función obtenida de la librería OpenCV. Mas detalles acerca de que funciones se utilizaron y con que parámetros se pueden encontrar en el anexo ??.
- Nivel de intensidad: el color del núcleo es de intensidad baja. Para obtener el nivel de intensidad se toma la imagen en escala de grises y se calcula la intensidad media de cada núcleo.

Capítulo 4. Flujo de trabajo

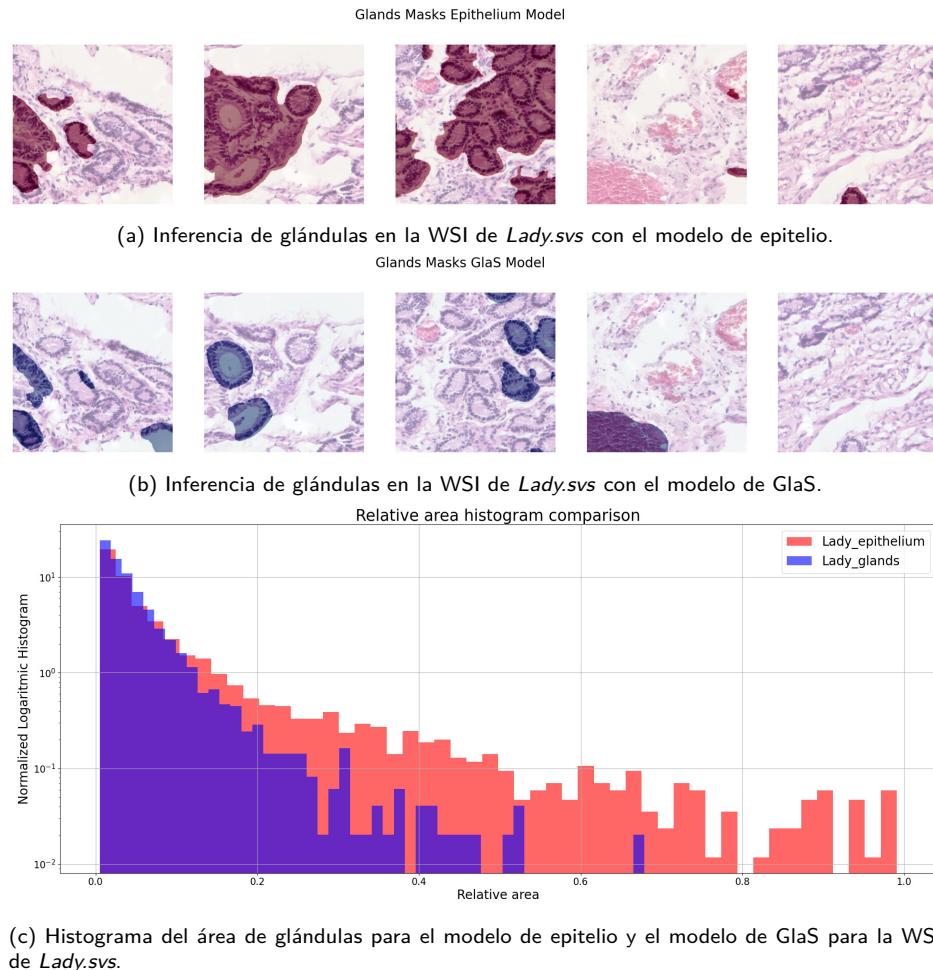


Figura 4.4: Comparación de inferencias para ambos modelos de detección de glándulas teniendo en cuenta el área relativa de las glándulas para la WSI de *Lady.svs*.

- Circularidad: se define mediante la siguiente fórmula:

$$\text{circularidad} = \frac{4\pi A}{P^2},$$

donde A representa el área y P el perímetro. Puede tomar un valor de 1 (máximo valor posible) cuando la segmentación es un círculo, y $\pi/4$ cuando es un cuadrado. Los linfocitos suelen ser bien redondos, por lo que se espera que tomen un valor cercano a 1 para este descriptor. Al igual que para obtener el área, para hallar el perímetro se utilizó una función de OpenCV destinada para ello. Mas información acerca de su aplicación se puede encontrar en el anexo ??.

- Textura: de acuerdo a lo mencionado con anterioridad 3.2, la textura de los linfocitos es uniforme. Se utilizaron tres medidas distintas para evaluar esta magnitud. Para todos los casos fue necesario hallar previamente el histograma normalizado de los niveles de intensidad en escala de grises, que

4.4. Extracción de datos cuantitativos

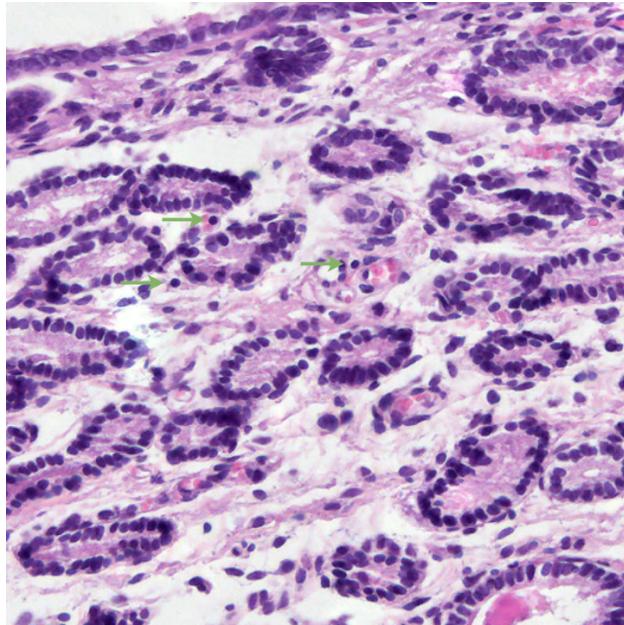


Figura 4.5: Parche recortado de la WSI *Lady.svs*. Se señalan con flechas verdes los linfocitos presentes en la imagen. Pude observarse que estos núcleos tienden a ser más pequeños, circulares y oscuros que el resto.

quedan comprendidos en la segmentación. A este histograma se lo denota como $p_z(i)$, donde z es una variable aleatoria que toma niveles de gris, esto es $z = i$ con i en el rango $[0, 255]$.

La primera medida utilizada se calcula de la siguiente forma:

$$R(z) = 1 - \frac{1}{1 + \sigma^2(z)},$$

donde $\sigma^2(z)$ representa el desvío estándar de la variable z en la imagen. Esta medida toma valores cercanos a 0 cuando la zona es poco texturada, y valores próximos a 1 cuando la textura es fuerte.

Otra medida utilizada es:

$$U(z) = \sum_{i=0}^{255} p_z^2(i),$$

que también refleja el grado de variabilidad de la textura. Como p_z toma valores en el rango $[0, 1]$ y su suma es igual a 1, el valor de U es máximo cuando todos los niveles de intensidad son igualmente probables (uniformidad máxima) y decrece en otros casos.

Por último, la tercer medida de textura implementada es la entropía, definida como:

Capítulo 4. Flujo de trabajo

$$e(z) = - \sum_{i=0}^{255} p_z(i) \log_2 p_z(i).$$

La entropía es una medida de variabilidad y vale 0 para regiones constantes.

Se comienza el análisis estudiando si fijar un umbral para una de las características permite identificar posibles candidatos. Luego se continúa analizando la distribución de las características dos a dos, para ver si fijar umbrales en dos características permite obtener la clasificación deseada. Y por último se aplica un algoritmo para agrupar las distintas instancias de acuerdo al grado de similitud que presentan entre sí, en base a todas las características halladas.

Capítulo 5

Resultados

El objetivo principal del trabajo consiste en explorar la incorporación de nuevas técnicas para el análisis de biopsias endometriales de yeguas. Las técnicas a analizar debían provenir principalmente de las áreas de procesamiento de imágenes y aprendizaje automático. También se propuso estudiar herramientas de trabajo y *frameworks* para el análisis de las muestras biológicas, así como librerías para el procesamiento digital de las mismas.

Con respecto al procesamiento de imágenes, se propuso identificar estructuras biológicas de interés en las muestras, por medio de la aplicación de modelos de aprendizaje automático de segmentación. Para el caso de las glándulas y núcleos se utilizaron modelos preentrenados que se encontraban disponibles y tenían un desempeño aceptable en las imágenes de trabajo; pero para la segmentación de fibrosis se decidió entrenar un modelo con datos propios.

En la sección 3.3 se describió el primer acercamiento al *framework* de trabajo, utilizado para etiquetar datos y entrenar un modelo de segmentación de fibrosis (QA). También se expusieron las dificultades que se encontraron al utilizar la herramienta, y un nuevo modelo al que se recurrió ante las dificultades atravesadas. El desenlace llevó a tener que modificar la base de datos de fibrosis y remover las imágenes sin etiquetas positivas.

Todos los cambios implementados aportaron para que el modelo comenzara a aprender. En este capítulo se describen los entrenamientos posteriores, tanto con el modelo de QA como con el de Pytorch-UNet. Se detallan las particularidades de cada entrenamiento; como por ejemplo, cantidad de épocas, métodos de aumento de datos, inicialización de los pesos de la red, y técnicas de normalización de color. Sobre este último punto, también se presentan los resultados obtenidos al entrenar el modelo de HistAuGAN, junto con un análisis de sus limitaciones.

En cuanto a la arquitectura de los modelos, estas fueron introducidas previamente y no se aplicaron cambios adicionales en esta sección. Se presenta la metodología de evaluación implementada y los resultados obtenidos a partir de estos entrenamientos.

En lo que refiere al estudio de los *frameworks* y herramientas, luego de analizar varios recursos disponibles y probar su desempeño sobre los datos de trabajo, se optó por utilizar la librería FAST. Además de las facilidades que brinda esta

Capítulo 5. Resultados

librería para trabajar con WSIs, poder implementar la propuesta final en Python resulta conveniente porque permite aplicar técnicas de posprocesamiento y cálculos de datos cuantitativos, con herramientas comúnmente utilizadas en el área de procesamiento de imágenes.

El flujo de trabajo punta a punta, que es donde se aplican las técnicas seleccionadas para analizar una WSI de forma completa, permite combinar la información de las diferentes estructuras biológicas estudiadas a partir de una única ejecución. En este capítulo se presenta el análisis de los resultados obtenidos al aplicar el flujo de trabajo descrito en la sección 4.

Resulta oportuno aclarar que el módulo donde se realiza el cálculo de resultados cuantitativos fue desarrollado con fines exploratorios. Se estudia la posibilidad de analizar la distribución del área de las glándulas, con los dos modelos disponibles para ello; y se analiza la viabilidad de clasificar linfocitos, sin contar con datos etiquetados para entrenar un algoritmo supervisado.

5.1. Entrenamiento de HistAuGAN

Se entrenó el modelo de HistAuGAN con los tres dominios de trabajo para aplicar una técnica de normalización de color sobre las imágenes de entrenamiento. Se utilizaron imágenes de tamaño 1000×1000 con nivel de magnificación de $10\times$. La base de datos de entrenamiento contenía 241 imágenes del dominio violeta, 213 del dominio rosado y 226 del dominio rojo. Los parámetros de entrenamiento fueron un redimensionamiento de las imágenes a 256×256 y un recorte aleatorio de 216×216 .

Se tomaron cuatro imágenes de cada dominio de tamaño 1000×1000 para conformar una base de datos de prueba. Vale aclarar que estas imágenes no integraban el conjunto de entrenamiento. Se pueden ver los resultados de inferencia del modelo entrenado sobre algunas de las imágenes de prueba en la figura 5.1

Los resultados de la transferencia de dominio con el modelo de HistAuGAN son muy buenos. Las imágenes generadas son realistas y logran captar la distribución del espacio de color de cada dominio. Sin embargo, se puede ver en varias de las imágenes generadas un patrón cuadriculado de 4×4 . Este patrón se observa principalmente en el fondo de la muestra, donde los tonos son más claros y uniformes. A partir de las pruebas realizadas se observó que el patrón no variaba de tamaño al modificar las dimensiones de las imágenes de entrada a la red, ni el valor de redimensionamiento. En la figura 5.2 se muestran algunos ejemplos del patrón cuadriculado observado.

Al investigar el origen de este defecto, se encontró que los patrones cuadriculados son un artefacto común de los modelos generativos. Estos modelos cuentan con capas de deconvolución, también conocidas como convolución traspuesta, que realizan un proceso de interpolación para pasar de un espacio de baja dimensión a otro de mayor. Cuando el tamaño del núcleo de convolución no es divisible entre el tamaño del paso, se produce solapamiento sobre algunos píxeles, por lo que el modelo pasa dos veces por el mismo lugar y se suma la información generada. Considerando que los modelos cuentan con varias capas de deconvolución, observamos

5.1. Entrenamiento de HistAuGAN

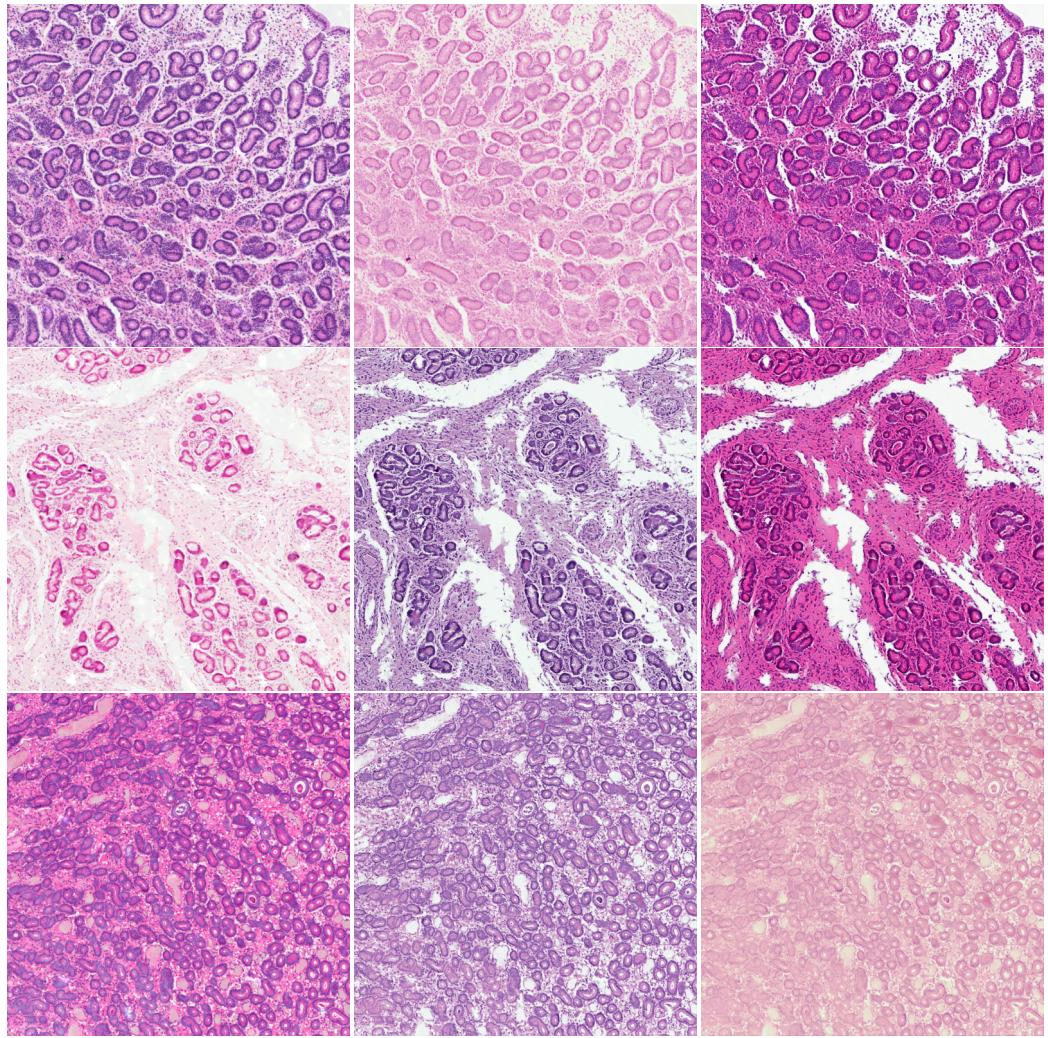


Figura 5.1: Resultados con imágenes de prueba del modelo entrenado de HistAuGAN. La columna de la izquierda contiene las imágenes originales, a la derecha de cada imagen se encuentran las imágenes generadas por el modelo en los otros dominios.

que este problema afecta a varios niveles de escala de la red.

El artículo [58] propone una alternativa sencilla a las capas de deconvolución que es realizar una interpolación del tipo vecino más cercano o interpolación bilineal, seguido de una capa convolucional. Esta modificación no fue puesta en práctica debido a que excedía el alcance del proyecto, sin embargo se deja expuesta esta línea de trabajo para futuros acercamientos.

En cuanto a cómo integrar el algoritmo seleccionado con el proyecto de estudio, se consideran dos posibilidades que podrían tener efectos positivos en los entrenamientos de los modelos de segmentación. El primero podría ser transformar todas las imágenes que entran al modelo a un mismo dominio, tanto las de entrenamiento como las de prueba, de forma de reducir la variabilidad de color que pueden tener las estructuras a detectar. La segunda opción evaluada consiste en pasar todas las

Capítulo 5. Resultados

imágenes de entrenamiento a todos los dominios existentes en la base de datos. Esto constituye una técnica de aumentación de datos, y por lo tanto introduce mayor cantidad de imágenes para entrenar.

Ambas opciones son válidas y presentan sus ventajas. La primera opción implica que el modelo de segmentación no deba ser reentrenado cada vez que observa un nuevo dominio. La desventaja es que requiere de un preprocesamiento al hacer inferencia que conlleva mayor tiempo. Por otro lado, la segunda opción tiene como ventaja que independiza la detección de las estructuras de los colores, por lo que el modelo ya entrenado podría servir para imágenes de dominios nunca vistos por éste. Sin embargo, la desventaja es que entrenar un modelo de esta forma es más costoso computacionalmente, ya que las imágenes de entrenamiento se multiplican por la cantidad de dominios existentes en el problema.

5.2. Entrenamientos del modelo de fibrosis

Una vez que se encontró la causa del mal desempeño de los modelos entrenados con F-DB, mencionada en la sección 3.3.4, se experimentó cambiando algunos hiperparámetros de los modelos segmentadores de fibrosis para observar su impacto en el desempeño sobre el conjunto de prueba.

Para el modelo de QA se hicieron tres entrenamientos: sin aumentación de datos, con aumentación de datos, y con la normalización de color de HistAuGAN. Con el modelo de Pytorch-UNet se experimentó haciendo *transfer learning* entre BLCA-DB y F-DB, y también con la normalización de color de HistAuGAN.

En esta sección se describen las características de los entrenamientos, el conjunto de pruebas, y finalmente se presentan los resultados obtenidos.

5.2.1. Aumentaciones de datos conservadas de QA

Como fue mencionado en la sección 3.3.2, las aumentaciones de datos originales de QA introducen algunos artefactos en las imágenes que no son naturales al problema en cuestión. A la hora de entrenar el modelo final, se aplicó el siguiente subconjunto de las transformaciones originales de modo de no introducir dichos artefactos:

- Normalización: todas las imágenes son normalizadas en el rango $[0, 1]$ previo a entrar a la red.
- Rotación: cada imagen tiene probabilidad 0.5 de ser rotada 180° respecto a los ejes vertical y horizontal.
- *Blur*: se aplica un filtro pasabajos con kernel de tamaño aleatorio (entre 3 y 7), con probabilidad 0.25.
- Brillo: se modifica el brillo de cada imagen, con probabilidad 0.25. Este se aumenta o disminuye una cantidad aleatoria entre 0 y 20%.

5.2. Entrenamientos del modelo de fibrosis

- Contraste: se aumenta/disminuye el contraste, con probabilidad 0.25, una cantidad aleatoria entre 0 y 20 %.
- Gamma: se transforma, con probabilidad 0.25, la intensidad (I) de la imagen de acuerdo a la siguiente expresión

$$I_{transformada} = I^\gamma, \quad (5.1)$$

donde γ toma un valor aleatorio entre 0.8 y 1.2, y la potencia se aplica píxel a píxel.

- HSV: con probabilidad 0.5, la imagen se convierte desde el espacio de color RGB al espacio HSV (*Hue-Saturation-Value*). A sus valores en cada canal se le suma/resta un valor aleatorio en los rangos (0; 20), (0; 10), y (0; 10) respectivamente. Luego se devuelve la imagen al espacio RGB.

En la figura 5.3 se puede ver el efecto de cada una de estas transformaciones sobre una imagen de prueba (recordar que los parámetros de la transformación son aleatorios y varían en cada aplicación). En la figura 5.4 se puede ver el efecto de aplicarlas todas al mismo tiempo.

Capítulo 5. Resultados

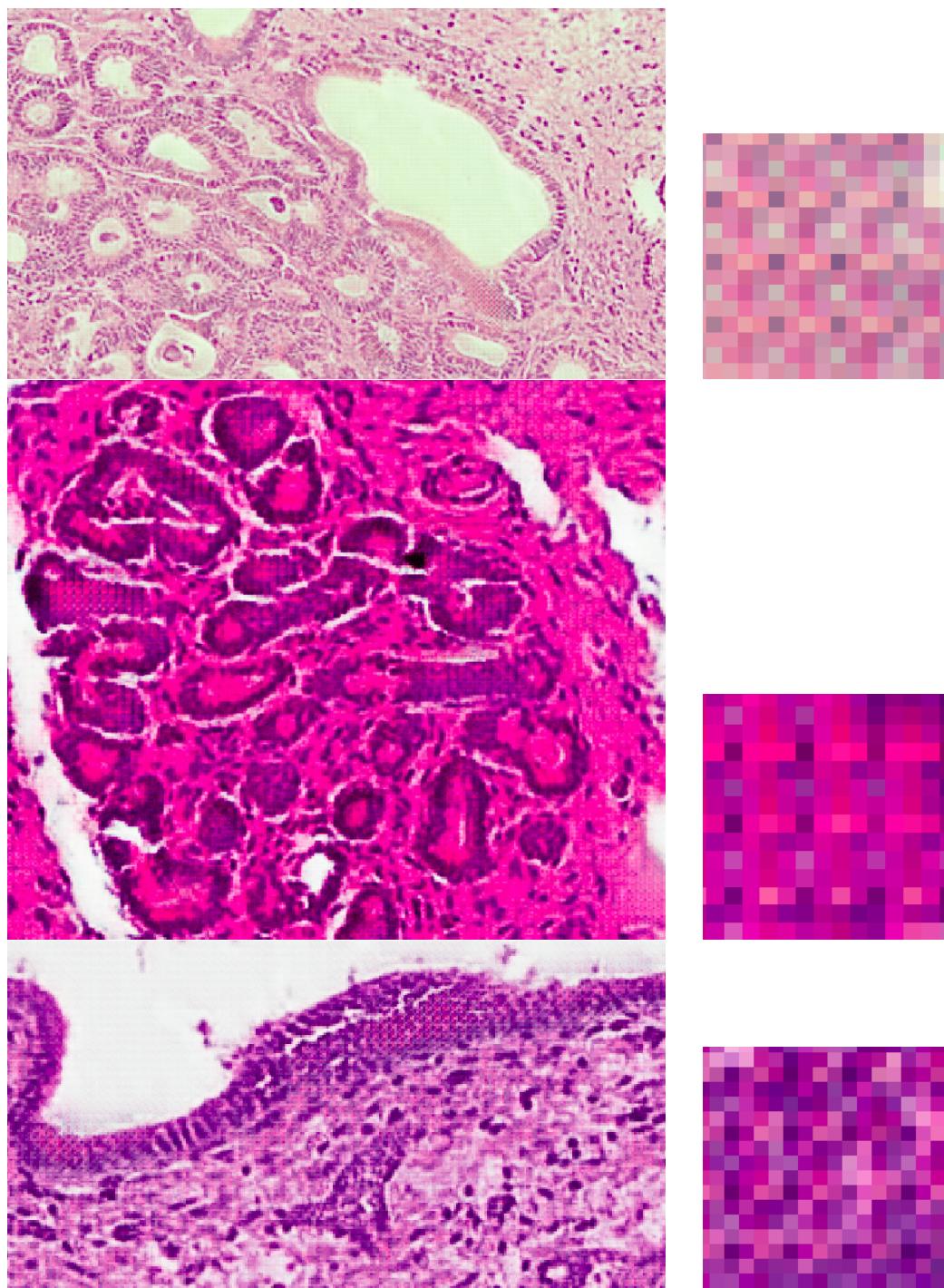


Figura 5.2: Patrón cuadriculado de 4×4 píxeles obtenido en imágenes generadas con el modelo de HistAuGAN.

5.2. Entrenamientos del modelo de fibrosis

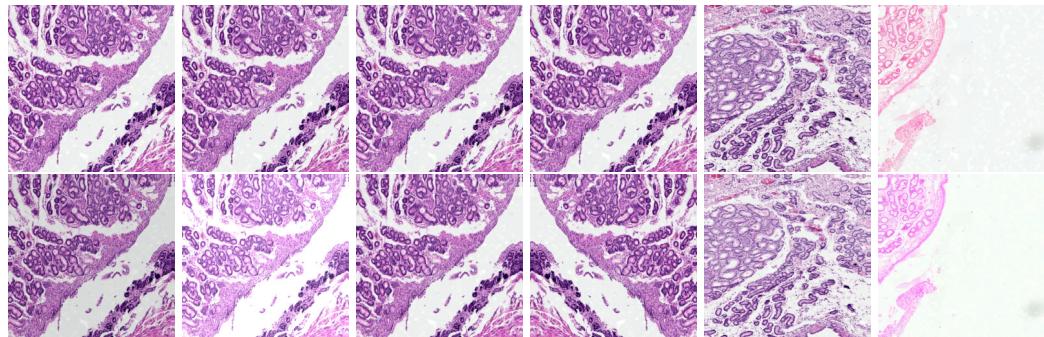


Figura 5.3: Transformaciones de datos conservadas de QA. En la fila de arriba se ven las imágenes originales. En la fila de abajo, las transformaciones aplicadas una a la vez. De izquierda a derecha: contraste, brillo, *blur*, rotación, *gamma*, HSV.

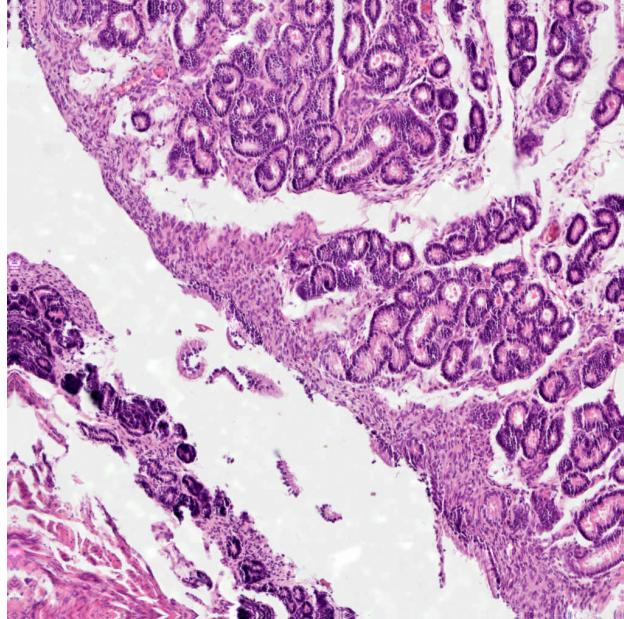


Figura 5.4: Ejemplo de imagen de F-DB transformada con las aumentaciones de datos conservadas de QA.

Capítulo 5. Resultados

5.2.2. F-DB aumentada con HistAuGAN

Con el fin de tener un modelo que generalice sobre una mayor cantidad de dominios de color, se generó una versión aumentada de F-DB, donde se transformó cada imagen desde su dominio original a los dos dominios de color restantes, mediante la red de HistAuGAN presentada en la sección anterior 5.1. Luego de esta aumentación de datos, F-DB quedó conformada por 664 imágenes, de las cuales 140 pertenecen al dominio violeta, 229 pertenecen al dominio rosado, y 295 pertenecen al rojo. A su vez, 532 de ellas pertenecen al conjunto de entrenamiento, y las 132 restantes al de validación. Ejemplos de los datos aumentados se pueden ver en la figura 5.1.

Con la base aumentada, y excluyendo las aumentaciones mencionadas en la sección 5.2.1, se entrenó el modelo segmentador de QA, de la misma forma que se describió en secciones anteriores: se entrenó el *autoencoder*, y se usaron sus pesos para inicializar el modelo segmentador.

5.2.3. *Transfer learning* entre BLCA-DB y F-DB

En la sección 3.3.3 se probó hacer *transfer learning* entre un modelo segmentador de núcleos y uno de fibrosis, de modo de mitigar supuestos problemas de inicialización del segmentador de QA. Durante el desarrollo de ese experimento se descubrió el verdadero problema, el cual no tenía que ver con la inicialización del modelo. Sin embargo, al tratarse de tareas relacionadas, es interesante ver como se desempeña una UNet entrenada de esta manera y compararla con el resto de los segmentadores de fibrosis.

Entonces, para este entrenamiento, fue utilizado el proyecto de Pytorch-UNet. Primero, el segmentador fue entrenado (como se describe en la sección 3.3.3) con la totalidad de BLCA-DB, durante 30 épocas, con datos escalados a 0.5 veces el tamaño original (es decir, imágenes de 500×500 píxeles). Este modelo alcanzó un coeficiente de Dice promedio de 0.69 sobre el conjunto de prueba de BLCA-DB.

Luego el segmentador fue re-entrenado con la totalidad de F-DB, también con imágenes escaladas a 0.5 veces su tamaño original. Esta vez, se congelaron los pesos del espacio latente y se mantuvo fijo el *learning rate* en 1×10^{-5} durante las 100 épocas que duró el entrenamiento. La reducción de tamaño de la entrada a la red durante el entrenamiento se hizo para habilitar un mayor tamaño de *batch*. En estos entrenamientos un *batch* es de 8 imágenes, y no es posible tomar más debido a limitaciones computacionales. También se realizaron estos mismos entrenamientos con las bases de datos a tamaño original; en estos casos el mayor tamaño de *batch* posible fue de 3.

5.2.4. Conjunto de prueba

Para evaluar los distintos experimentos, se utilizará un conjunto de prueba que consiste de 35 imágenes de tamaño 1000×1000 obtenidas con magnificación $10\times$. Estas fueron etiquetadas durante la generación de F-DB y no participaron de los entrenamientos. Todas presentan fibrosis.

5.2. Entrenamientos del modelo de fibrosis

#E	AD	#IB	TE	LR	DC	Duración (hs)
300	-	8	1000×1000	1×10^{-3}	0.45	3
300	QA	8	1000×1000	1×10^{-3}	0.48	3
288	HistAuGAN	8	1000×1000	1×10^{-3}	0.48	9

Tabla 5.1: Resultados de los entrenamientos del modelo segmentador de fibrosis de QA sobre los datos de prueba.

#E	AD	#IB	TE	LR	DC	Duración (hs)
10	-	3	1000×1000	1×10^{-5}	0.17	5
100	-	8	500×500	1×10^{-5}	0.52	5
130	HistAuGAN	8	500×500	1×10^{-5}	0.54	5

Tabla 5.2: Resultados de los entrenamientos del modelo segmentador de fibrosis de Pytorch-UNet sobre los datos de prueba.

Para medir el desempeño de los modelos se calculó el coeficiente de Dice (según la ecuación 3.4) de las predicciones de cada modelo sobre la base de prueba.

5.2.5. Resultados

A continuación se detallan los resultados de los entrenamientos descritos sobre el conjunto de prueba.

En las tablas 5.1 y 5.2 se muestran las siguientes características de los entrenamientos: número de épocas (#E), técnica de aumentación de datos aplicada (AD), cantidad de imágenes del *batch* (#IB), tamaño de la entrada a la red en píxeles (TE), *learning rate* (LR), el valor del coeficiente de Dice promedio sobre el conjunto de prueba (DC), y la duración del entrenamiento en horas. La tabla 5.1 incluye los resultados de los entrenamientos del modelo de QA; la tabla 5.2 muestra los de Pytorch-UNet. En la figura 5.5 se muestra la inferencia del mejor modelo sobre una imagen del conjunto de prueba.

Capítulo 5. Resultados

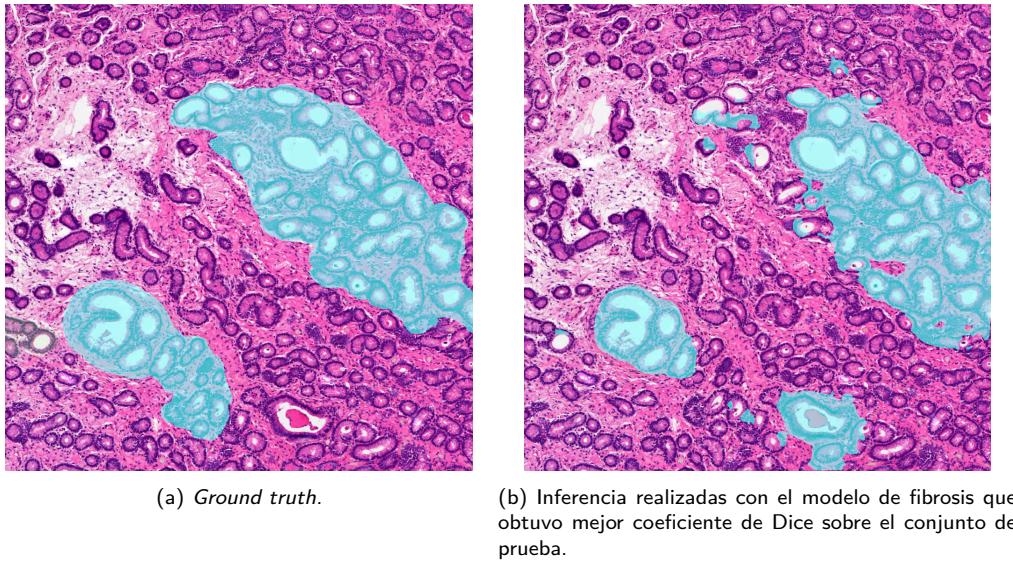


Figura 5.5: Imagen del conjunto de prueba. Inferencia del mejor modelo segmentador de fibrosis.

5.2.6. Conclusiones

Comparando los entrenamientos del modelo de QA, se puede ver que la aumentación de datos, tanto las descritas en la sección 5.2.1, como las de HistAuGAN, mejoran los resultados respecto al entrenamiento sin aumentaciones de datos, aunque levemente.

Respecto a los entrenamientos con la red de Pytorch-UNet, se observa primero que el entrenamiento con tamaño de *batch* de 3 imágenes duró 10 épocas; fue cortado por límite de tiempo, ya que luego de 5 hs no alcanzó un resultado aceptable. En comparación, el entrenamiento con tamaño de *batch* de 8 imágenes escaladas a 0.5 veces su resolución original, alcanzó los resultados que se muestran en la tabla en el mismo tiempo.

Por último, comparando los resultados entre los experimentos de QA y Pytorch-UNet, se observa que el mejor modelo es el resultante del *transfer learning* entre BLCA-DB y F-DB, entrenado con la aumentación de HistAuGAN. Por lo tanto, se puede concluir que esta técnica es pertinente a la hora de resolver el problema de segmentar fibrosis, y que HistAuGAN ayudó a mejorar el desempeño del modelo.

5.3. Extracción de información cuantitativa

En la siguiente sección se detallan los experimentos e información resultante de aplicar el flujo de trabajo presentado en el capítulo anterior. Los datos obtenidos de las máscaras de segmentación se utilizaron para llevar a cabo un análisis de la distribución, densidad, forma y color de las estructuras biológicas. En particular el análisis se concentra en la información extraída para las glándulas y núcleos. Combinado esta información junto con la obtenida del modelo de fibrosis, se abre un amplio camino a futuras investigaciones.

5.3. Extracción de información cuantitativa

5.3.1. Experimentos realizados con máscaras de glándulas

Comparación entre modelos

Durante el desarrollo del proyecto se hizo énfasis en lograr mejorar la segmentación de glándulas, ya que la calidad de la inferencia no era satisfactoria para extraer cierto tipo de características, como forma o cantidad. Como se describió en la sección 3.1, inicialmente se experimentó con el modelo de segmentación de epitelio. Posteriormente, al concluir que su inferencia no era adecuada para caracterizar la forma de las glándulas, se decidió buscar nuevos modelos de segmentación. El modelo entrenado con la base de datos de GlaS parecía ser mas adecuado para esta tarea, ya que efectivamente fue entrenado con máscaras de glándulas. Sin embargo, en la práctica este modelo demostró una alta tasa de falsos negativos, por lo que no fue conveniente para un cálculo de densidad. A modo exploratorio se optó por realizar un análisis de densidad de las glándulas, calculando la cantidad de píxeles detectados por instancia sobre área total del parche.

En la figura 5.6 se observan los histogramas en escala logarítmica, generados para diferentes WSIs, comparando ambos modelos. Estos histogramas representan el área de cada glándula individualmente. Tal como se había observado cualitativamente con las imágenes de inferencia, el histograma del modelo de epitelio posee mayor cantidad de inferencias para un gran rango de áreas en comparación con el modelo de glándulas, validando así la hipótesis inicial. Otra interpretación de esos histogramas, es que el modelo de epitelio tiende a sobre-estimar el área ocupada por las glándulas, ya que cada instancia de la inferencia puede englobar un conjunto de glándulas. En la figura 5.7 puede observarse la inferencia del modelo de epitelio para varios rangos de área. Por ejemplo en el rango comprendido de $0.06mm^2$ a $0.075mm^2$ el modelo infiere una única gran instancia, sin embargo se trata de un conjunto de glándulas aglomeradas.

Una utilidad de interés para estos datos, es prestar especial atención a aquellas imágenes con rango de áreas mayores a cierto umbral. Estas imágenes podrían representar glándulas dilatadas o nidos glandulares en caso de tener un única inferencia de gran área. Por ejemplo, en la figura 5.8 se observan aquellas muestras mayores a un umbral de 0.8 para la WSI de *Lady.svs*. Es decir aquellos parches donde el existe una instancia de glándulas cuya área sea superior al 80 % del área del parche. Se observa efectivamente que varias de las glándulas se encuentran dilatadas y muchas otras agrupadas. Como conocemos las posiciones de las imágenes dentro de la WSI podemos saber en que parte del mapa de calor se encuentran.

Comparación entre muestras de una misma yegua

Otra aplicación interesante es comparar las inferencias de dos muestras distintas de una misma yegua. Esto podría ser de utilidad cuando se tiene una yegua con historial de muestras y quiere compararse su evolución a lo largo del tiempo. O por ejemplo si se quiere compara el útero de la yegua antes y después de un acontecimiento, por ejemplo una intervención quirúrgica o tener crías.

Para las muestras disponibles se tienen dos WSIs **44-Pre.svs** y **44-D5.svs**,

Capítulo 5. Resultados

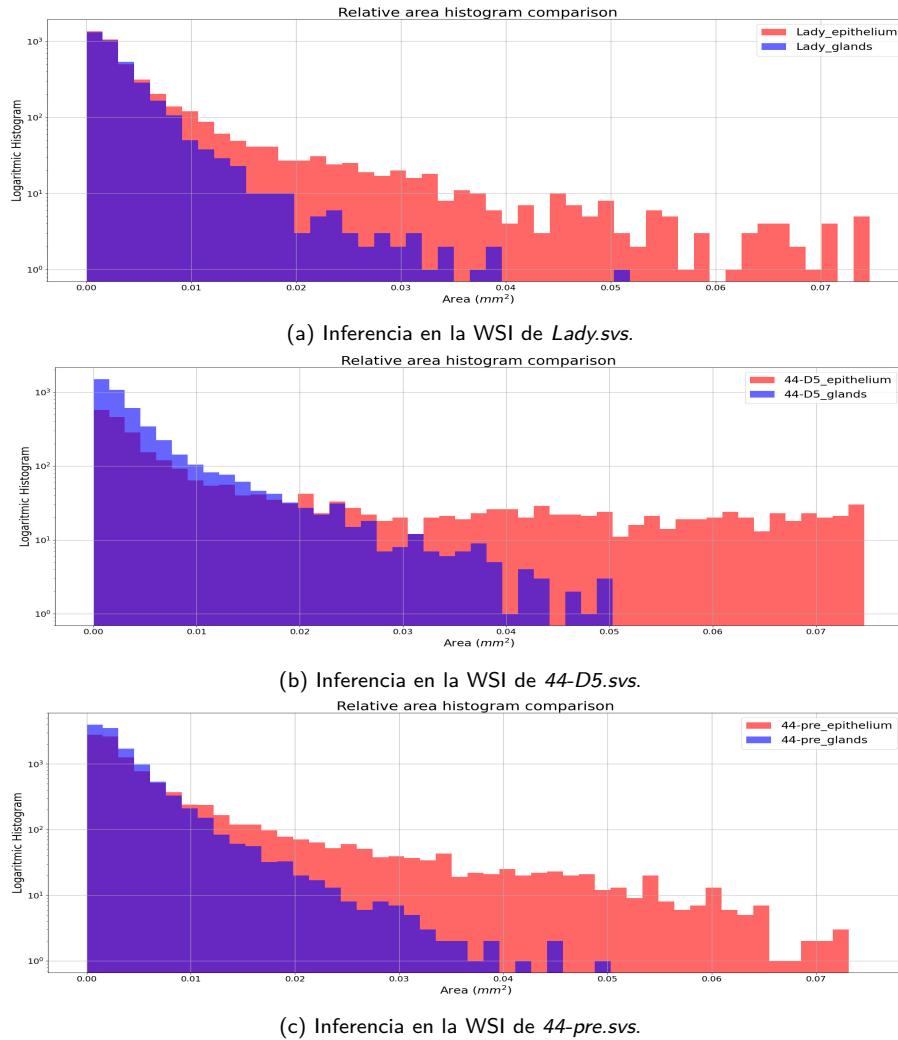


Figura 5.6: Comparación de histogramas en escala logarítmica para una misma WSI con ambos modelos de glándulas, el modelo de epitelio (en rojo) y el modelo de glándulas de GlaS (en azul).

correspondientes a la yegua 44. El sufijo D5 corresponde a que la muestra fue extraída cinco días luego de la ovulación. En esta etapa la yegua se encuentra en el diestro y por lo tanto las glándulas se presentan mas secretoras. Por otra parte el sufijo Pre corresponde a que la muestra fue extraída previo a un análisis clínico particular, sin embargo no se conoce el día del ciclo que fue extraída. En la figura 5.9 se muestran los histogramas de las glándulas para ambos modelos. Se puede observar que si bien los histogramas son distintos para cada modelo, la muestra 44-D5.svs tiene mayor densidad de glándulas que la muestra 44-pre.svs. Es decir, que la información brindada por ambos histogramas es análoga. Para complementar el análisis se incluyen las figuras 5.9c y 5.9d que corresponden a las imágenes de las muestras de 44-D5.svs y 44-Pre.svs respectivamente, a una escala de 400 μ m. Se puede ver gran densidad glandular en ambos casos, por un

5.3. Extracción de información cuantitativa

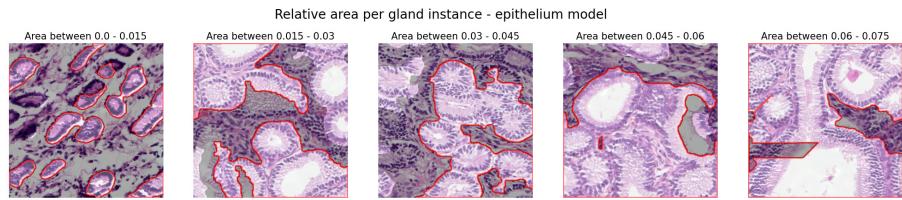


Figura 5.7: Inferencias realizadas con el modelo de epitelio que representan distintas instancias de glándulas detectadas, con valores en rangos de áreas indicados en el título de cada imagen.

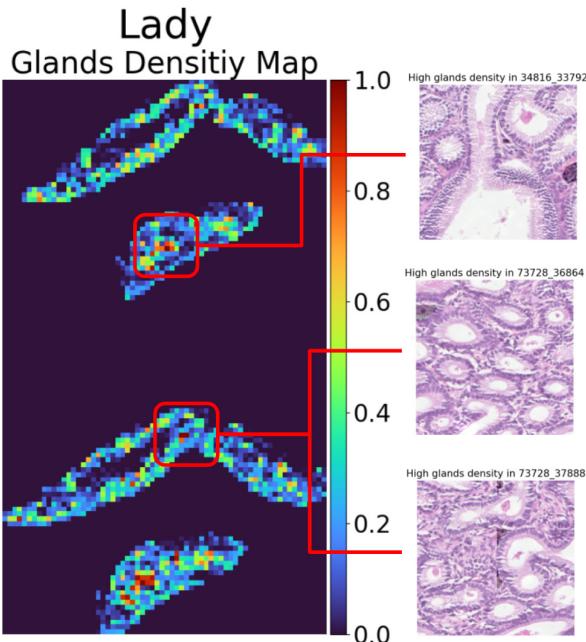


Figura 5.8: A la izquierda se observa el mapa de calor de toda la WSI para la segmentación de glándulas. A la derecha, glándulas con áreas mayores al 80 % y se señala su ubicación dentro del mapa.

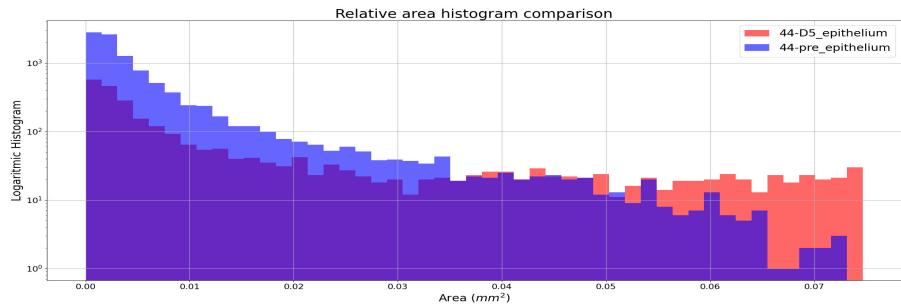
lado en la figura 5.9d se ven algunos nidos, mientras que en la figura 5.9c posee varias glándulas dilatadas.

Inferencia en distintos dominios de color

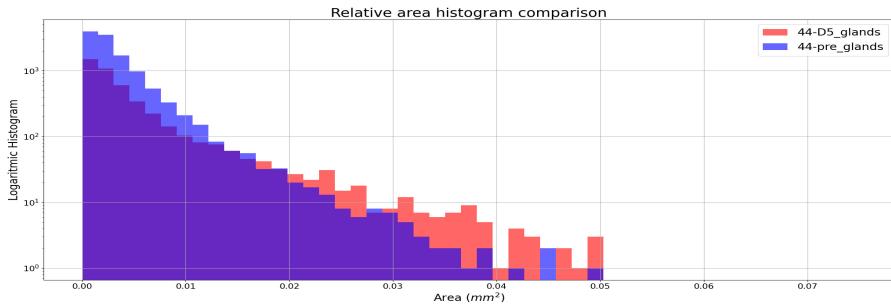
En secciones anteriores se presentaron las problemáticas surgidas a partir de la variabilidad en las tinciones de las muestras. A continuación se presenta la comparación de las inferencias entre distintos dominios de color, y como éstas presentan limitaciones en cada dominio. En esta sección se muestran resultados obtenidos con el modelo de epitelio.

En la figura 5.10 se muestran los histogramas y visualizaciones de inferencia para dos imágenes del dominio rosado. Es interesante observar como en la figura 5.10a el histograma de inferencias está concentrado en los valores de área mas pequeños y no hay inferencias de gran área. Esto se ve reflejado en la visualiza-

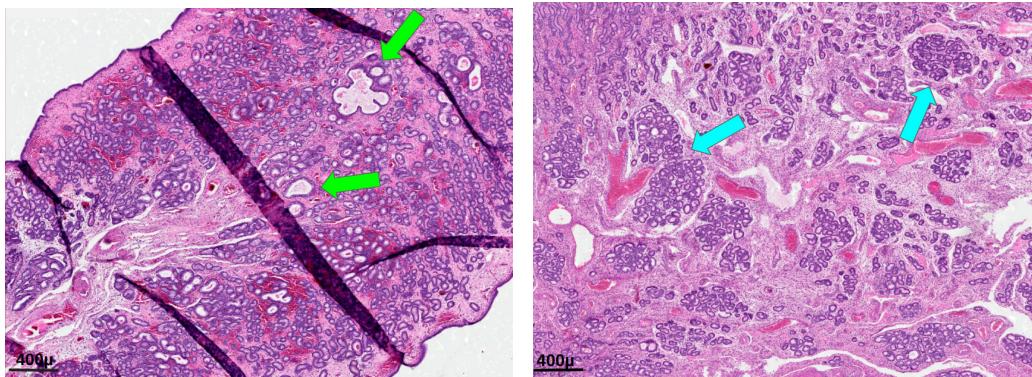
Capítulo 5. Resultados



(a) Comparación de histogramas de las WSIs 44-pre.svs y 44-D5.svs con el modelo de epitelio.



(b) Comparación de histogramas de las WSIs 44-pre.svs y 44-D5.svs con el modelo de GlaS.



(c) Imagen de 44-D5.svs con barra de escala a 400 μm . La flecha verde señalan glándulas dilatadas.

(d) Imagen de 44-pre.svs con barra de escala a 400 μm . Las flechas celestes señalan nidos en la imagen.

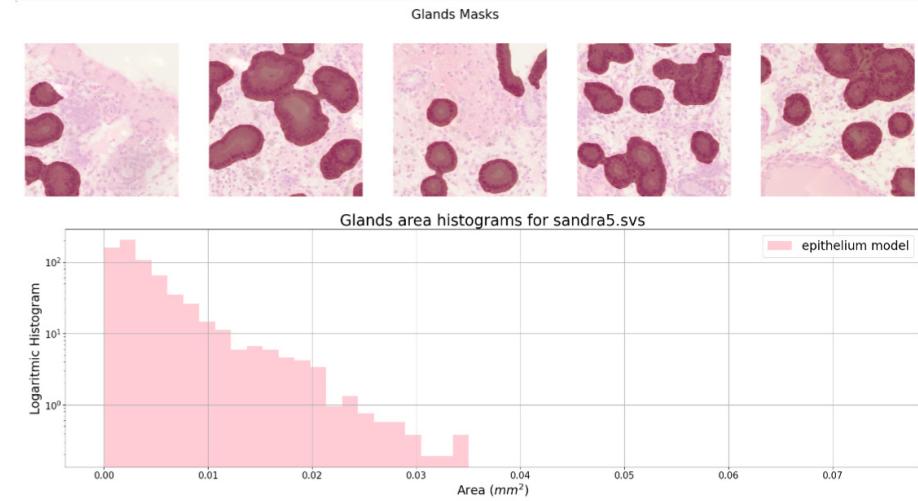
Figura 5.9: Comparación de una misma yegua en dos muestras distintas.

ción, donde las máscaras parecen tener suficiente detalle en su contorno. Esto se puede deber a que, en esta muestra, las glándulas parecen estar muy dispersas, a diferencia de los ejemplos anteriores.

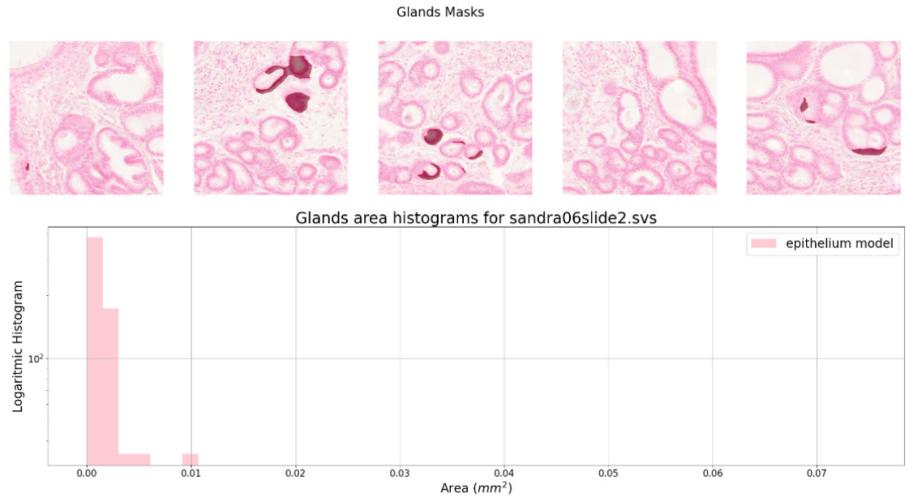
Por otra parte en la figura 5.10b el histograma está concentrado en glándulas de mucho menor área. Mirando solo esta información el usuario podría asumir que se trata de una muestra con baja densidad glandular. Sin embargo, en las imágenes se observa que las glándulas están muy dilatadas y las inferencias son muy escasas. Por lo tanto esa inferencia no es buena para esta muestra y el histograma induce a una conclusión errónea.

Por otro lado, también se hicieron inferencias en imágenes del dominio rojo.

5.3. Extracción de información cuantitativa



(a) Inferencia para la WSI de sandra5.svs e histograma de densidad de glándulas.



(b) Inferencia para la WSI de sandra06slide2.svs e histograma de densidad de glándulas.

Figura 5.10: Inferencias con el modelo de epitelio para muestras del dominio rosado. En la figura a) se ven imágenes del dominio rosado con baja densidad glandular, mientras que la figura b) muestra glándulas dilatadas. El modelo muestra dificultad para detectar glándulas de gran área.

En la figura 5.11 se tiene la visualización del histograma junto con inferencias de diferentes rangos área. Se observa que las inferencias en este dominio de color tienden a ser menos precisas, tomando conjuntos de glándulas. Es por esto que los valores del histograma tienden a dispersarse sobre todos los rangos, sin embargo no son buenas inferencias.

Capítulo 5. Resultados

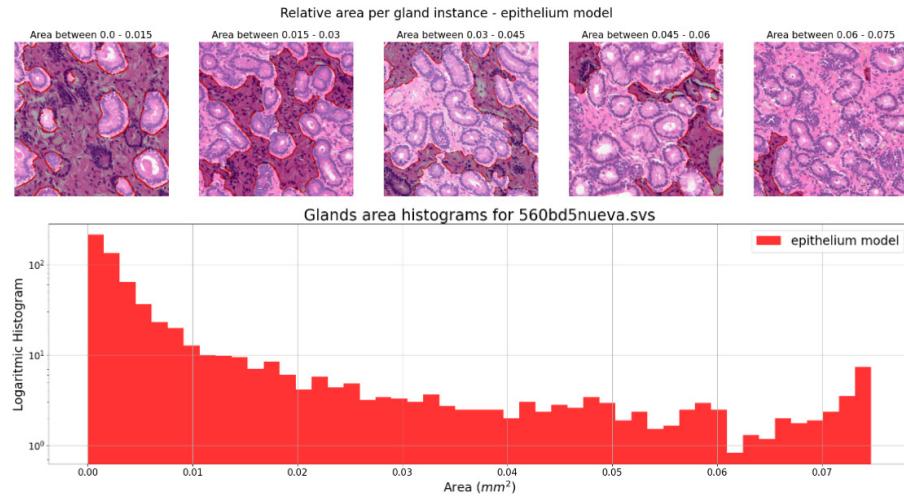


Figura 5.11: Inferencia con el modelo de epitelio para la WSI de 560bd5nueva.svs del dominio rojo, e histograma de densidad de glándulas.

5.3.2. Experimentos realizados con máscaras de núcleos

La primera etapa necesaria para extraer información de núcleos es el cálculo de características a partir de las imágenes y las máscaras de segmentación. Además de las características introducidas en la sección 4.4.2 (área, circularidad, nivel de intensidad y 3 métricas para evaluar textura), también se calcula el perímetro de cada núcleo, y se obtienen los píxeles que conforman su contorno. Todos los datos son almacenados en una estructura de datos tabular, junto con el nombre de la imagen donde se encuentra dicho núcleo, para futuras referencias.

El procedimiento de extracción de datos de 9 439 núcleos (100 imágenes en este caso), demora 26.45s; es decir, un aproximado de 0.0028s por núcleo. Una WSI suele tener unas 1600 imágenes y cada imagen puede tener 100 núcleos. Lo que daría un total de 7.5 minutos para extraer las características de los núcleos, por WSI. En resumen, se observa que es una tarea que requiere mucho tiempo. En caso que se quiera analizar las mismas imágenes en más de una ocasión, sería recomendable guardar la información de los núcleos, por ejemplo en un archivo *csv*, para evitar tener que volver a calcularlas. Queda a criterio del usuario incorporar esta modificación.

A efectos de obtener resultados con la implementación desarrollada se extrajeron características de núcleos de 300 imágenes provenientes de 3 WSIs (100 imágenes de cada una), todas pertenecientes al dominio violeta. En total se calcularon características de 31 890 núcleos. Vale aclarar que los datos fueron calculados a partir de imágenes con nivel de magnificación de 40×, por lo que los datos de perímetro y área son comparables siempre que se trabaje con el mismo nivel de magnificación. También es relevante notar que las características de textura, que utilizan el histograma normalizado de los niveles de intensidad del núcleo en escala de grises, pueden variar entre un dominio y otro; por lo que tampoco serían comparables estas magnitudes para imágenes de distintos dominios.

5.3. Extracción de información cuantitativa

Una vez que se contó con los datos, el primer paso consiste en filtrar los núcleos con área muy pequeña. Estas instancias se observó que se trataban de detecciones espúreas por lo que se optó por desestimarlas. Luego se procedió a normalizar en el rango $[0, 1]$ las características halladas para la clasificación. Se excluyeron los percentiles menores a 0.1 y mayores a 99.9, para cada característica, de forma de evitar valores atípicos. Luego se graficaron las distribuciones de cada una; estas gráficas se presentan en la figura 5.12. Analizando las distribuciones de a una no se observa ninguna que permita señalar una clase de núcleos claramente diferenciable del resto.

A pesar de que a partir de los histogramas no se observa ninguna característica que permita diferenciar alguna clase de núcleos, se hizo la prueba de fijar un umbral para la circularidad y observar aquellos núcleos que quedaban por encima de ese valor. Tal como se comentó previamente, los linfocitos son bastante redondos, por lo que se esperaba que esta clasificación arrojara resultados positivos.

El umbral elegido corresponde al percentil 90 de los valores de circularidad. En la figura 5.13 se observan los núcleos detectados con este método; también se señaló en la imagen original un núcleo redondo no detectado. La clasificación obtenida no fue satisfactoria, algunos linfocitos quedaban afuera de la selección, mientras que otros que no eran linfocitos quedaban adentro. Esto se puede deber a que la máscara no se ajustaba lo suficiente a la forma del núcleo. También puede hacer falta otras formas de caracterizar la circularidad para complementar la clasificación.

Se probó variar el valor del umbral seleccionando percentiles más altos y más bajos, para ver si todos los posibles linfocitos quedaban dentro de la detección. Sin embargo no se observaron mejoras. Se concluyó que hacían falta más variables para obtener una mejor clasificación, como por ejemplo relacionadas a la intensidad. En la imagen se puede ver que algunos de los núcleos seleccionados tienen una coloración muy tenue.

Se continúo el estudio comparando dos a dos las características. El objetivo era el mismo, lograr visualizar clases de núcleos diferenciables, en este caso en las distribuciones conjuntas. En la figura 5.14 se presentan las distribuciones dos a dos. No se observa para ninguna combinación de dos características una separación entre clases de núcleos. Se prosiguió el análisis combinando más características para lograr una clasificación. Para ello se probó un algoritmo de agrupación.

Con el fin de evaluar el desempeño del algoritmo, se confeccionó una base de datos de prueba. Se tomaron 6 imágenes de cada WSI de las que se trajeron las características de los núcleos, y se identificaron aquellos que podían llegar a ser linfocitos. Se señalaron con una flecha sobre la imagen estos núcleos, tal como se muestra en la figura 4.5. Es importante mencionar que las características de los núcleos de prueba también fueron normalizadas, teniendo en cuenta los rangos establecidos según los datos de entrenamiento.

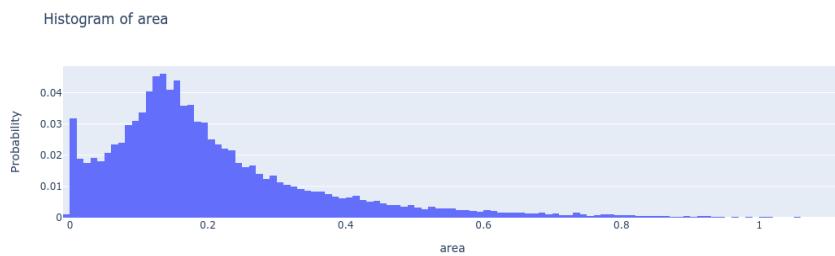
El algoritmo de agrupación implementado se trata de un k -means, implementando con la librería *scikit-learn*. Se probaron 3 modelos distintos variando la cantidad de clases a detectar, tres, cinco u ocho. Se observó que en el caso del modelo con tres clases, había una clase que detecta la mayoría de los linfocitos pero incluía

Capítulo 5. Resultados

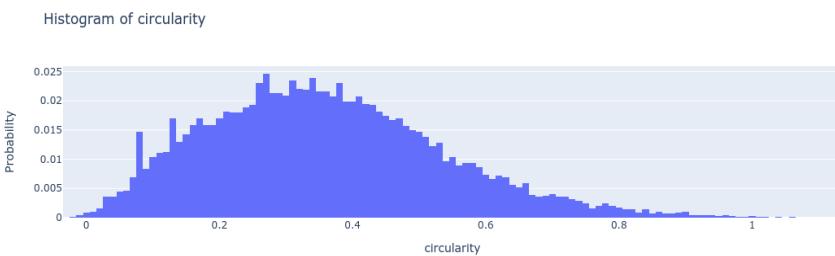
también muchos otros núcleos que no correspondían. Por otro lado, al realizar inferencia con el modelo de ocho clases se observó que eran demasiadas y no había una clase principal que se pudiera asociar a los linfocitos. Por último, el modelo con cinco clases fue el que obtuvo mejores resultados. En la figura 5.15 se puede ver un ejemplo de la inferencia realizada con este modelo. Sin embargo, hay dos clases que se asocian con linfocitos; de acuerdo a los colores vistos en la imagen de referencia, estas serían la roja y la rosada. En términos generales se observa que los núcleos presentes en la clase rosada son más circulares, sin embargo presentan texturas más variadas que los que quedan comprendidos en la clase roja.

Las pruebas realizadas con las segmentaciones de núcleos fueron de carácter exploratorio. El análisis comenzó con el filtrado de las detecciones espúreas y la normalización de las características. Luego se observó que fijar umbrales sobre una de las características o sobre dos de ella en simultáneo, no era suficiente para reconocer linfocitos. Por último, el algoritmo de agrupación utilizado fija umbrales sobre todo el conjunto de características y aún así tampoco se lograron clasificar correctamente los linfocitos. Para poder lograr identificar estos núcleos se deberían explorar otras técnicas. En la sección 6.2 se comenta acerca de qué otras pruebas se podrían realizar para continuar con esta investigación.

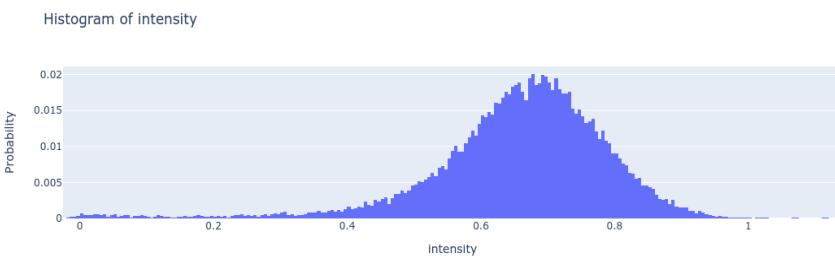
5.3. Extracción de información cuantitativa



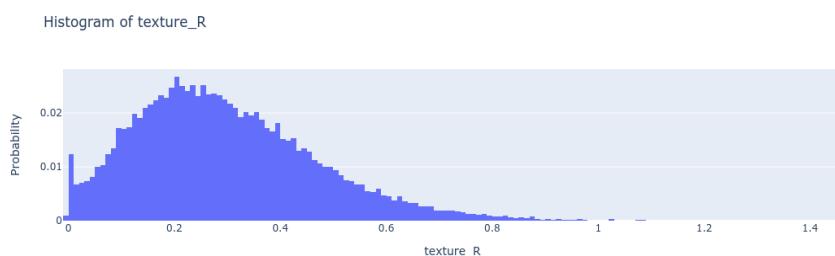
(a) Distribución del área.



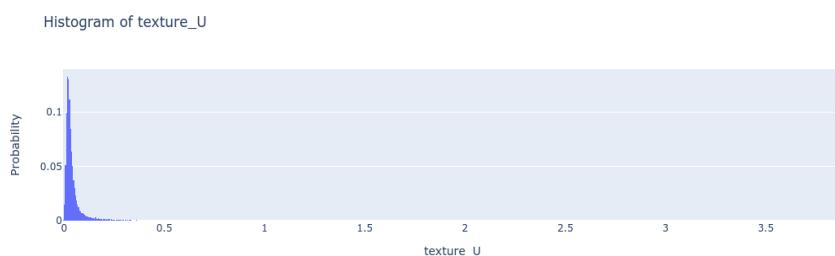
(b) Distribución de la circularidad.



(c) Distribución de la intensidad media.

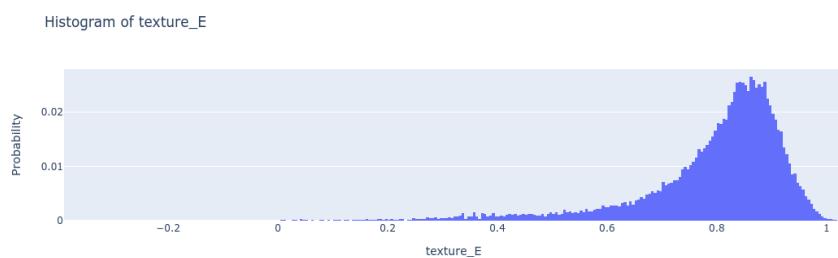


(d) Distribución de la textura R.



(e) Distribución de la textura U en escala logarítmica.

Capítulo 5. Resultados



(f) Distribución de la textura E.

Figura 5.12: Histograma de las características halladas para la clasificación de núcleos, normalizadas en el rango [0, 1].

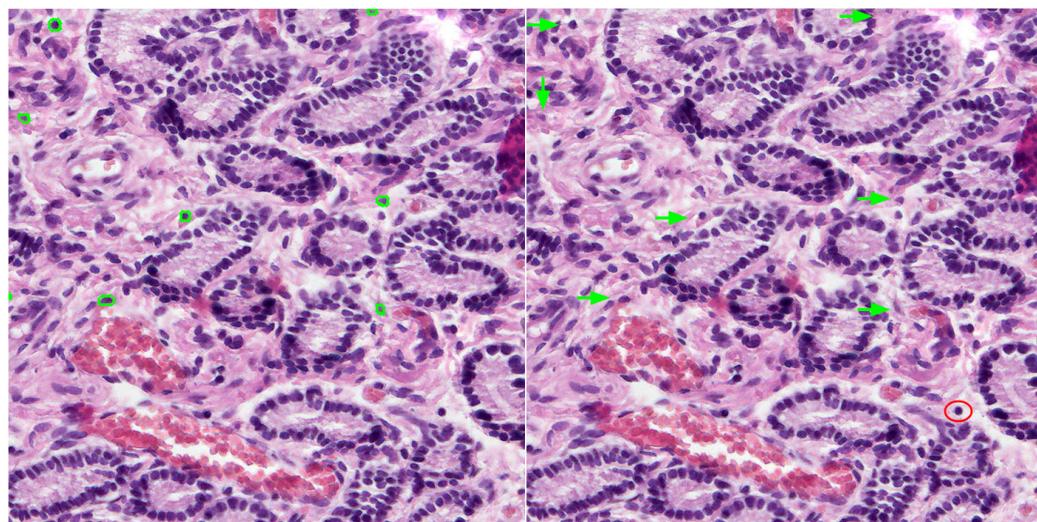


Figura 5.13: En la imagen de la izquierda, detección de núcleos con valor de circularidad mayor al percentil 90. En la derecha, parche original con los núcleos detectados señalados con flechas verdes. En la misma imagen se marcó con un círculo rojo otro núcleo redondo no detectado con este método.

5.3. Extracción de información cuantitativa

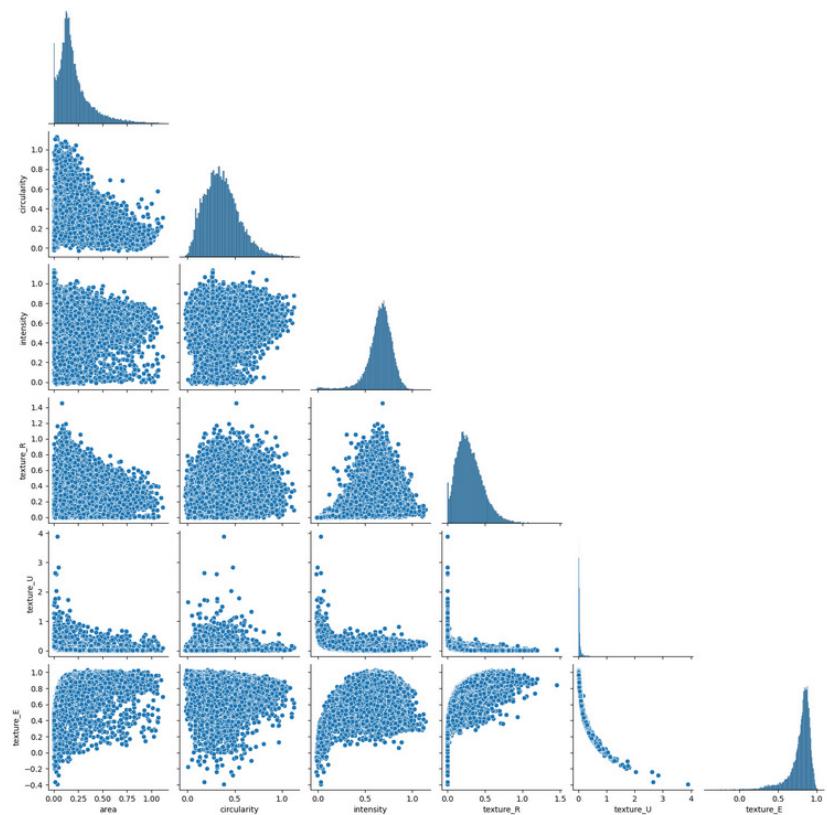


Figura 5.14: Distribución de las características de núcleos tomadas de a 2. Se puede notar que las gráficas ubicadas en la diagonal representan la distribución de cada característica. No se observa para ninguna combinación de dos características una separación entre clases de núcleos.

Capítulo 5. Resultados

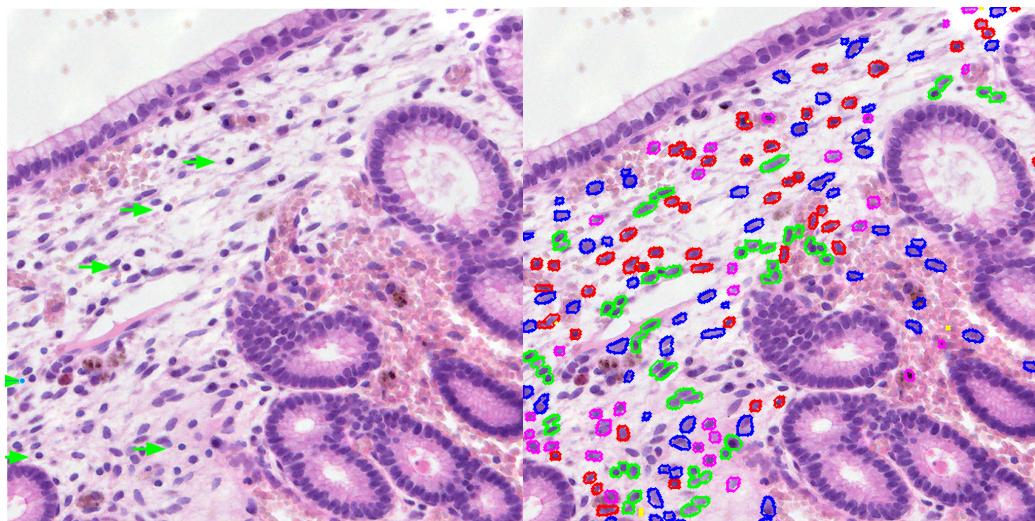


Figura 5.15: La figura de la izquierda muestra una imagen del conjunto de prueba con los linfocitos señalados. A la derecha, los resultados obtenidos con el modelo k -means entrenado con 5 clases. Las clases predichas se pintan cada una con un color distinto. Se observa que los linfocitos señalados quedan comprendidos en las clases rosada y roja.

Capítulo 6

Conclusiones

6.1. Conclusiones generales

Este proyecto de fin de carrera comienza con la interiorización en la temática del análisis clínico de las biopsias endometriales de yeguas, la revisión del estado del arte del procesamiento de imágenes clásico y modelos de aprendizaje automático en el área de las imágenes médicas, y el estudio de las herramientas y metodologías utilizadas habitualmente. La información estudiada está referida a la medicina humana. A diferencia de esta última, donde las investigaciones en el área están en pleno crecimiento, en el campo de la veterinaria no se encontró bibliografía que indicara que se estuviera implementando técnicas de aprendizaje automático para el estudio de las muestras biológicas. De todas formas, las metodologías y las técnicas aplicadas son análogas, ya que en la mayoría de los casos se tratan tejidos biológicos de similares características.

Se encontraron múltiples herramientas para trabajar con este tipo de imágenes. Varias de estas están centradas en la visualización y etiquetado de datos, y en muchos casos también cuentan con funcionalidades para obtener medidas de forma manual. Otra rama de herramientas más reciente, incorpora el entrenamiento de modelos de aprendizaje automático por medio de una interfaz gráfica, buscando evitar que el usuario deba tener conocimientos de programación para entrenarlos. Si bien estas aplicaciones son intuitivas y completas en cuanto a sus capacidades, contar con conocimientos de programación ayuda a adaptar los modelos al problema a tratar, modificando aspectos como la arquitectura, métricas y parámetros utilizados.

De todas las herramientas exploradas se destaca la librería FAST. Creada especialmente para el análisis de imágenes del área médica, es muy completa en cuanto a las tareas que permite realizar, es compatible con muchos formatos, está optimizada para el manejo de recursos, y por sobre todas las cosas, está en continuo desarrollo, incorporando mejoras y nuevas funcionalidades. Además, en las ocasiones que fue necesario recurrir a sus creadores por consultas acerca de como utilizar algunos módulos, sus respuestas siempre fueron claras y de mucha ayuda.

En cuanto a las imágenes de trabajo se constató una notable presencia de

Capítulo 6. Conclusiones

artefactos en este tipo de imágenes. El método de generación de las láminas es un proceso manual y las muestras obtenidas pueden alterarse a causa de variaciones menores; como por ejemplo el pliegue del tejido, que sea demasiado fino y se desgarre, o que quede algún residuo comprendido entre la muestra y el portaobjetos. Para un humano estos artefactos no son difíciles de detectar y desestimar. Sin embargo, para un algoritmo de detección estos desperfectos podrían acaparar el foco de atención de las muestras. Por otra parte, las WSIs son tan grandes que es posible descartar las zonas donde se observan estos artefactos y continuar analizando el resto. De todas formas se observó que su presencia es muy alta, y podría decirse que todas las muestras contienen al menos un desperfecto.

A pesar de ello, se pudieron aplicar modelos de segmentación pre-entrenados con otros tejidos, y un modelo entrenado con datos recopilados para el proyecto. Los modelos pre-entrenados se emplearon sin modificaciones y sus resultados fueron satisfactorios para la segmentación de glándulas y núcleos, tal como se describió en el capítulo 5. Se observa que una mejora posible en su funcionamiento podría ser un ajuste fino para que aprendan a reconocer el tejido y los dominios de trabajo.

Por otro lado, el modelo de segmentación de fibrosis fue trabajado en profundidad. Los cambios introducidos a la estructura base, provista por el *framework* de QuickAnnotator, dieron lugar a un conocimiento acabado del modelo (UNet) y las métricas adecuadas para este trabajo (la entropía cruzada y el coeficiente de Dice). De los cambios introducidos es oportuno mencionar que la inicialización de los pesos de la red, con una base de datos que contiene muchas imágenes etiquetadas y fue creada para realizar una tarea similar (segmentación de núcleos), fue una buena práctica y mejoró los resultados de entrenamiento. Se puede ver la tabla 5.2 para mayor referencia.

Otra técnica que también contribuyó a la mejora de los resultados fue aplicar normalización de color en las imágenes de entrenamiento. Además, debido a que cada imagen se transformaba a los otros dos dominios de trabajo, se logró de esta forma triplicar la base de datos. HistAuGAN presentó resultados muy sorprendentes, a pesar de los patrones cuadriculados generados artificialmente por este tipo de redes. Las imágenes generadas son realistas y representan de forma satisfactoria las tonalidades de los distintos dominios. Sin embargo, se puede concluir que la mejora introducida por esta técnica no es lo suficientemente significativa en comparación con el trabajo que conlleva (una diferencia de 0.02 en el coeficiente de Dice de *test*). Además, se notó una clara diferencia en los tiempos de entrenamiento en el modelo de QA. En la tabla 5.1 se observa que cuando se introduce la aumentación de datos con HistAuGAN se triplican los tiempos de entrenamiento, lo cual tiene sentido por el aumento en la cantidad de datos.

Adicionalmente, también se modificó la base de datos etiquetada de fibrosis. Los inconvenientes en el entrenamiento de los modelos determinaron que la cantidad de imágenes sin etiquetas positivas entorpecían el aprendizaje de los algoritmos, lo que llevó a remover estas imágenes de la base de datos. Fue esta modificación la que permitió el entrenamiento. La función de costo utilizada no estaba diseñada para ser entrenada con datos de estructuras anómalas, como lo es la fibrosis; las imágenes con etiqueta completamente negativa hacen que los mo-

6.1. Conclusiones generales

delos aprendan rápidamente que predecir todos los píxeles como negativos decrece la *loss* y una vez que alcanzaban este estado no se modifican sus pesos.

Cuando por fin se logró entrenar los modelos de segmentación de fibrosis, los mejores valores de Dice reportados son del orden de 0.5. Vale recordar que el coeficiente de Dice toma valores en el rango [0, 1], siendo 1 una segmentación óptima. Al observar las inferencias en la figura A.5, se deduce que el modelo aprende a reconocer glándulas dilatadas o nidos glandulares tomando como referencia la lúmina de las glándulas. Parece ser que el nivel de magnificación utilizado (10×) no fue suficiente para que aprendiera a reconocer las fibras de colágeno ubicadas alrededor de las glándulas, que es de hecho lo que origina la patología. Una mejora al modelo podría ser etiquetar datos a otro nivel nivel de magnificación, etiquetando únicamente las fibras, para que aprenda a reconocer esta estructura.

Otra observación al respecto del entrenamiento del modelo de fibrosis es que la ejecución del mismo debió ser realizada en ClusterUY. Los recursos computacionales personales no fueron suficientes para soportar el proceso. El uso del ClusterUY fue una herramienta muy accesible y beneficiosa para el proyecto. Su manejo fue sencillo y permitió ejecutar los entrenamientos en tiempo y forma.

En una etapa posterior a los modelos de segmentación se trabajó con las máscaras generadas por estos. Los resultados obtenidos del flujo de trabajo permitieron concluir que las segmentaciones de los modelos, principalmente con el modelo de epitelio, no eran suficientemente precisas como para obtener métricas en base a la forma de las estructuras, a pesar de los refinamientos subsiguientes. Las evaluaciones realizadas en la sección 5.3.1 permiten ver que los segmentadores de glándulas no tienen buenos resultados a la hora de separar instancias individuales; al hacer un análisis de densidad glandular sobre la WSI, no es posible determinar si un alto valor de densidad se debe a que hay muchas glándulas, o las glándulas que hay están dilatadas.

En cuanto a las máscaras de núcleos se podría agregar que las características halladas no fueron suficientes para lograr la clasificación de linfocitos buscada. Una mejora a realizar podría ser incorporar más características, por ejemplo algunas relacionadas al color, ya que las halladas hacen referencia a la imagen en escala de grises.

Más allá de las particularidades de los resultados experimentales obtenidos con los modelos de núcleos y glándulas, se quiere destacar que estos resultados fueron posibles gracias al flujo de trabajo punta a punta implementado durante el proyecto. Este integra las tareas, modelos de aprendizaje automático, *frameworks* y técnicas de trabajo necesarias para el estudio de las biopsias endometriales, en un mismo entorno y en una única ejecución.

Por último, queremos destacar la colaboración permanente de la Doctora en Veterinaria María José Estradé, que fue quien vio la posibilidad de incorporar metodologías de procesamiento de imágenes en el análisis clínico de las muestras biológicas, y la referente técnica en materia de equinos. Su participación activa y orientadora demostró su interés real en el desarrollo de herramientas que complementen el trabajo de los especialistas.

En la medicina humana son notorios los avances de investigación en *frameworks*

Capítulo 6. Conclusiones

y algoritmos de aprendizaje automático, mientras que las investigaciones en el área de la veterinaria se encuentran naturalmente demoradas. Este trabajo cumplió con los objetivos generales planteados inicialmente en cuanto a estudiar la posibilidad de incorporar nuevas técnicas para el análisis de biopsias endometriales de yeguas. Los resultados obtenidos son una primera versión de la herramienta, que sin dudas tienen lugar a mejoras para seguir construyendo sobre la implementación ya realizada.

6.2. Trabajo a futuro

Durante el desarrollo del proyecto se identificaron varias áreas que podrían mejorar aspectos de la implementación, conducir a resultados más precisos, y abrir nuevas líneas de investigación no exploradas.

Algunas de estas ideas se basan en mejorar el flujo de trabajo general para que este se ensamble de manera más óptima. Por ejemplo, realizar la inferencia y el procesamiento de las máscaras en un mismo bloque, o acelerar la inferencia de las mismas logrando aumentar el tamaño de *batch* en la librería FAST. En relación a la visualización, se podría volver a unir las máscaras de segmentación, alinearlas y guardarlas en un formato *tiff* piramidal. De esta manera, sería posible superponer la WSI y su máscara para una mejor visualización.

También se podrían realizar tareas en pos de mejorar la calidad de las máscaras; por ejemplo, utilizar las detecciones de núcleos y buscar el contorno de las glándulas que no fueron detectadas. En muchos casos, la disposición de los núcleos del epitelio glandular hace evidente que pertenecen a una glándula y no a otra estructura. Otro refinamiento posible podría ser utilizar la segmentación de glándulas, detectando aquellas que presentan área mayor a cierto umbral, para complementar la detección de fibrosis y corregir la segmentación.

Se observa que la falta de datos etiquetados limitó el alcance de los resultados del proyecto. Una línea de investigación posible podría ser replicar el método de etiquetado de forma no supervisada implementado en la creación de la base BLCA-DB [51] para etiquetar linfocitos. Éste método permitió crear uno de los repositorios de segmentación de tinciones H&E más grandes y disponibles públicamente.

Otra forma distinta de mitigar el problema de la escasez de datos etiquetados podría ser modificar el enfoque del modelo de segmentación de fibrosis, y entrenar un modelo de detección de anomalías. La fibrosis se trata de una patología, por lo que no está presente en todas las yeguas, ni en todas las WSIs. Los modelos de detección de anomalías más recientes se entrena de forma de aprender la distribución normal de los datos, por lo que no hace falta generar una base con datos segmentados.

Por último, sin dudas que aplicar alguna técnica de interpretabilidad al modelo de segmentación de fibrosis complementaría el análisis realizado, y confirmaría la sospecha al respecto de que el modelo se centra sobre las glándulas y no sobre los fibra de colágeno, al momento de realizar la predicción.

Apéndice A

Experimentos del modelo de fibrosis

A.1. Primeros experimentos del modelo de fibrosis

El etiquetado de datos de fibrosis fue un proceso que se basó en anotar algunas imágenes y entrenar un modelo de forma iterativa. Para estos entrenamientos se utilizó una NVIDIA GeForce GTX 1650 Ti Mobile con 4Gb de memoria. Los parámetros utilizados para este entrenamiento fueron `batchsize = 32`, `numearlystopepochs = 100` (cantidad de épocas sin disminuir la *loss* de validación antes de parar el entrenamiento) y `pclassweight` aproximadamente 0,95. Este ultimo parámetro consiste en un coeficiente para balancear los datos positivos y se calcula en cada entrenamiento. Las siguientes figuras corresponden a los primeros entrenamientos de este modelo. Los experimentos son parcialmente comparables ya que para cada entrenamiento se agregaban más imágenes etiquetadas. Es decir que cada iteración del modelo posee un *dataset* con mas imágenes que el anterior. A pesar de este punto se señalan algunas observaciones.

En primer lugar las primeras dos iteraciones del modelo presentan curvas de entrenamiento con inclinación más acentuada que las siguientes iteraciones, y pendiente negativa. Para las iteraciones tres, cuatro, cinco y seis, las métricas de entrenamiento del modelo no varían significativamente. Por otra parte, la función de costo en validación alcanza picos de forma aleatoria, mientras que las métricas de *accuracy*, tasa de verdaderos positivos y tasa de verdaderos negativos son mucho mas ruidosas que las de entrenamiento. Este fue el primer indicio que el modelo no estaba entrenando de forma correcta. A pesar de esto se utilizó la base de datos generada, debido a que todas las anotaciones fueron realizadas de forma manual y no se utilizó la asistencia del modelo. Por ultimo, se observa que para la última iteración del modelo (la número seis) la tasa de verdaderos positivos es notoriamente inferior a las iteraciones cuatro y cinco. Esto puede deberse a que en la última iteración se agregaron imágenes con coloración notoriamente distinta a las imágenes anteriormente etiquetadas, imágenes pertenientes a los otros dominios de trabajo.

Apéndice A. Experimentos del modelo de fibrosis

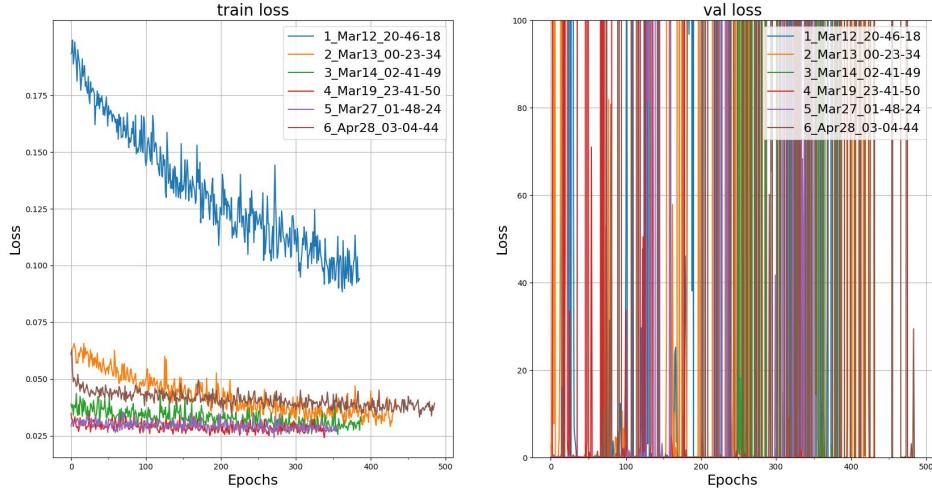


Figura A.1: Función de costo en entrenamiento y validación. La métrica utilizada se trata de la entropía cruzada.

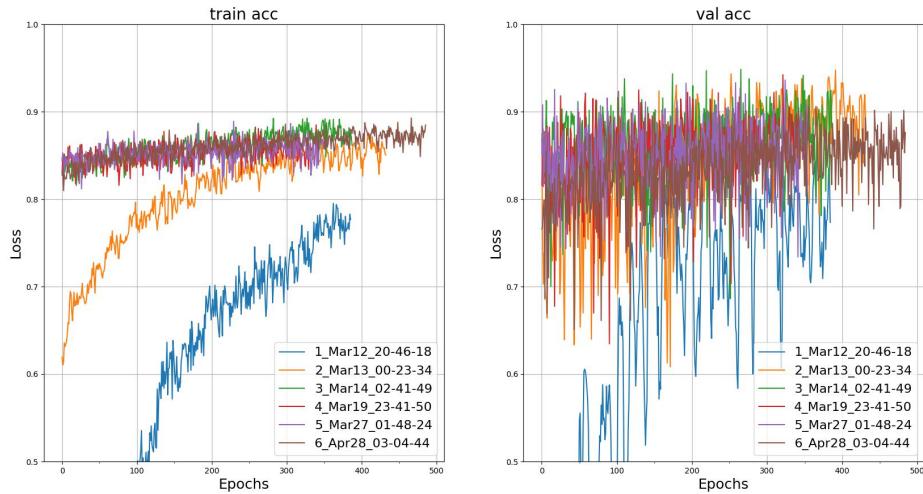


Figura A.2: Métrica de *accuracy* para entrenamiento y validación.

A.2. Migración de la base de datos de QuickAnnotator

Esto implicó migrar el proyecto anterior cambiando el tamaño de las *ROIs* a 1000×1000 pixeles, mediante el uso del script *import_annotations_cli.py* incluido en la base de código de QA. Este paso es necesario ya que genera la estructura de directorios y archivos que necesitan los *scripts* de entrenamiento para funcionar correctamente.

A.2. Migración de la base de datos de QuickAnnotator

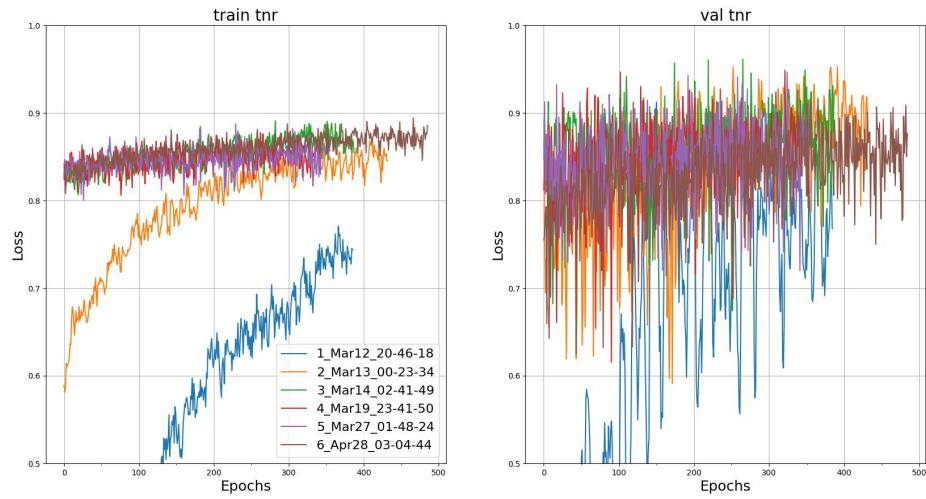


Figura A.3: Tasa de verdaderos negativos para entrenamiento y validación.

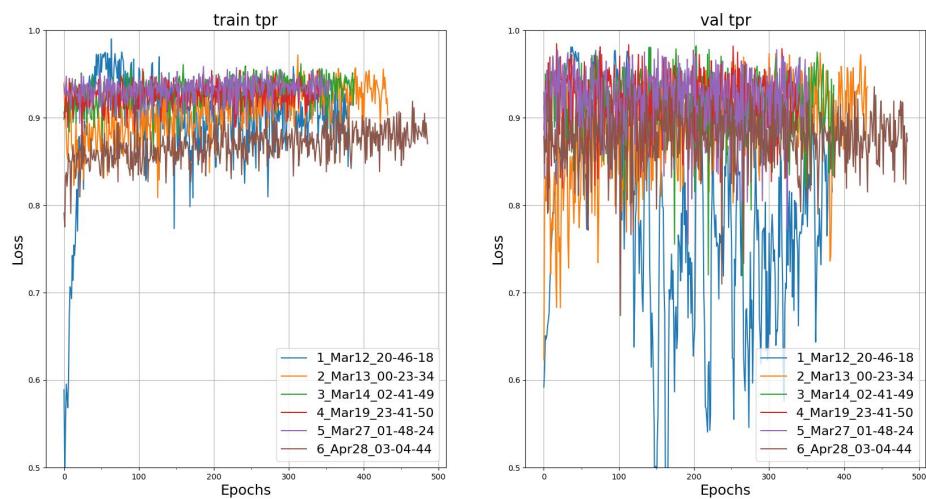


Figura A.4: Tasa de verdaderos positivos para entrenamiento y validación.

Apéndice A. Experimentos del modelo de fibrosis

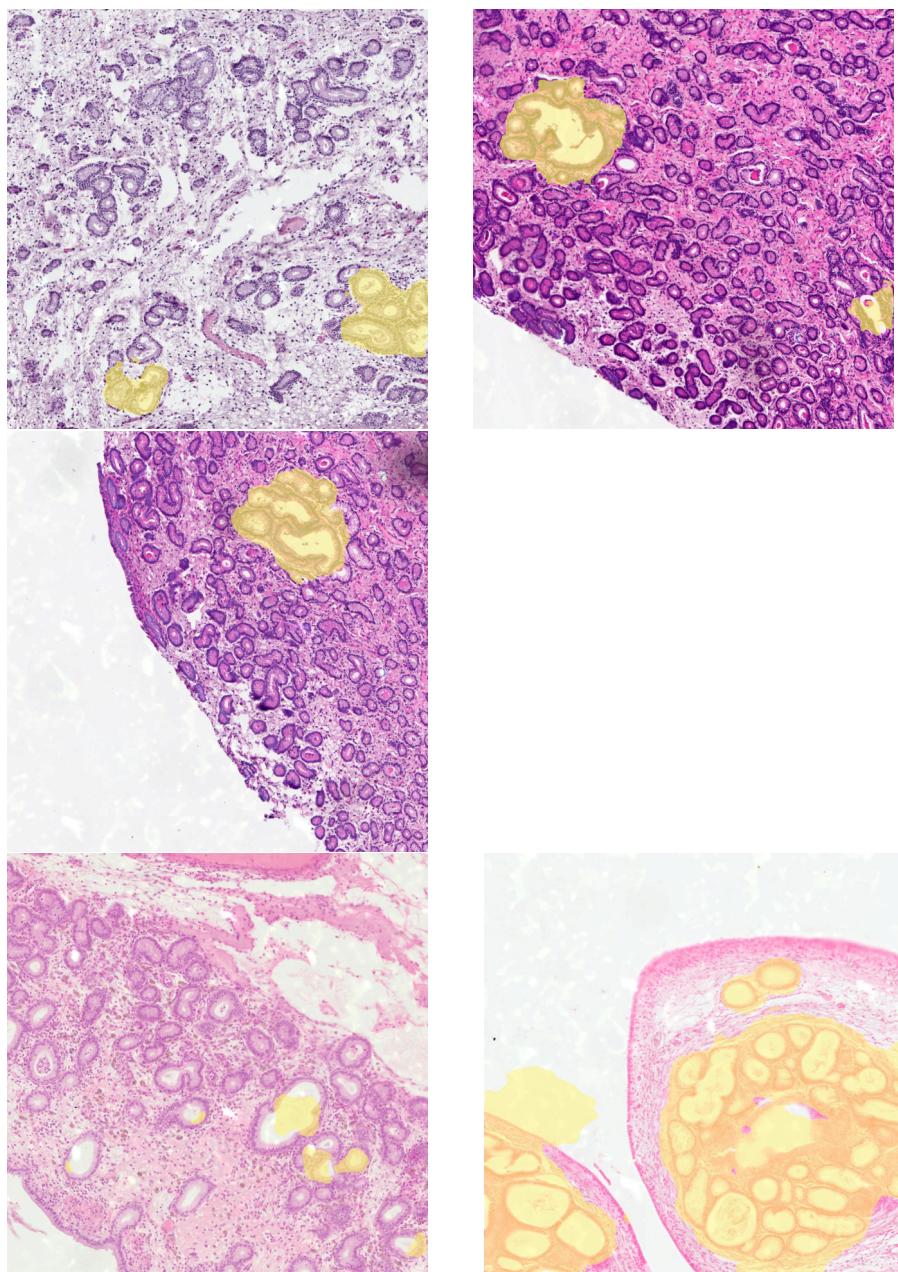


Figura A.5: Inferencia del mejor modelo segmentador de fibrosis sobre los datos de prueba.

A.3. Resultados de la inferencia sobre los datos de prueba de fibrosis

A.3.1. Mejor modelo

Apéndice B

Configuración del ambiente de entrenamiento en ClusterUY

QA consiste en una interfaz gráfica que permite al usuario pedir entrenamientos, y cuenta con un *backend* que los ejecuta. Los hiperparámetros, los archivos que conforman los conjuntos de datos, entre otros aspectos configurables que son relevantes para el entrenamiento, son mantenidos en una base de datos. El *backend* utiliza la base de datos para ejecutar los *scripts* de entrenamiento de QA llamados *train_ae.py*, que entrena el AutoEncoder (modelo base), y *train.py*, que entrena la U-Net (modelo de segmentación). Ambos son implementados en Python usando PyTorch.

Para entrenar la U-Net, fue necesario transferir el proyecto de QA (junto con los datos) a ClusterUY. Siguiendo la guía [59] se implementó un *script* que permite pedir recursos para ejecutar *train_ae.py* y *train.py* manualmente (sin el *framework* completo de QA).

Los recursos pedidos para correr los *script* fueron: 4 CPUs, 32GB de memoria RAM, GPU Nvidia P100 (12GB VRAM), y un total de 10hs de ejecución. El tiempo de ejecución fue estimado en base a entrenamientos anteriores, para ser confirmado luego de una corrida de prueba.

La versión de Python disponible en ClusterUY al momento de prueba era *Python 3.4*. Como QA requiere una versión superior, y no fue posible instalar paquetes mediante el manejador de paquetes de *CentOS 7* (debido a falta de permisos), se realizó una instalación local de *Python 3.9* descargando su código fuente y compilándolo en el *cluster*.

B.1. Pipelines de FastPathology

B.1.1. Pipeline para uso del modelo segmentador de núcleos

```
PipelineName "Nuclei segmentation"
```

```
PipelineDescription "Patch-wise segmentation of cell nuclei at 20x magnification"
```

```
PipelineInputData WSI "Whole-slide image"
```

Apéndice B. Configuración del ambiente de entrenamiento en ClusterUY

```
PipelineOutputData Segmentation stitcher 0
Attribute classes "Background; Cell nuclei"

### Processing chain

ProcessObject tissueSeg TissueSegmentation
Input 0 WSI

ProcessObject patch PatchGenerator
Attribute patch-size 256 256
Attribute patch-magnification 20
Attribute patch-overlap 0.1
Input 0 WSI
Input 1 tissueSeg 0

ProcessObject network SegmentationNetwork
Attribute scale-factor 0.003921568627451
Attribute model "$CURRENT_PATH$/.. models/high_res_nuclei_unet.onnx"
Input 0 patch 0

ProcessObject stitcher PatchStitcher
Input 0 network 0

### Renderers
Renderer imgRenderer ImagePyramidRenderer
Input 0 WSI

Renderer segRenderer SegmentationRenderer
Attribute border-opacity 0.5
Input 0 stitcher 0
```

Apéndice C

Cálculo de características con funciones implementadas por OpenCV

A partir de las máscaras de núcleos se obtiene el contorno de cada instancia con la función *findContours* de OpenCV. De acuerdo a la documentación de OpenCV ??, los parámetros de esta función son: la máscara, un modo de devolución del contorno, y un método de aproximación del contorno. La devolución del contorno adquiere relevancia cuando en la figura que se quiere aplicar la operación tiene varios contornos anidados, es decir, al menos un contorno se encuentra adentro de otro. Por este motivo es importante establecer como se quiere obtener la información que devuelve esta función, para poder interpretarla claramente. De los métodos que ofrece OpenCV, que se pueden ver en detalle en el artículo ??, el seleccionado en este caso se trata de *RETR_EXTERNAL*. Esta opción determina que solo se quiere obtener el contorno exterior. En cuanto al método de aproximación de contorno, es posible obtener una aproximación de las coordenadas de los píxeles que lo conforman, o por el contrario, no aproximar nada y obtener las coordenadas de todos los píxeles. En el mismo artículo también se describen las dos opciones posibles. En este caso se optó por el método *CHAIN_APPROX_SIMPLE* que comprime la información del contorno de forma que cuando se tiene una línea horizontal, vertical o diagonal, solo se indican las coordenadas de los píxeles de los extremos.

El resultado de utilizar la función *findContours* con los parámetros indicados anteriormente es una tupla. La primer componente de la tupla se trata del contorno de la instancia indicada por la máscara de segmentación. La segunda componente es la jerarquía del contorno devuelto. En este caso este último dato no tiene relevancia porque únicamente se devuelve el contorno exterior.

A partir de la información del contorno es que se calcula el perímetro y el área del núcleo, la implementación de OpenCV de las funciones utilizadas se puede encontrar en el repositorio ???. El perímetro se obtiene con la función de OpenCV *arcLength*. Esta función calcula el perímetro como la suma de la distancia entre cada píxel del contorno ($[x_i, y_i]$) y el inmediato anterior ($[x_{i-1}, y_{i-1}]$). Vale aclarar que la métrica de distancia utilizada se trata de la distancia euclídea. La siguiente ecuación representa la forma de cálculo:

Apéndice C. Cálculo de características con funciones implementadas por OpenCV

$$perímetro = \sum_{i=1}^{N-1} \sqrt{(x_i - x_{i-1})^2 + (y_i - y_{i-1})^2},$$

donde N representa el total del píxeles que conforman el contorno.

Por otro lado, para obtener el área del núcleo se aplica la función *contourArea*. Esta función obtiene el área aplicando el teorema de Green ???. Este teorema de cálculo vectorial relaciona la integral de línea de una curva cerrada \mathcal{C} con la integral doble sobre un región del plano determinada por la misma curva:

$$\oint_{\mathcal{C}} (Ldx + Mdy) = \iint_{\mathcal{D}} \left(\frac{\partial M}{\partial x} - \frac{\partial L}{\partial y} \right) dx dy.$$

Si tomamos \mathcal{C} como el contorno de la instancia detectada, y las funciones L y M como las funciones expresada a continuación:

$$M = \frac{1}{2}x \Rightarrow \frac{\partial M}{\partial x} = \frac{1}{2}$$

$$L = -\frac{1}{2}y \Rightarrow \frac{\partial L}{\partial y} = -\frac{1}{2},$$

$$\oint_{\mathcal{C}} -\frac{1}{2}y dx + \frac{1}{2}x dy = \iint_{\mathcal{D}} 1 dx dy.$$

La integral de la región \mathcal{D} se trata del área que queremos hallar, por lo que para obtener el resultado buscado debemos resolver la integral de línea en la curva cerrada \mathcal{C} . Dicha integral se puede expresar en el dominio discreto de la siguiente forma:

$$\begin{aligned} \sum_x \sum_y -\frac{1}{2}y_i(x_i - x_{i-1}) + \frac{1}{2}x_i(y_i - y_{i-1}) &= \\ \frac{1}{2} \sum_x \sum_y -y_i x_i + y_i x_{i-1} + x_i y_i - x_i y_{i-1} &= \\ \frac{1}{2} \sum_x \sum_y y_i x_{i-1} - x_i y_{i-1} \end{aligned} \tag{C.1}$$

La expresión C.1 es la que se utiliza en OpenCV al aplicar la función *contourArea*. A partir de las funciones implementadas por OpenCV, es necesario notar que el área y el perímetro de las instancias de núcleos detectadas no tienen valores enteros.

Referencias

- [1] Z Cortés-Vidauri, C Aréchiga-Flores, M Rincón-Delgado, F Rochín-Berumen, M López-Carlos, and G Flores-Flores. Revisión: El ciclo reproductivo de la yegua. abanico veterinario. 8 (3): 14-41. issn-e 2007-428x, issn 2448-6132, 2018.
- [2] TA Snider, C Sepoy, and GR Holyoak. Equine endometrial biopsy reviewed: observation, interpretation, and application of histopathologic data. *Theriogenology*, 75(9):1567–1581, 2011.
- [3] Sandra Schöniger and Heinz-Adolf Schoon. The healthy and diseased equine endometrium: A review of morphological features and molecular analyses. *Animals*, 10(4):625, 2020.
- [4] TA Snider, C Sepoy, and GR Holyoak. Equine endometrial biopsy reviewed: observation, interpretation, and application of histopathologic data. *Theriogenology*, 75(9):1567–1581, 2011.
- [5] RM Kenney and PA Doig. Equine endometrial biopsy. *Current therapy in theriogenology*, 2(3):723–729, 1986.
- [6] Halo ai. <https://indicalab.com/halo-ai/>. Accedido: 2023-11-04.
- [7] Rajarsi Gupta, Tahsin Kurc, Ashish Sharma, Jonas S Almeida, and Joel Saltz. The emergence of pathomics. *Current Pathobiology Reports*, 7:73–84, 2019.
- [8] Famke Aeffner, Mark D Zarella, Nathan Buchbinder, Marilyn M Bui, Matthew R Goodman, Douglas J Hartman, Giovanni M Lujan, Mariam A Molani, Anil V Parwani, Kate Lillard, et al. Introduction to digital image analysis in whole-slide imaging: a white paper from the digital pathology association. *Journal of pathology informatics*, 10(1):9, 2019.
- [9] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.
- [10] Adrian B Levine, Colin Schlosser, Jasleen Grewal, Robin Coope, Steve JM Jones, and Stephen Yip. Rise of the machines: advances in deep learning for cancer diagnosis. *Trends in cancer*, 5(3):157–169, 2019.
- [11] Arjun Punabhai Vekariya. *A deep learning based pipeline for metastatic breast cancer classification from whole slide images*. PhD thesis, Arlington, 2017.

Referencias

- [12] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.
- [13] What is the difference between semantic segmentation, object detection and instance segmentation? <https://datascience.stackexchange.com/questions/52015/what-is-the-difference-between-semantic-segmentation-object-detection-and-instance-segmentation>. Accedido: 2023-11-04.
- [14] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3431–3440, 2015.
- [15] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention—MICCAI 2015: 18th International Conference, Munich, Germany, October 5–9, 2015, Proceedings, Part III* 18, pages 234–241. Springer, 2015.
- [16] Vijay Badrinarayanan, Alex Kendall, and Roberto Cipolla. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE transactions on pattern analysis and machine intelligence*, 39(12):2481–2495, 2017.
- [17] Runtian Miao, Robert Toth, Yu Zhou, Anant Madabhushi, and Andrew Janowczyk. Quick annotator: an open-source digital pathology based rapid image annotation tool. *The Journal of Pathology: Clinical Research*, 7(6):542–547, 2021.
- [18] Henrik Sahlin Pettersen, Ilya Belevich, Elin Synnøve Røyset, Erik Smistad, Melanie Rae Simpson, Eija Jokitalo, Ingerid Reinertsen, Ingunn Bakke, and André Pedersen. Code-free development and deployment of deep segmentation models for digital pathology. 8, 2022. Accedido: 2023-11-04.
- [19] M.P. Humphries, P. Maxwell, and M. Salto-Tellez. Qupath: The global impact of an open source digital pathology system. *Computational and Structural Biotechnology Journal*, 19:852–859, 2021. Accedido: 2023-11-04.
- [20] André Pedersen, Marit Valla, Anna M Bofin, Javier Pérez De Frutos, Ingerid Reinertsen, and Erik Smistad. Fastpathology: an open-source platform for deep learning-based research and decision support in digital pathology. *Ieee Access*, 9:58216–58229, 2021.
- [21] Marc Macenko, Marc Niethammer, James S Marron, David Borland, John T Woosley, Xiaojun Guan, Charles Schmitt, and Nancy E Thomas. A method for normalizing histology slides for quantitative analysis. In *2009 IEEE international symposium on biomedical imaging: from nano to macro*, pages 1107–1110. IEEE, 2009.

Referencias

- [22] Johann Heinrich Lambert and August Beer. Ley de beer-lambert. https://es.wikipedia.org/wiki/Ley_de_Beer-Lambert. Accedido: 2023-11-04.
- [23] Arnout C Ruifrok, Dennis A Johnston, et al. Quantification of histochemical staining by color deconvolution. *Analytical and quantitative cytology and histology*, 23(4):291–299, 2001.
- [24] Marc Aubreville. Microscopy stain variations and how to estimate them. <https://towardsdatascience.com/microscopy-stain-variations-and-how-to-estimate-them-baca29f57ee9>. Accedido: 2023-11-04.
- [25] Erik Reinhard, Michael Adhikhmin, Bruce Gooch, and Peter Shirley. Color transfer between images. *IEEE Computer graphics and applications*, 21(5):34–41, 2001.
- [26] Daniel L Ruderman, Thomas W Cronin, and Chuan-Chin Chiao. Statistics of cone responses to natural images: implications for visual coding. *JOSA A*, 15(8):2036–2045, 1998.
- [27] Abhishek Vahadane, Tingying Peng, Amit Sethi, Shadi Albarqouni, Lichao Wang, Maximilian Baust, Katja Steiger, Anna Melissa Schlitter, Irene Esposto, and Nassir Navab. Structure-preserving color normalization and sparse stain separation for histological images. *IEEE transactions on medical imaging*, 35(8):1962–1971, 2016.
- [28] Daniel D Lee and H Sebastian Seung. Learning the parts of objects by non-negative matrix factorization. *Nature*, 401(6755):788–791, 1999.
- [29] Sophia J Wagner, Nadieh Khalili, Raghav Sharma, Melanie Boxberg, Carsten Marr, Walter de Back, and Tingying Peng. Structure-preserving multi-domain stain color augmentation using style-transfer with disentangled representations. In *Medical Image Computing and Computer Assisted Intervention-MICCAI 2021: 24th International Conference, Strasbourg, France, September 27–October 1, 2021, Proceedings, Part VIII 24*, pages 257–266. Springer, 2021.
- [30] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *Advances in neural information processing systems*, 27, 2014.
- [31] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE international conference on computer vision*, pages 2223–2232, 2017.
- [32] Hsin-Ying Lee, Hung-Yu Tseng, Qi Mao, Jia-Bin Huang, Yu-Ding Lu, Manneesh Singh, and Ming-Hsuan Yang. Drit++: Diverse image-to-image translation via disentangled representations. *International Journal of Computer Vision*, 128:2402–2417, 2020.

Referencias

- [33] Mehdi Mirza and Simon Osindero. Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784*, 2014.
- [34] Li Deng. The mnist database of handwritten digit images for machine learning research. *IEEE Signal Processing Magazine*, 29(6):141–142, 2012.
- [35] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1125–1134, 2017.
- [36] Carlo Alberto Barbano and André Pedersen. Eidoslab/torchstain: v1.2.0-stable. <https://doi.org/10.5281/zenodo.6979540>, August 2022.
- [37] André Pedersen. Fast-stain-normalization. <https://github.com/andreped/fast-stain-normalization>.
- [38] Sophia J. Wagner. Histaugan. <https://github.com/sophiajw/HistAuGAN>.
- [39] Camelyon17 - grand challenge. <https://camelyon17.graniibd-challenge.org/>.
- [40] Henrik Sahlin Pettersen, Ilya Belevich, Elin Synnøve Røyset, Erik Smistad, Melanie Rae Simpson, Eija Jokitalo, Ingerid Reinertsen, Ingunn Bakke, and André Pedersen. Code-free development and deployment of deep segmentation models for digital pathology. *Frontiers in Medicine*, 8:816281, 2022.
- [41] Korsuk Sirinukunwattana, Josien PW Pluim, Hao Chen, Xiaojuan Qi, Pheng-Ann Heng, Yun Bo Guo, Li Yang Wang, Bogdan J Matuszewski, Elia Bruni, Urko Sanchez, et al. Gland segmentation in colon histology images: The glas challenge contest. *Medical image analysis*, 35:489–502, 2017.
- [42] Wai Po Kevin Teng. Gland-segmentation. <https://github.com/twpkevin06222/Gland-Segmentation>, 2022.
- [43] Jacqueline H Carr and Bernadette F Rodak. *Atlas de hematología clínica*. Ed. Médica Panamericana, 2009.
- [44] Nikita Shvetsov, Morten Grønnesby, Edvard Pedersen, Kajsa Møllersen, Lill-Tove Rasmussen Busund, Ruth Schwienbacher, Lars Ailo Bongo, and Thomas Karsten Kilvaer. A pragmatic machine learning approach to quantify tumor-infiltrating lymphocytes in whole slide images. *Cancers*, 14(12):2974, 2022.
- [45] Simon Graham, Quoc Dang Vu, Shan E Ahmed Raza, Ayesha Azam, Yee Wah Tsang, Jin Tae Kwak, and Nasir Rajpoot. Hover-net: Simultaneous segmentation and classification of nuclei in multi-tissue histology images. *Medical image analysis*, 58:101563, 2019.

Referencias

- [46] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Identity mappings in deep residual networks. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part IV 14*, pages 630–645. Springer, 2016.
- [47] Jevgenij Gamper, Navid Alemi Koohbanani, Ksenija Benet, Ali Khuram, and Nasir Rajpoot. Pannuke: an open pan-cancer histology dataset for nuclei instance segmentation and classification. In *Digital Pathology: 15th European Congress, ECDP 2019, Warwick, UK, April 10–13, 2019, Proceedings 15*, pages 11–19. Springer, 2019.
- [48] Sagar Vaze, Weidi Xie, and Ana IL Namburete. Low-memory cnns enabling real-time ultrasound segmentation towards mobile deployment. *IEEE Journal of Biomedical and Health Informatics*, 24(4):1059–1069, 2020.
- [49] Kien Rea. How to select the correct magnification and patch size for digital pathology projects. <http://www.andrewjanowczyk.com/how-to-select-the-correct-magnification-and-patch-size-for-digital-pathology>. 2022. Accedido: 2023-11-04.
- [50] Iturriaga S. Nesmachnow S. Cluster-uy: Collaborative scientific high performance computing in uruguay. *Communications in Computer and Information Science*, 1151, 2019.
- [51] Le Hou, Rajarsi Gupta, John S Van Arnam, Yuwei Zhang, Kaustubh Sivalanka, Dimitris Samaras, Tahsin M Kurc, and Joel H Saltz. Dataset of segmented nuclei in hematoxylin and eosin stained histopathology images of ten cancer types. *Scientific data*, 7(1):185, 2020.
- [52] Documentación de pytorch - conv2d. <https://pytorch.org/docs/1.8.1/generated/torch.nn.Conv2d.html?highlight=conv2d#torch.nn.Conv2d>. Accedido: 2023-11-04.
- [53] Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pages 249–256. JMLR Workshop and Conference Proceedings, 2010.
- [54] Mina Amiri, Rupert Brooks, and Hassan Rivaz. Fine-tuning u-net for ultrasound image segmentation: different layers, different outcomes. *IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control*, 67(12):2510–2518, 2020.
- [55] Jane Westendorf, Bruce Wobeser, and Tasha Epp. IIB or not IIB, part 1: retrospective evaluation of kennedy–doig categorization of equine endometrial biopsies at a veterinary diagnostic laboratory and comparison with published reports. *Journal of Veterinary Diagnostic Investigation*, 34(2):206–214, 2022.

Referencias

- [56] Erik Smistad, Mohammadmehdi Bozorgi, and Frank Lindseth. Fast: framework for heterogeneous medical image computing and visualization. *International Journal of computer assisted radiology and surgery*, 10:1811–1822, 2015.
- [57] Erik Smistad, Andreas Østvik, and André Pedersen. High performance neural network inference, streaming, and visualization of medical images using fast. *Ieee Access*, 7:136310–136321, 2019.
- [58] Augustus Odena, Vincent Dumoulin, and Chris Olah. Deconvolution and checkerboard artifacts. *Distill*, 2016.
- [59] Especificación de un trabajo - clusteruy. https://cluster.uy/ayuda/como_ejecutar/.

Índice de tablas

1.1. Escala de Kenney-Doig. Adaptada de [4].	8
3.1. Métricas de iteraciones del modelo de segmentación de fibrosis con el <i>framework</i> de QA al momento de generar el <i>dataset</i>	53
5.1. Resultados de los entrenamientos del modelo segmentador de fibrosis de QA sobre los datos de prueba.	85
5.2. Resultados de los entrenamientos del modelo segmentador de fibrosis de Pytorch-UNet sobre los datos de prueba.	85

Esta página ha sido intencionalmente dejada en blanco.

Índice de figuras

1.1.	Imágenes que ilustran el proceso de obtención de una biopsia uterina de yegua.	3
1.2.	Muestra de endometrio de yegua teñida con H&E [3]. Se señalan las distintas estructuras de interés.	3
1.3.	Diferencias de la disposición glandular entre cada período del ciclo reproductivo. Adaptada de [3].	4
1.4.	Muestras de epitelio con endometrosis. Adaptado de [3].	6
1.5.	Muestras de epitelio con endometritis [3].	7
1.6.	Laguna linfática. Adaptada de [4].	8
1.7.	A la izquierda imagen original. En el medio segmentación semántica, se reconocer la clase silla y dónde están ubicadas. A la derecha segmentación de instancias, además de reconocer la clase silla, se diferencia cada instancia con un color distinto [13].	11
1.8.	Arquitectura de la U-Net. Los 4 bloques de <i>downsampling</i> forman el camino contractivo. Le sigue el llamado espacio latente. Los 4 bloques de <i>upsampling</i> forman el camino expansivo. Adaptado de [15].	13
1.9.	Parche de biopsia endometrial de yegua segmentada en superpíxeles generados por QuickAnnotator. Se agrupan los píxeles en regiones de color similar.	14
1.10.	Parche de biopsia endometrial de yegua con glándula segmentada usando QuickAnnotator. El color celeste representa la clase positiva, mientras que la etiqueta rosada representa la clase negativa. Los superpíxeles ayudan al etiquetado ya que cada región se pinta del mismo color.	15
1.11.	Flujo de trabajo propuesto en el proyecto NoCodeSeg [18]. Se comienza por abrir una WSI en QuPath y anotar estructuras de interés (imagen superior izquierda). La imagen y su máscara son recortados en parches, y con ellos se entrena un modelo segmentador, a través de DeepMIB. Luego se hace inferencia sobre nuevos parches de la WSI con el modelo entrenado, y se exportan a QuPath, donde se modifican a gusto del usuario, para volver a repetir este proceso. . .	16
2.1.	Ejemplos de imágenes de igual resolución a distinto nivel de magnificación.	20

Índice de figuras

2.2. Comparación de las WSIs iniciales y la nueva partida. Las imágenes 2.2a y 2.2b corresponden a la yegua 560, mientras que las imágenes 2.2c y 2.2d corresponden a la yegua 148.	21
2.3. WSI del <i>dataset</i> - archivo <i>sandram6.svs</i> . La flecha verde señala glándulas dilatadas y dispuestas en forma de nido. La flecha celeste marca las fibras de colágeno que envuelven al nido de glándulas, generando el efecto de la patología llamada fibrosis. El tamaño puede compararse con las estructuras de las figuras 2.2c y 2.2d, cuya escala también es de $100\mu\text{m}$	22
2.4. Recortes de WSIs que muestran los tres dominios presentes en el conjunto de imágenes.	23
2.5. A la izquierda se visualiza una gráfica de una imagen del dominio violeta en el espacio RGB y a la derecha en el espacio OD. Se observa que en el espacio OD los píxeles violetas son linealmente separables de los rosados. Imagen adaptada de [21].	25
2.6. En la gráfica de la izquierda en el espacio OD se señalan los 2 vectores propios principales. En la gráfica 2D de la derecha se muestra la proyección de todos los píxeles sobre el plano formado por los 2 vectores propios principales. También se señalan en la imagen los vectores de tinción. Figura adaptada de [24].	26
2.7. Imagen obtenida del informe de Cycle-GAN [31] que muestra los componentes del modelo y explica gráficamente la función de pérdida de consistencia de ciclo. La entrada $x \in X$ se convierte al dominio Y a través de G , para luego ser devuelta a X por la función F , obteniendo \hat{x} . La función de consistencia de ciclo es calculada entre x y \hat{x} . El proceso es análogo para la entrada $y \in Y$	30

Índice de figuras

2.8. Imagen obtenida de [32]. En la sección (a) se expone la metodología propuesta para el entrenamiento. Cada dominio (X, Y) cuenta con un codificador de contenido (E_X^c, E_Y^c) y otro de atributos (E_X^a, E_Y^a). A cada entrada (x, y) se le calcula los vectores de contenido y atributos utilizando los codificadores de su dominio. Luego, el generador de cada dominio (G_X, G_Y) toma como entrada los atributos de su dominio y el contenido del otro, generando imágenes intermedias. En este paso se calculan las funciones de costo de los adversarios de cada dominio (L_{adv}^{domain}). Luego se calculan los vectores de cada imagen generada y se repite el ciclo anterior, donde G_X acepta el vector proveniente de E_X^a , y el vector E_Y^c . Lo análogo se hace para G_Y . Cada generador genera una imagen (\hat{x}, \hat{y}) que, de ser perfecto, sería idéntica a la original. Las secciones (b) y (c) representan distintas formas de utilizar el modelo para hacer inferencia. En (b) se genera una imagen del dominio Y a partir de un vector de contenidos del dominio X , y un vector de atributos aleatorio. Esto da como salida una imagen del dominio Y con el contenido de la entrada. En (c) se genera una imagen del dominio Y a partir del vector de contenido de una imagen del dominio X y el vector de atributos de una imagen del dominio Y .	32
2.9. Figura obtenida del informe de HistAuGAN [29]. En la sección (a) se muestran ejemplos de imágenes de histopatología con tinción H&E. En la sección (b) se presenta el flujo de trabajo para entrenar el modelo, y en la sección (c) para hacer inferencia.	33
2.10. Imágenes obtenidas a partir de WSIs recopiladas para el proyecto. Estas imágenes fueron utilizadas para probar los algoritmos de Macenko et al. y Reinhard et al.	34
2.11. Imágenes obtenidas con el algoritmo de Macenko et al. En las figuras (a) y (b) se muestran cada tinción por separado, y la figura (c) es una combinación de las anteriores.	35
2.12. Resultado obtenido con el algoritmo de Reinhard et al. Imagen llevada del dominio original (rosado) al dominio de referencia (violeta).	36
2.13. Las figuras (a) y (b) se tratan de imágenes obtenidas a partir de WSIs recopiladas para el proyecto. La figura (c) es resultado alcanzado con el algoritmo de Vahadane et al.	37
2.14. Inferencia usando HistAuGAN sobre imágenes de la base de datos.	38
2.15. Imágenes utilizadas para probar los algoritmos de normalización de color y resultados de cada uno. HistAuGAN obtiene los mejores resultados.	39
3.1. Imágenes de entrenamiento del modelo de epitelio de colon: A la izquierda, la imagen original recortada. En el centro la predicción del modelo reportado y a la derecha las anotaciones. La flecha roja muestra los límites aproximados para el tamaño mínimo del orificio del túbulo (lleno o no lleno) utilizado durante las anotaciones.	42

Índice de figuras

3.2. Filtro de relleno de huecos: En la imagen de la izquierda se tiene la inferencia cruda del modelo. La flecha roja señala la discontinuidad en el borde de la glándula. En la imagen del centro se tiene la máscara luego del <i>closing</i> con los bordes de las glándulas unidos. Mientras que en la imagen de la derecha se tienen las máscaras luego de aplicar el algoritmo de <i>Flood Fill</i>	43
3.3. Filtro de eliminación de epitelio luminal: al comparar las máscaras binarias se coloca el tejido en el canal rojo y el epitelio en el canal verde. Para la visualización se observa que los píxeles donde coinciden ambas máscaras poseen color amarillo (coinciden los canales rojo y verde), mientras que las instancias que no coinciden (debido a la erosión) tienen color verde. Se desea eliminar las instancias de la máscara de epitelio que contienen píxeles de color verde.	44
3.4. Resultados no satisfactorios del modelo de segmentación de epitelio de colon.	45
3.5. Inferencias con el modelo de segmentación de glándulas. A pesar de que las imágenes de entrenamiento de este modelo provienen de tejidos distintos al caso de estudio, se observa la similitud entre ambos.	46
3.6. Células del sistema inmune en muestras con tinción H&E. Adaptado de [43].	46
3.7. Inferencia del modelo de Shvetsov et al. [44] sobre imagen del proyecto. Los n úcleos pintados de amarillo se clasifican como células muertas, los azules como células inflamatorias, los verdes como neoplásicos, y los turquesa como epitelial no neoplásicos. Hay una categoría adicional que no aparece en la imagen que son los n úcleos del tejido conectivo.	48
3.8. Inferencia del modelo de segmentación de n úcleos con arquitectura Lightweight U-Net y entrenado con la base de datos de PanNuke, sobre imagen del proyecto.	49
3.9. Imágenes del proceso de posprocesamiento realizado sobre las máscaras de n úcleos. Se eliminan las instancias de n úcleos que coinciden con las de gl ándulas. Por otro lado se dilatan las gl ándulas y se eliminan las instancias con área mayor a cierto umbral, asumiendo que corresponden a gl ándulas no detectadas.	50
3.10. Ejemplos de parches de igual tama ño a distintos niveles de magnificación.	51
3.11. Flujo de trabajo para el etiquetado de Fibrosis. A la izquierda se tiene la WSI que es recortada en imágenes de 1000×1000 . En el centro se observa una imagen que tiene tres ROIs etiquetadas, y a la derecha se tiene la ROI seleccionada de tamaño 512×512	51

Índice de figuras

<p>3.12. Ejemplos de las aumentaciones de datos de QuickAnnotator. En la fila de arriba se observan las imágenes originales. En la de abajo, las transformadas. Se puede ver como la técnica de manejo de bordes, el <i>blur</i>, y las modificaciones sobre el color de las imágenes hacen que se pierda detalle de las estructuras, valioso a la hora de entrenar.</p> <p>3.13. Métricas de los entrenamientos correspondientes al proceso de <i>active learning</i> realizados con QuickAnnotator. A la izquierda se gráfica la entropía cruzada sobre el conjunto de entrenamiento. A la derecha se presenta el <i>accuracy</i> sobre el mismo conjunto.</p> <p>3.14. Comparación entre tamaños de imágenes de entrenamiento de la red de QuickAnnotator.</p> <p>3.15. Experimento de tamaño de imagen: Error cuadrático medio entre las imágenes de entrenamiento del <i>autoencoder</i>, promediado por época.</p> <p>3.16. Experimento de tamaño de imagen: Entropía cruzada ponderada (<i>loss</i>) y relación de verdaderos positivos (TPR), en función del numero de épocas. La gran variación del TPR entre épocas, junto con la divergencia de la <i>loss</i> en validación, que se ven en la figura 3.16b, revelan un problema con el entrenamiento.</p> <p>3.17. Experimento de función de costo: Función de costo de Dice (<i>loss</i>) y TPR sobre conjuntos de entrenamiento (train) y validación (val), en función de las épocas.</p> <p>3.18. Experimento de aumentado de datos: Entropía Cruzada y TPR sobre conjuntos de entrenamiento y validación, en función de las épocas. Las métricas sobre el conjunto de entrenamiento (train) muestran un comportamiento normal. El TPR sobre el conjunto de validación (val) muestra gran variabilidad a lo largo de las épocas, y la <i>loss</i> no muestra una disminución respecto a su valor original.</p> <p>3.19. Ejemplos de parches de BLCA-DB (figuras (a) y (b)) junto a sus máscaras (figuras (c) y (d)), originarios del trabajo [51]. Cada parche es de tamaño 1000×1000 píxeles. Los píxeles de color blanco de las máscaras pertenecen a clase positiva (núcleo canceroso).</p> <p>3.20. Métricas de entrenamiento de PyTorch-UNet sobre BLCA-DB, en función del <i>batch</i> (5 épocas en total).</p> <p>3.21. Métricas de entrenamiento de PyTorch-UNet sobre base de datos de fibrosis, en función del <i>batch</i> (10 épocas en total).</p> <p>3.22. Métricas de entrenamiento de PyTorch-UNet sobre base de datos de fibrosis, en función del <i>batch</i> (100 épocas en total).</p> <p>3.23. Métricas del entrenamiento de QA sobre base de datos de fibrosis en función de las épocas, sin datos completamente negativos.</p> <p>4.1. Flujo de trabajo propuesto, dividido en cuatro módulos principales: inferencia de modelos, alineación y extracción de parches, refinamiento de segmentación y extracción de datos.</p>	<p>53</p> <p>54</p> <p>55</p> <p>55</p> <p>56</p> <p>57</p> <p>58</p> <p>60</p> <p>62</p> <p>63</p> <p>64</p> <p>65</p> <p>68</p>
-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	-----------------------------------------------------------------------------------------------------------------------------------

Índice de figuras

4.2. Etapa de extracción y alineación de máscaras. Inferencias de glándulas y núcleos previos al refinamiento. Inferencias del modelo de fibrosis. Se observa que la magnificación de $40\times$ es adecuada para observar las primeras dos clases, no así la tercera.	71
4.3. Mapas de calor de la densidad de glándulas, núcleos y fibrosis, de la WSI <i>Lady.svs</i> , generados utilizando los modelos de segmentación. La escala de los mapas de glándulas y fibrosis está normalizada, y la densidad en esos casos se mide como la cantidad de píxeles segmentados sobre el total de píxeles, por parche. En el caso del mapa de densidad de núcleos, la densidad se calcula como la cantidad de núcleos por parche.	72
4.4. Comparación de inferencias para ambos modelos de detección de glándulas teniendo en cuenta el área relativa de las glándulas para la WSI de <i>Lady.svs</i>	74
4.5. Parche recortado de la WSI <i>Lady.svs</i> . Se señalan con flechas verdes los linfocitos presentes en la imagen. Pude observarse que estos núcleos tienden a ser más pequeños, circulares y oscuros que el resto. 75	75
5.1. Resultados con imágenes de prueba del modelo entrenado de HistAuGAN. La columna de la izquierda contiene las imágenes originales, a la derecha de cada imagen se encuentran las imágenes generadas por el modelo en los otros dominios.	79
5.2. Patrón cuadriculado de 4×4 píxeles obtenido en imágenes generadas con el modelo de HistAuGAN.	82
5.3. Transformaciones de datos conservadas de QA. En la fila de arriba se ven las imágenes originales. En la fila de abajo, las transformaciones aplicadas una a la vez. De izquierda a derecha: contraste, brillo, <i>blur</i> , rotación, <i>gamma</i> , HSV.	83
5.4. Ejemplo de imagen de F-DB transformada con las aumentaciones de datos conservadas de QA.	83
5.5. Imagen del conjunto de prueba. Inferencia del mejor modelo segmentador de fibrosis.	86
5.6. Comparación de histogramas en escala logarítmica para una misma WSI con ambos modelos de glándulas, el modelo de epitelio (en rojo) y el modelo de glándulas de GlaS (en azul).	88
5.7. Inferencias realizadas con el modelo de epitelio que representan distintas instancias de glándulas detectadas, con valores en rangos de áreas indicados en el título de cada imagen.	89
5.8. A la izquierda se observa el mapa de calor de toda la WSI para la segmentación de glándulas. A la derecha, glándulas con áreas mayores al 80 % y se señala su ubicación dentro del mapa.	89
5.9. Comparación de una misma yegua en dos muestras distintas.	90

Índice de figuras

5.10. Inferencias con el modelo de epitelio para muestras del dominio rosado. En la figura a) se ven imágenes del dominio rosado con baja densidad glandular, mientras que la figura b) muestra glándulas dilatadas. El modelo muestra dificultad para detectar glándulas de gran área.	91
5.11. Inferencia con el modelo de epitelio para la WSI de 560bd5nueva.svs del dominio rojo, e histograma de densidad de glándulas.	92
5.12. Histograma de las características halladas para la clasificación de núcleos, normalizadas en el rango [0, 1].	96
5.13. En la imagen de la izquierda, detección de núcleos con valor de circularidad mayor al percentil 90. En la derecha, parche original con los núcleos detectados señalados con flechas verdes. En la misma imagen se marcó con un círculo rojo otro núcleo redondo no detectado con este método.	96
5.14. Distribución de las características de núcleos tomadas de a 2. Se puede notar que las gráficas ubicadas en la diagonal representan la distribución de cada característica. No se observa para ninguna combinación de dos características una separación entre clases de núcleos.	97
5.15. La figura de la izquierda muestra una imagen del conjunto de prueba con los linfocitos señalados. A la derecha, los resultados obtenidos con el modelo <i>k</i> -means entrenado con 5 clases. Las clases predichas se pintan cada una con un color distinto. Se observa que los linfocitos señalados quedan comprendidos en las clases rosada y roja.	98
A.1. Función de costo en entrenamiento y validación. La métrica utilizada se trata de la entropía cruzada.	104
A.2. Métrica de <i>accuracy</i> para entrenamiento y validación.	104
A.3. Tasa de verdaderos negativos para entrenamiento y validación. . . .	105
A.4. Tasa de verdaderos positivos para entrenamiento y validación. . . .	105
A.5. Inferencia del mejor modelo segmentador de fibrosis sobre los datos de prueba.	106

Esta es la última página.
Compilado el viernes 10 noviembre, 2023.
<http://iie.fing.edu.uy/>