



UNIVERSIDAD DE CONCEPCIÓN
FACULTAD DE INGENIERÍA
DEPARTAMENTO DE INGENIERÍA ELÉCTRICA



**GENERACIÓN DE BASE DE DATOS Y ALGORITMO DE SEGMENTACIÓN
AUTOMÁTICA DE NÚCLEOS HEPÁTICOS EN IMÁGENES 3D DE
MICROSCOPIA MEDIANTE REDES NEURONALES**

POR

Jorge Andrés Vergara Estrada

Memoria de Título presentada a la Facultad de Ingeniería de la Universidad de Concepción
para optar al grado académico de Ingeniero Civil Biomédico

Profesor(es) Guía:

Dr. Pamela Beatriz Guevara Alvez

Dr. Fabián Josue Segovia Miranda

Dr. Hernán Morales Navarrete

Comisión

Dr. Sebastián Eugenio Godoy Medel

Agosto 2024

Concepción

(Chile)

© 2024 Jorge Andrés Vergara Estrada

© 2024 Jorge Andrés Vergara Estrada

Se autoriza la reproducción total o parcial, con fines académicos, por cualquier medio o procedimiento, incluyendo la cita bibliográfica del documento.

Agradecimientos

Son muchas las personas a las que debo agradecer por haber sido parte de este proceso, un camino lleno de estudio, esfuerzo y perseverancia. Aunque hubo muchos altos y bajos, siempre seguimos adelante, y ahora, al llegar a esta meta, quiero expresar mi más sincera gratitud.

En primer lugar, quiero agradecer a mi mamá y a mi papá, quienes han sido un apoyo fundamental para que yo esté aquí. No tengo palabras para expresar toda mi gratitud hacia ellos. Fueron las personas que, en los momentos más difíciles, siempre me ayudaron a tomar las mejores decisiones. Aunque a veces tomamos algunas malas, siempre seguimos adelante gracias a su inquebrantable apoyo. Muchas gracias a toda mi familia por estar siempre a mi lado.

También quiero agradecer al Profesor Fabián Segovia, la Profesora Pamela Guevara y el Profesor Hernán Morales por darme la oportunidad de trabajar en un equipo multidisciplinario. Gracias por el conocimiento brindado y por proporcionarme las herramientas necesarias para llevar a cabo este trabajo.

Un agradecimiento especial a mis amigos y estudiantes de doctorado, Dilan y Cristian, a mi amigo y a mi compañero Jean, por acompañarme y ayudarme en todo este proceso en el laboratorio. Su apoyo ha sido invaluable para terminar esto.

A Luis, por alegrarme y acompañarme incluso en los días más ocupados. Gracias por tu cariño y amor incondicional, por siempre estar ahí, a pesar del poco tiempo que tenías.

Por último, a mi mejor amigo Sebastián y a mi grupo de amigos, con quienes compartí muchos momentos que me ayudaron a despejar la mente en tiempos difíciles. En los momentos abrumadores, siempre estuvieron ahí para escucharme y disfrutar de buenos momentos juntos. Además, a mis compañeros de carrera y a mi gran amiga Beatriz, gracias por el apoyo mutuo en los momentos más difíciles de la carrera.

Sé que estas palabras son pocas y probablemente me falten muchas personas por mencionar, pero quiero agradecer todas las personas que formaron parte de este proceso, que me ayudaron no solo a formarme como profesional, sino también como persona.

Resumen

La comprensión de la organización de los órganos y la coordinación celular para formar tejidos funcionales es fundamental en biología celular y del desarrollo. La histología de los tejidos es crucial para el diagnóstico de diversas patologías, incluyendo el cáncer. Sin embargo, las evaluaciones histopatológicas tradicionales en dos dimensiones (2D) resultan en una pérdida de contexto tridimensional (3D) del tejido. El hígado, un órgano vital con múltiples funciones, depende de una compleja arquitectura 3D para su funcionamiento adecuado. Por ello, es esencial conocer la información espacial exacta de los núcleos y celular para comprender sus funciones e interacciones.

Para superar estas limitaciones, se utilizan modelos geométricos y softwares como Motion Tracking (MT), procesan las imágenes de hígado obtenidas mediante microscopía confocal con fluorescencia, generando una reconstrucción 3D de alta resolución de todos los componentes celulares y subcelulares importantes en el tejido. Un paso clave es la segmentación de núcleos celulares, la cual enfrenta desafíos en núcleos muy cercanos.

Las redes neuronales convolucionales (CNNs) son especialmente efectivas para el procesamiento y análisis de datos visuales. La Residual Attention U-Net 3D es una variante avanzada de CNN diseñada para la segmentación de imágenes volumétricas, combinando la arquitectura U-Net con bloques residuales y mecanismos de atención para mejorar la precisión de la segmentación. En esta memoria, se entrenó un modelo basado en la red mencionada, para lo cual se generó una base de datos de núcleos 3D de alta calidad en imágenes de microscopía de fluorescencia del lobulillo hepático. Además, para entrenar esta red neuronal se decidió un enfoque multitarea, realizando segmentación binaria de los núcleos hepáticos y predicción de un campo vectorial para estimar los centroides de cada núcleos y generar la segmentación por instancia [1, 2, 3, 4, 5].

Para la creación de la base de datos, se segmentaron aproximadamente 2900 núcleos 3D en tres sujetos de control (C1, C2 y C3) en las regiones periportal, central y pericentral del lobulillo hepático. Los sujetos C1 y C2 se utilizaron para el entrenamiento, mientras que el sujeto C3 se empleó para la evaluación. Además del método de segmentación de MT, se utilizó Cellpose [3] para comparar los resultados. Nuestro modelo propuesto mostró una precisión media superior al 17 %, 28 % y 33 % en comparación con MT en las regiones central, periportal y pericentral, respectivamente. También superó a Cellpose en un 5 %, 17 % y 7 % en las mismas regiones, respectivamente. En particular, en la región periportal, nuestro modelo logró resultados significativamente mejores, demostrando su capacidad para manejar la segmentación en regiones complejas del tejido hepático.

Abstract

The understanding of organ organization and cellular coordination to form functional tissues is fundamental in cell and developmental biology. Tissue histology is crucial for the diagnosis of various pathologies, including cancer. However, traditional two-dimensional (2D) histopathological evaluations result in a loss of three-dimensional (3D) context of the tissue. The liver, a vital organ with multiple functions, relies on a complex 3D architecture for its proper functioning. Therefore, it is essential to know the exact spatial information of the nuclei and cells to understand their functions and interactions.

To overcome these limitations, geometric models and software such as Motion Tracking (MT) are used to process liver images obtained through confocal fluorescence microscopy, generating a high-resolution 3D reconstruction of all important cellular and subcellular components in the tissue. A key step is the segmentation of cell nuclei, which faces challenges in very close nuclei.

Convolutional neural networks (CNNs) are particularly effective for processing and analyzing visual data. The Residual Attention U-Net 3D is an advanced variant of CNN designed for the segmentation of volumetric images, combining the U-Net architecture with residual blocks and attention mechanisms to improve segmentation accuracy. In this study, a model based on the mentioned network was trained, for which a high-quality 3D nuclei database in fluorescence microscopy images of the hepatic lobule was generated. Additionally, a multitask approach was chosen to train this neural network, performing binary segmentation of hepatic nuclei and predicting a vector field to estimate the centroids of each nucleus and generate instance segmentation [1, 2, 3, 4, 5].

For the database creation, approximately 2900 3D nuclei were segmented in three control subjects (C1, C2, and C3) in the periportal, central, and pericentral regions of the hepatic lobule. Subjects C1 and C2 were used for training, while subject C3 was used for evaluation. In addition to the MT segmentation method, Cellpose [3] was used to compare the results. Our proposed model showed an average precision improvement of 17 %, 28 %, and 33 % compared to MT in the central, periportal, and pericentral regions, respectively. It also outperformed Cellpose by 5 %, 17 %, and 7 % in the same regions, respectively. Notably, in the periportal region, our model achieved significantly better results, demonstrating its ability to handle segmentation in complex regions of hepatic tissue.

Tabla de contenidos

CAPITULO 1. INTRODUCCIÓN.....	13
1.1 INTRODUCCIÓN GENERAL	13
1.2 OBJETIVOS	14
1.2.1 <i>Objetivo General</i>	14
1.2.2 <i>Objetivos Específicos</i>	14
1.3 ALCANCES Y LIMITACIONES	15
1.4 METODOLOGÍA.....	15
1.4.1 <i>Generación de la Base de Datos</i>	15
1.4.2 <i>Estudio Bibliográfico</i>	16
1.4.3 <i>Desarrollo del Modelo de Segmentación</i>	16
1.4.4 <i>Entrenamiento del Modelo</i>	16
1.4.5 <i>Predicción y Reconstrucción de Imágenes</i>	16
1.4.6 <i>Evaluación de Resultados</i>	17
1.5 TEMARIO	17
1.6 DISCUSIÓN	18
CAPITULO 2. MARCO TEÓRICO	19
2.1 INTRODUCCIÓN	19
2.2 FUNCIÓN Y ESTRUCTURA DEL HÍGADO.....	19
2.2.1 <i>Arquitectura del lóbulillo hepático</i>	19
2.2.2 <i>Tipos celulares hepáticos</i>	20
2.2.2.1 <i>Hepatocitos</i>	20
2.2.2.2 <i>Células no parenquimatosas</i>	21
2.2.3 <i>Regiones del lobulillo hepático</i>	22
2.3 FLUORESCENCIA Y MICROSCOPIA	23
2.3.1 <i>Fenómeno de Fluorescencia</i>	23
2.3.2 <i>Microscopía de Fluorescencia</i>	23
2.3.3 <i>Microscopía de Fluorescencia Confocal</i>	24
2.3.4 <i>Marcadores Fluorescentes</i>	25

2.3.5	<i>Formatos de imagen utilizados</i>	26
2.3.5.1	Archivos “.tiff”	26
2.4	SEGMENTACIÓN DE NÚCLEOS A TRAVÉS DE MOTION TRACKING	26
2.5	DEEP LEARNING PARA INVESTIGACIÓN BIOLÓGICA	28
2.5.1	<i>Deep Learning</i>	28
2.5.2	<i>Arquitectura de Redes Neuronales</i>	29
2.5.3	<i>Redes Neuronales Convolucionales</i>	30
2.5.3.1	Capa de convolución	31
2.5.3.2	Capa de agrupación	33
2.5.3.3	Capa totalmente conectada	34
2.5.4	<i>Métodos de entrenamamiento</i>	35
2.5.4.1	Aprendizaje Supervisado	36
2.5.4.2	Aprendizaje No Supervisado y Semi-Supervisado	37
2.5.5	<i>Aumento de Datos</i>	38
2.5.6	<i>Deep Learning para segmentación de núcleos</i>	39
2.5.6.1	Segmentación de imagen	39
2.5.6.2	Algoritmos de deep learning para segmentación de núcleos	40
2.6	ALGORITMO DE DESCENSO DE GRADIENTE.....	45
2.7	DISCUSIÓN	46
CAPITULO 3. GENERACIÓN DE BASE DE DATOS.....		47
3.1	INTRODUCCIÓN	47
3.2	GENERACIÓN DE BASE DE DATOS.....	47
3.2.1	<i>Adquisición y Procesamiento de las Imágenes</i>	47
3.2.2	<i>Segmentación manual</i>	48
3.2.2.1	Napari	49
3.2.3	<i>Proceso de segmentación</i>	50
3.3	DISCUSIÓN	52
CAPITULO 4. IMPLEMENTACIÓN		53
4.1	INTRODUCCIÓN	53
4.2	ARQUITECTURA DEL MODELO	53
4.3	ENTRENAMIENTO DEL MODELO	54
4.3.1	<i>Pre-procesamiento</i>	55
4.3.2	<i>Conjunto de entrenamiento</i>	56

4.3.3	<i>Pérdida, parámetros y entrenamientos del modelo</i>	57
4.4	DISCUSIÓN	58
CAPITULO 5. PREDICCIÓN Y RESULTADOS		59
5.1	INTRODUCCIÓN	59
5.2	DESAFÍOS EN LA PREDICCIÓN Y RECONSTRUCCIÓN DE IMÁGENES GRANDES.....	59
5.2.1	<i>Preparación de la imagen</i>	60
5.2.2	<i>Unión de parches</i>	61
5.3	POST-PROCESADO	63
5.3.1	<i>Metodología</i>	64
5.4	RESULTADOS.....	65
5.5	DISCUSIÓN	66
CAPITULO 6. CUANTIFICACIÓN DE RESULTADOS		67
6.1	INTRODUCCIÓN	67
6.2	CUANTIFICACIÓN DE RESULTADOS	67
6.3	DISCUSIÓN	68
CAPITULO 7. CONCLUSIONES		69
7.1	DISCUSIÓN	69
7.2	CONCLUSIONES	70
7.3	TRABAJO A FUTURO	71
CAPITULO 8. GLOSARIO		73
CAPITULO 9. REFERENCIAS		73
Anexo		83
A FUNDAMENTOS EN DEEP LEARNING		83
A.1	LOTES Y ÉPOCAS	83
A.2	TASA DE APRENDIZAJE	83
A.3	FUNCIÓN DE COSTO/PÉRDIDA	84
A.4	MÉTRICAS PARA LA EVALUACIÓN DEL RENDIMIENTO DE LOS MODELOS	86
B EVALUACIÓN DE RESULTADOS POR REGIÓN DEL LOBULILLO HEPÁTICO		89

Índice de Tablas

3.1	Cantidad de núcleos en cada sujeto (C1, C2 y C3) por región del lobulillo hepático. . . .	51
4.2	Inicio y fin de parches de 128 píxeles con un paso de 128 píxeles en los ejes X, Y y Z. . .	57
4.3	Parámetros de configuración del modelo utilizados en el experimento.	58
6.4	Resultados del rendimiento de los modelos en C3.	67
A.1	Resultados en la región periportal de C3.	89
A.2	Resultados en la región central de C3.	89
A.3	Resultados en la región pericentral de C3.	89
A.4	Resultados de la calidad de la segmentación en C3.	90
A.5	Precisión promedio para varios umbrales de IoU (τ) en la región periportal	90
A.6	Precisión promedio para varios umbrales de IoU (τ) en la region central	90
A.7	Precisión promedio para varios umbrales de IoU (τ) en la region pericentral	90

Índice de Figuras

2.1	Representación esquemática del eje CV-PV de un lobulillo hepático.	20
2.2	Modelo clásico del lobulillo hepático, unidad funcional del hígado.	21
2.3	Anatomía hepática y lóbulillo hepático.	22
2.4	Perfiles de Jablonski y Diagrama del cambio de Stokes.	24
2.5	Marcadores utilizados para cada tipo de célula y estructura en tejido hepático.	25
2.6	Script de segmentación de núcleos de Motion Tracking.	27
2.7	Diagrama de Relaciones de AI, ML, y DL.	29
2.8	Representación gráfica del modelo de neurona de McCulloch-Pitts o perceptrón y red neuronal retroalimentada de una red totalmente conectada.	30
2.9	Diagrama esquemático de una arquitectura básica de red neuronal convolucional.	31
2.10	Ejemplo de como una computadora ve una imagen como una serie de números.	31
2.11	Un ejemplo de operación de convolución con un tamaño de filtro de 3×3	32
2.12	Ejemplos de cómo los filtros en capas convolucionales extraen diferentes características de un tensor de entrada.	33
2.13	Relleno con ceros	34
2.14	Aplicación de una capa de agrupación máxima.	35
2.15	Aplanamiento (<i>flattening</i>) de una capa convolucional o de agrupación en un vector 1D.	36
2.16	Aprendizaje supervisado.	37
2.17	Diagrama de segmentación semántica y segmentación por instancias	40
2.18	Arquitectura U-Net.	41
2.19	Ejemplos de polígonos convexos y estrellados.	43
2.20	Arquitectura de StarDist 2D.	43
2.21	Arquitectura de Cellpose.	45
2.22	Ejemplo de la utilización del algoritmo de descenso de gradiente para estimar los centroides de cada núcleo.	46
3.23	Vistas ortogonales de una muestra núcleos hepáticos.	49
3.24	Vizualización y plugin de interpolación en Napari.	50

3.25	Proceso de interpolación de máscaras de núcleos 3D.	51
3.26	Imágenes y máscaras de núcleos hepáticos de C3.	52
4.27	Bloques de arquitectura convolucional.	54
4.29	Generación de flujo vectorial (gradiente en X, Y y Z).	54
4.28	Arquitectura del modelo 3D U-Net con bloques residuales, puertas de atención y 2 de- codificadores para una multitarea.	55
4.30	Comparación de imágenes sin y con pre-procesado.	56
5.31	Ejemplo de una reconstrucción de una imagen sin interpolación y con una interpolación sigmoide en el plano axial (XY) y en un plano ortogonal (YZ).	60
5.32	Proceso en el que se le agrega un relleno con ceros a la imagen que queremos predecir .	60
5.33	Ejemplo de una función lineal y una función sigmoide para la transición de α con 16 coeficientes.	61
5.34	Interpolación entre parches con solapamiento.	62
5.35	Proceso de interpolación entre parches adyacentes con solapamiento.	63
5.36	Resultados de la red.	63
5.37	Esquemas de la metodología del post-procesamiento para obtener una segmentación por instancia de los núcleos.	65
5.38	Resultado de segmentación de C3.	65
6.39	Precisión promedio para varios umbrales de IoU (τ) de los 3 modelos en la región peri- portal, central y pericentral	68
A.1	Early stopping.	84
A.2	Modelo de red neuronal con Dropout.	85
A.3	Tasa de aprendizaje.	86
A.4	Pérdida Focal.	87
A.5	Diagrama de intersección sobre unión (IoU).	87

Capítulo 1. Introducción

1.1 Introducción General

La comprensión de cómo los órganos se organizan y las células se coordinan para formar tejidos funcionales es fundamental en biología celular y del desarrollo [6]. La histología de los tejidos es crucial en el estudio y diagnóstico de diversas patologías, incluyendo el cáncer. Tradicionalmente, en las evaluaciones histopatológicas, tanto en entornos clínicos como investigación, se utilizan secciones de tejido bidimensional (2D). Esto resulta en pérdida de contexto tridimensional (3D) del tejido, generando un conocimiento incompleto del contexto tisular y los determinantes espaciales en varias patologías. Dificultando la comprensión de, por ejemplo, patrones de crecimiento de los tumores en su entorno local [7].

El hígado es un órgano crítico en el cuerpo humano que cumple una amplia gama de funciones, incluido el metabolismo, la desintoxicación, la síntesis de proteínas y la producción de sustancias bioquímicas que ayudan a la digestión [8]. Estas diversas funciones dependen de una arquitectura 3D compleja donde diferentes tipos de células coexisten e interactúan entre sí de manera coordinada, por lo que conocer la información espacial exacta de una célula es clave para comprender su función [9].

Para superar estas limitaciones, existen modelos geométricos que, a través de softwares como Motion Tracking (MT) [10], procesan las imágenes del tejido obtenidas mediante microscopía confocal con fluorescencia, generando una reconstrucción 3D de alta resolución de todos los componentes celulares y subcelulares importantes en el tejido, este caso hígado. Un paso clave en la reconstrucción es la segmentación de los núcleos celulares. En [10] se reporta que MT tiene una precisión del 90 % en la segmentación en las mejores condiciones. Aún así suele no detectar algunos núcleos y dejar pegados otros. Una modificación en los parámetros del software para intentar mejorar estos resultados puede llevar a la sobre segmentación de los núcleos que anteriormente estaban bien. Además si consideramos a núcleos que no son hepatocitos la calidad de la segmentación disminuye drásticamente.

Debido a la gran cantidad de núcleos presentes en las imágenes 3D (aproximadamente miles), corregir y modificar los resultados de la segmentación para analizar correctamente el tejido puede tomar desde semanas hasta meses por imagen, lo cual no es sostenible a largo plazo. Por esta razón, en los peores casos, se prefiere eliminar estos núcleos, lo que implica una pérdida parcial de información pero permite mantener la estructura general del tejido para su análisis. Se han aplicado muchos métodos tradicionales de procesamiento de imágenes, como morfología, extracción de características y crecimiento de regiones, para analizar imágenes de microscopía biológica [11]. Hoy en día, los métodos que han mostrado mejor

desempeño son los supervisados en aprendizaje profundo (DL) [1][12]. Consecuentemente, la disponibilidad de estas etiquetas se ha convertido en uno de los mayores cuellos de botella para métodos basados en datos [13][14], estando estos limitados en su mayoría a 2D.

En este contexto, se propone generar una base de datos de núcleos 3D de alta calidad basada en imágenes de microscopía confocal con fluorescencia. Esta base de datos se utilizará para entrenar una red neuronal 3D basada en una Residual Attention U-Net [15]. El objetivo es mejorar las herramientas disponibles para la reconstrucción 3D de tejido hepático. Esto permitirá disminuir el impacto negativo de una mala segmentación de núcleos. Así, se podrá analizar la información espacial de este tejido con mayor precisión y en menor tiempo. Además, se abre la posibilidad de trabajar con tejido afectado por cáncer o en proceso de regeneración, donde debido a las anomalías existentes en el contenido celular y sub-celular [16] la segmentación de los núcleos en este tejido es aún más compleja.

1.2 Objetivos

1.2.1 Objetivo General

Desarrollar un modelo basado en redes neuronales que produzca segmentaciones precisas y robustas de núcleos celulares 3D en tejido hepático.

1.2.2 Objetivos Específicos

- Generar una base de datos (GT) de núcleos hepáticos 3D en imágenes de microscopía de fluorescencia confocal mediante segmentación manual, utilizando el software Napari [17] en Python [18].
- Desarrollar un pipeline integral y entrenar un modelo de segmentación de núcleos celulares en imágenes 3D del tejido hepático, utilizando como GT la base de datos generada.
- Evaluar los resultados del modelo desarrollado en comparación con un modelo del estado del arte y la metodología del laboratorio (Motion Tracking), utilizando como GT la base de datos generada.

1.3 Alcances y Limitaciones

- Se colaboró con el laboratorio de Cell and Tissue Architecture Lab bajo la co-supervisión del Profesor Fabián Segovia. Los alumnos del laboratorio se encargaron de la adquisición y procesamiento de las imágenes de microscopía.
- Debido a la gran diferencia en la proporción de los distintos tipos de células presentes en las imágenes, como hepatocitos (aproximadamente 80 %), células endoteliales, células de Kupffer y células estrelladas, es posible que la segmentación de los núcleos de los tipos celulares menos representados tenga una menor precisión.
- Se utilizaron software y hardware de alto rendimiento del laboratorio de Cell and Tissue Architecture Lab para llevar a cabo el desarrollo del trabajo. El modelo fue entrenado en un ordenador con el sistema operativo Windows 11 Pro, utilizando un entorno virtual de Ubuntu 22.04 y TensorFlow 2.13.1 [19]. La configuración del hardware incluye un procesador AMD Ryzen Threadripper 3960X con 24 núcleos a 3.80 GHz, 256 GB de memoria RAM y una tarjeta gráfica NVIDIA GeForce RTX 3090 con 24 GB de memoria VRAM.
- El rendimiento del modelo depende tanto de la calidad y cantidad de las imágenes segmentadas. Solo se segmentaron imágenes de tejido hepático en ratones control, es decir, no tenían ninguna patología o condición asociada. Esto limita la evaluación del rendimiento del modelo únicamente a estos sujetos.

1.4 Metodología

En esta memoria, se propuso y desarrolló un pipeline integral para la segmentación de núcleos celulares en imágenes de microscopía de fluorescencia de tejido hepático, utilizando técnicas avanzadas de procesamiento de imágenes y redes neuronales profundas. A continuación, se describe la metodología empleada en este estudio:

1.4.1 Generación de la Base de Datos

1. **Selección de Muestras:** Se seleccionaron muestras de tejido hepático de tres sujetos de control (C1, C2 y C3) para garantizar la variabilidad y generalización en el entrenamiento del modelo.

2. **Segmentación Manual:** Utilizando el software Napari, se segmentaron manualmente aproximadamente 2900 núcleos 3D, que sirvieron como verdad fundamental o *ground truth* (GT) para el entrenamiento y evaluación del modelo.

1.4.2 Estudio Bibliográfico

Se realizó un estudio bibliográfico exhaustivo sobre los distintos tipos de redes neuronales y algoritmos utilizados para la segmentación de núcleos 3D en imágenes de microscopía. Además se utilizaron de manera preliminar redes de segmentación 2D como Stardist [1], Cellpose [3] y U-Net [20] para entender el funcionamiento y estructura general de la segmentación en imágenes a través de redes neuronales.

1.4.3 Desarrollo del Modelo de Segmentación

1. **Arquitectura del Modelo:** Se desarrolló una red neuronal teniendo como base una Residual Attention U-Net 3D [15], que combina la arquitectura U-Net con bloques residuales y mecanismos de atención para mejorar la precisión de la segmentación.
2. **Enfoque Multitarea:** Se decidió que la red neuronal fuera entrenada con un enfoque multitarea, realizando segmentación binaria de los núcleos y predicción de un campo vectorial para estimar los centroides de cada núcleo. Con el objetivo de poder segmentar individualmente cada núcleo de la muestra.

1.4.4 Entrenamiento del Modelo

Para el entrenamiento del modelo, se utilizaron las imágenes de los sujetos de control C1 y C2. Las imágenes segmentadas manualmente se dividieron en conjuntos de entrenamiento y validación para garantizar una evaluación adecuada del rendimiento del modelo durante el proceso de entrenamiento.

1.4.5 Predicción y Reconstrucción de Imágenes

El sujeto C3 se utilizó para evaluar el rendimiento del modelo. Las imágenes de C3 se dividieron en parches y se procesaron a través de la red neuronal. Para mitigar el efecto de borde en imágenes de

microscopía de gran tamaño, se aplicó un solapamiento entre parches y una función de interpolación para unir los parches adyacentes de manera continua y precisa.

1.4.6 Evaluación de Resultados

Para la evaluación de los resultados, se compararon los desempeños de la segmentación automática obtenida por nuestro modelo con el método de segmentación de núcleos de Motion Tracking y el modelo del estado del arte, Cellpose. Se utilizaron métricas estándar de segmentación, como el Intersección sobre Unión (IoU), precisión, recall y F1-Score, para cuantificar y comparar el rendimiento de cada método.

1.5 Temario

- (I) **Capítulo 1. Introducción:** Introducción en forma general del trabajo, indicando los objetivos y alcances del proyecto.
- (II) **Capítulo 2. Marco Teórico:** Se presenta una revisión bibliográfica sobre la función, estructura y arquitectura del hígado, las técnicas de microscopía de fluorescencia confocal y los conceptos fundamentales de DL. Además, se describen los algoritmos de segmentación de núcleos celulares y los métodos de evaluación de rendimiento.
- (III) **Capítulo 3. Generación de Base de Datos:** Se expone el proceso de generación de una base de datos de núcleos celulares 3D mediante segmentación manual utilizando el software Napari [17].
- (IV) **Capítulo 4. Implementación:** Se detallan la arquitectura del modelo de segmentación automática, el conjunto de entrenamiento, los parámetros y el proceso de entrenamiento del modelo, además del hardware y software utilizados.
- (V) **Capítulo 5. Predicción y Resultados:** Se expone la aplicación del modelo de segmentación en las imágenes de evaluación, los desafíos en la predicción y reconstrucción de estas imágenes, y se presentan los resultados obtenidos.
- (VI) **Capítulo 6. Cuantificación de Resultados:** Se evalúan los resultados obtenidos mediante métricas específicas, comparando el rendimiento del modelo en diferentes regiones del tejido hepático utilizando 3 métodos de segmentación distintos.
- (VII) **Capítulo 7. Conclusiones:** Se discuten las conclusiones finales del trabajo, resaltando los hallazgos más importantes y proponiendo posibles líneas de investigación y mejoras futuras.

1.6 Discusión

La memoria aborda de manera efectiva la necesidad de comprender la estructura y arquitectura 3D del hígado, así como las bases del aprendizaje profundo aplicadas a la segmentación de núcleos celulares. Además de desarrollar un pipeline integral capaz de segmentar núcleos en grandes volúmenes de imágenes. La implementación de un modelo de red neuronal convolucional, la Residual Attention U-Net 3D, ha demostrado ser eficaz para superar los desafíos de segmentación de núcleos hepáticos en comparación con métodos tradicionales y contemporáneos como Motion Tracking (MT) y Cellpose. La generación de una base de datos de alta calidad, con aproximadamente 2900 núcleos 3D segmentados manualmente, proporciona un fundamento sólido para el entrenamiento y evaluación del modelo. Los resultados obtenidos, que muestran mejoras significativas en las regiones periportal, central y pericentral del lobulillo hepático, validan la eficacia del enfoque adoptado. Este trabajo no solo contribuye al campo de la biología celular y del desarrollo, sino que también establece una base sólida para futuras investigaciones en la segmentación de tejidos más complejos, abriendo nuevas posibilidades para estudios en hígados patológicos y en procesos de regeneración.

Capítulo 2. Marco Teórico

2.1 Introducción

En este capítulo se presenta una revisión respecto a la función, estructura y arquitectura del hígado, el software Motion Tracking (MT) utilizado en el laboratorio para segmentar núcleos hepáticos, y además una revisión conceptual de Deep Learning (DL), Redes Convolucionales, Algoritmos de DL basado en redes convolucionales para segmentación de núcleos y su forma de evaluación para comparar y validar resultados.

2.2 Función y Estructura del Hígado

El hígado es un órgano vital que realiza numerosas funciones, como el almacenamiento de nutrientes, la desintoxicación, el metabolismo y la síntesis de proteínas. Es un vínculo entre los sistemas digestivo y circulatorio del cuerpo a través de dos redes, los Canalículos Biliares (BC) y la red endotelial sinusoidal. La red BC transporta la bilis secretada por los hepatocitos [21]. La red endotelial sinusoidal transporta la sangre, facilitando así el intercambio de metabolitos y nutrientes entre el hígado y otros órganos [21]. El tejido hepático está constituido principalmente por células parenquimatosas (hepatocitos), pero contiene otros tipos celulares importantes (células no parenquimatosas) como las células endoteliales sinusoidales (SECs), las células estrelladas hepáticas (HSCs) y las células de Kupffer KC.

2.2.1 Arquitectura del lóbulillo hepático

El hígado está compuesto por unidades funcionales, los lobulillos hepáticos. La definición clásica de lóbulillo describe esta subestructura hepática como un hexágono donde una vena central (CV) está situada en el centro y las tríadas portales, una vena porta (PV), una arteria hepática (HA) y un conducto biliar, están ubicados en los vértices. La región entre el centro del lobulillo y un vértice, es decir, el espacio entre CV y PV, se denomina eje porto-central o eje CV-PV. En cada lóbulo, los Canalículos Biliares y los sinusoides forman dos redes tridimensionales entre la CV y las tríadas portales. Estas redes forman dos sistemas de flujo opuestos: la sangre fluye desde la PV y la HA hacia la CV, y la bilis fluye contra corriente desde el área de la CV hacia el conducto biliar, Fig. 2.1.

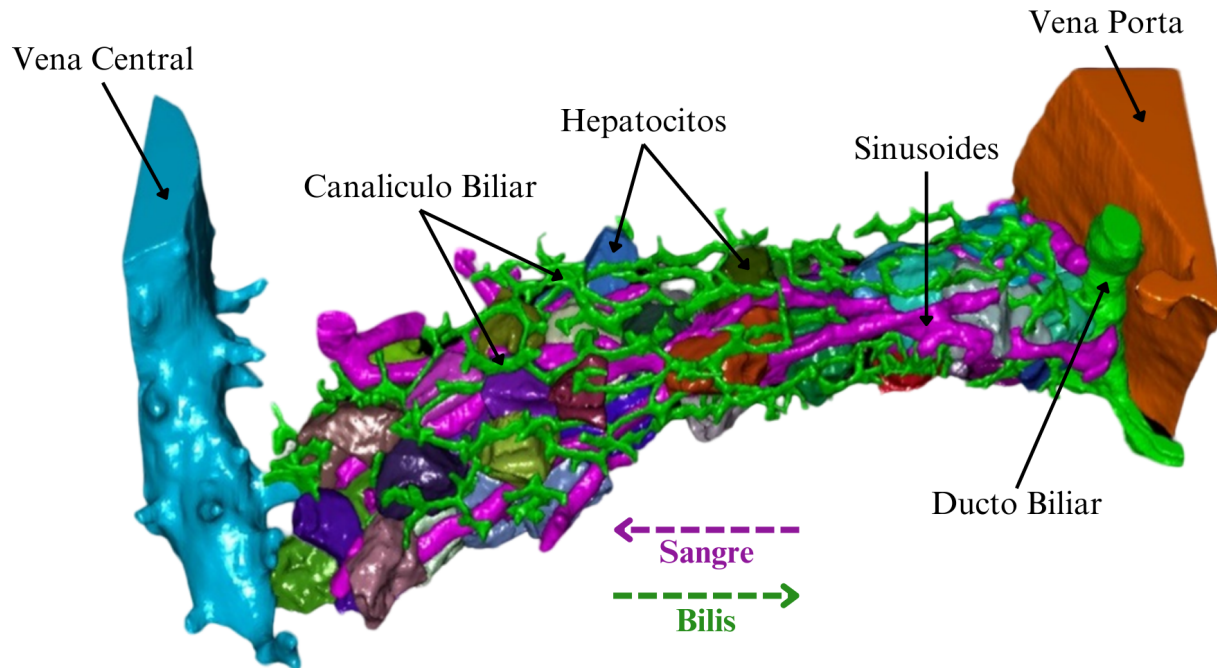


Fig. 2.1: Representación esquemática del eje CV-PV de un lobulillo hepático. El eje CV-PV corresponde a la región entre la vena central (CV) y la tríada portal (PV) de un lobulillo hepático. El eje está principalmente lleno de hepatocitos ($\approx 80\%$ del volumen hepático). Los dominios apicales de los hepatocitos forman una red tridimensional continua, el canaliculo biliar (BC), que transporta la bilis desde el área de la CV hasta el conducto biliar (mostrado en verde). Las células endoteliales sinusoidales (SECs) forman otra red (red sinusoidal, mostrada en magenta) que transporta sangre desde los vasos aferentes, la PV y la arteria hepática (mostrada en rojo), hacia la CV (mostrada en azul). Los hepatocitos intercambian nutrientes con la sangre a través de sus contactos con la red sinusoidal (contactos basales) y secretan bilis hacia el BC a través de los dominios apicales. Adaptado de [10].

2.2.2 Tipos celulares hepáticos

2.2.2.1 Hepatocitos

Los hepatocitos representan aproximadamente el 80% del volumen hepático (aproximadamente el 60% de la población total de células) y realizan las principales funciones del hígado tales como, desintoxicación, metabolismo y síntesis de proteínas. [21, 22].

2.2.2.2 Células no parenquimatosas

Las células no parenquimatosas comprenden solo aproximadamente el 6 % del volumen hepático total, pero desempeñan roles importantes en varias funciones hepáticas y en el crecimiento hepático. Los tres principales tipos de células no parenquimatosas en el hígado son las **células endoteliales sinusoidales** (SECs), las **células estrelladas** (HSC) y las **células de Kupffer** (KC). Las SECs forman una compleja red capilar que conecta los vasos aferentes con la CV y sirven para el transporte sanguíneo. Las SECs están en contacto con los hepatocitos a través de la matriz extracelular (ECM) y contienen pequeñas fenestraciones que facilitan la difusión de macromoléculas entre la sangre y los hepatocitos. Las células de Kupffer son macrófagos específicos del tejido ubicados dentro del capilar sinusoidal adherido a las SECs. Las KC actúan como mediadores de la respuesta inflamatoria, desempeñando un papel importante en la defensa inmunológica. Las células estrelladas se encuentran en el espacio de Disse entre las SECs y los hepatocitos. Sirven como almacenamiento de vitamina A y segregan componentes de la ECM. Además, las SECs y las células estrelladas están involucradas en el crecimiento hepático no solo por su propia proliferación, sino también afectando la proliferación de los hepatocitos, lo que muestra que la cooperación entre las células no parenquimatosas y los hepatocitos desempeña un papel importante en el mantenimiento y la función hepáticos.

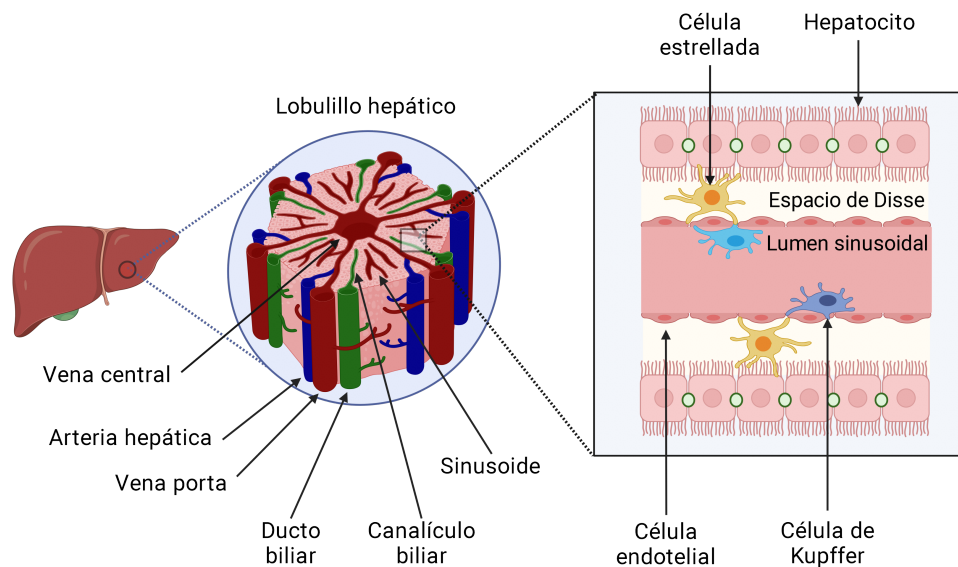


Fig. 2.2: Modelo clásico del lobulillo hepático, unidad funcional del hígado. Adaptado de [10, 23, 24].

2.2.3 Regiones del lobulillo hepático

Cada lóbulo del hígado tiene una disposición específica de las células y los vasos sanguíneos que se pueden dividir en diferentes regiones: la región periportal, la región pericentral, y la región central, Fig. 2.3.

- **Región Periportal (zona 1):** Regiones que rodean la triada portal (PV, HA y ducto biliar)
- **Región Pericentral (zona 3):** Regiones adyacentes a la vena central.
- **Región Central (zona 2):** Se refiere a las células entre estas regiones, también conocidas como hepatocitos lobulillares medios.

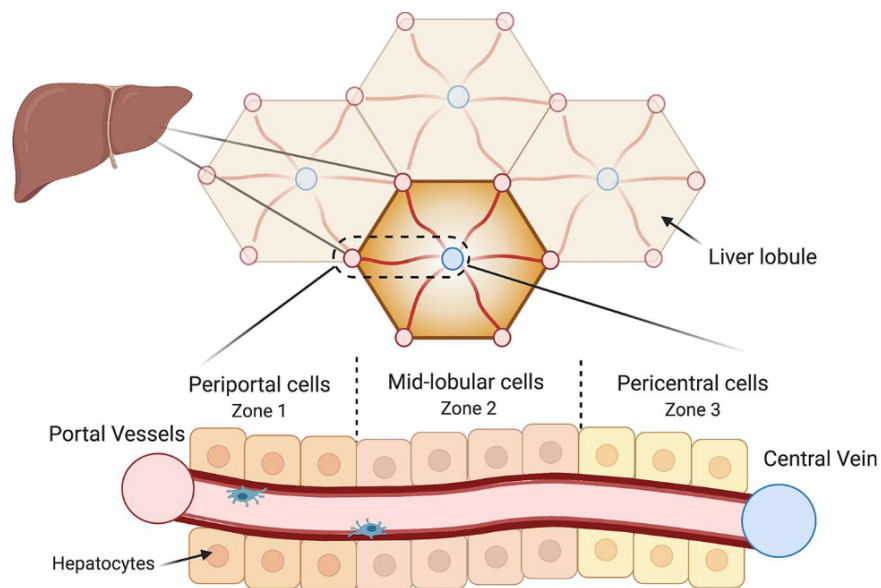


Fig. 2.3: Anatomía hepática y lóbulo hepático. El hígado está compuesto por unidades hexagonales llamadas lobulillos. Los hepatocitos periportales se encuentran en la periferia del lobulillo, seguidos de las células lobulillares centrales y, finalmente, los hepatocitos pericentrales rodean la vena central. Adaptado de [25].

Además, se ha visto en [10, 26] que la distribución de células inmunitarias y otros componentes en el hígado (como los hepatocitos, KC, SECs y HSC) están organizados de manera distinta según el área del tejido hepático, y que esta organización no es aleatoria sino que está relacionada con su función.

2.3 Fluorescencia y Microscopía

2.3.1 Fenómeno de Fluorescencia

La fluorescencia es un fenómeno físico que ocurre cuando una molécula fluorescente (conocida como fluoróforo o fluorocromo) absorbe un fotón de luz, lo que excita a la molécula a un estado energético más alto. Posteriormente, la molécula retorna a su estado fundamental, emitiendo un fotón de menor energía (mayor longitud de onda) en el proceso. Este fenómeno se puede describir en términos de los diagramas de energía de Jablonski, Fig. 2.4, donde se ilustran las transiciones electrónicas que tienen lugar durante la absorción y emisión de luz.

Cuando una molécula fluorescente absorbe energía luminosa, ocurre una excitación de un electrón desde el estado fundamental S_0 al primer estado excitado S_1 . Esta energía absorbida proviene generalmente de una fuente de luz externa, como un láser. Una vez que el electrón se encuentra en el estado excitado, la molécula puede perder parte de esta energía a través de vibraciones internas, un proceso conocido como relajación vibracional [27]. Finalmente, el electrón regresa al estado fundamental, emitiendo un fotón con menor energía que el fotón absorbido inicialmente. Esta emisión de energía en forma de luz es lo que se observa como fluorescencia.

La diferencia entre la longitud de onda de la luz absorbida y la luz emitida se conoce como el desplazamiento de Stokes (*Stokes shift*) [28]. Este desplazamiento determina que tan fácilmente se puede separar las señales de excitación y emisión o de la emisión de un segundo fluoróforo dentro de una muestra etiquetada múltiplemente, permitiendo a los investigadores distinguir la fluorescencia de fondo de la señal específica de los fluoróforos.

2.3.2 Microscopía de Fluorescencia

La microscopía de fluorescencia utiliza el fenómeno de fluorescencia para visualizar estructuras y moléculas específicas dentro de muestras biológicas. Se ha convertido en una herramienta esencial en la biología celular. Esta técnica permite a los investigadores visualizar tejido, células, orgánulos individuales y conjuntos de macromoléculas dentro de la célula [30]. Estas células son generalmente translúcidas en su estado natural, y el gran número de proteínas y macromoléculas son ópticamente indistinguibles entre sí, por lo que identificar una sola proteína u otra molécula de interés no es una tarea fácil y es similar a tratar de encontrar a una aguja en un pajar [31].

En la microscopía de fluorescencia, la muestra es iluminada con luz de una longitud de onda espe-

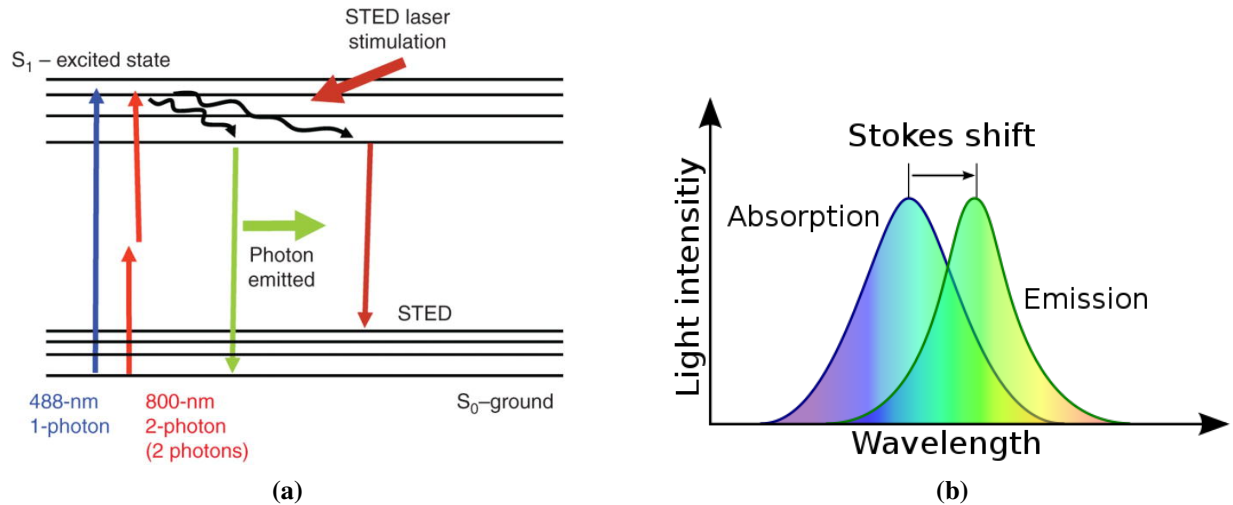


Fig. 2.4: Perfiles de Jablonski y Diagrama del cambio de Stokes. **(a)** Cambios en el estado electrónico de indicadores fluorescentes durante la excitación y emisión de fotones (perfiles de Jablonski). La excitación (de S_0 a S_1) inducida por luz láser de 488 nm (azul) requiere un fotón y para dos fotones de 800 nm (rojo) requiere dos fotones. Después de la relajación a los niveles de energía más bajos, la transición inversa (de S_1 a S_0) libera un fotón de longitud de onda más larga (verde). La incidencia de fotones a 592 nm (longitud de onda de depleción) induce la transición de S_1 a niveles altos de S_0 y la supresión de la fluorescencia. **(b)** Diagrama del cambio de Stokes entre los espectros de luz de absorción y emisión. Adaptado de [28, 29].

cífica que excita los fluoróforos presentes en la muestra. Los fluoróforos emiten luz de una longitud de onda mayor, que es recogida y enfocada por los lentes del microscopio. Los filtros ópticos se utilizan para separar la luz de excitación de la luz de emisión, asegurando que solo la fluorescencia específica de los fluoróforos sea observada y registrada [31].

Una de las principales ventajas de la microscopía de fluorescencia es la capacidad de marcar múltiples componentes celulares con diferentes fluoróforos, permitiendo la visualización simultánea de varias estructuras dentro de la misma célula. Esto es posible gracias a la amplia gama de fluoróforos disponibles, cada uno con características espectrales únicas [31].

2.3.3 Microscopía de Fluorescencia Confocal

La microscopía de fluorescencia confocal es una técnica avanzada de obtención de imágenes que mejora la resolución óptica en comparación con la microscopía de fluorescencia tradicional. Esta técnica utiliza un láser para escanear la muestra punto por punto y un orificio (*pinhole*) para eliminar la luz desenfocada, lo que permite obtener imágenes nítidas de un plano específico en la muestra. La capacidad de la microscopía confocal para obtener una alta resolución axial (z) es particularmente útil para crear

imágenes tridimensionales (3D) mediante la recolección de una serie de planos z (z -stack) y su posterior reconstrucción a través de softwares [32, 33].

En contraste con la microscopía de fluorescencia tradicional, donde toda la muestra se ilumina de manera uniforme y simultánea, la microscopía confocal enfoca la luz en puntos específicos de la muestra, reduciendo la interferencia de la luz de fondo. Este método permite resaltar los aspectos 3D de la muestra al utilizar fuentes de luz potentes, como láseres, para escanear múltiples niveles de la muestra [34].

Los microscopios confocales son controlados por computadoras y software especializado, lo que significa que las imágenes no se observan directamente a través de los oculares, sino en el monitor de la computadora. Este software permite ajustar la ganancia de excitación y obtener múltiples secciones en el plano z , facilitando la creación de reconstrucciones 3D precisas de muestras gruesas [34].

Entre las aplicaciones de la microscopía confocal en las ciencias biomédicas se encuentran el análisis de colocalización de biomoléculas, el estudio de cambios en la expresión y distribución de moléculas en organelos, y el análisis 3D de componentes celulares. La técnica es invaluable para obtener imágenes detalladas de tejidos y células, permitiendo una mejor comprensión de la estructura y función biológica [32].

2.3.4 Marcadores Fluorescentes

Las tinciones que se utilizan en el laboratorio para observar las distintas estructuras de interés en el tejido hepático se observan en Fig. 2.5.

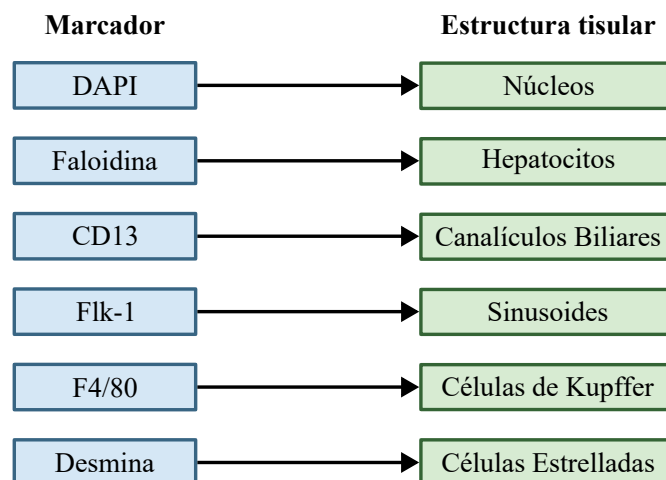


Fig. 2.5: Marcadores utilizados para cada tipo de célula y estructura en tejido hepático. Elaboración propia.

Además, como tenemos que generar una segmentación de núcleos hepáticos, nos centramos en imá-

genes de microscopía que tienen el marcador DAPI. El cual específicamente se une al ADN y permite la visualización clara de los núcleos en las imágenes de microscopía.

2.3.5 Formatos de imagen utilizados

2.3.5.1 Archivos “.tiff”

Un archivo TIFF, cuyo nombre es la sigla de “Tag Image File Format” (formato de archivo de imágenes con etiquetas), es un archivo informático que se emplea para almacenar información de imágenes y gráficos rasterizados [35]. El TIFF es una forma práctica de almacenar imágenes de alta calidad antes de editarlas para evitar formatos de archivo con pérdida de datos. Estas imágenes se visualizan comúnmente con el software Fiji [36] y suelen pesar sobre los 800 MB.

2.4 Segmentación de núcleos a través de Motion Tracking

Motion Tracking (MT) es una plataforma de software diseñada para el análisis y reconstrucción tridimensional de tejidos a partir de imágenes de microscopía confocal. Este software combina una variedad de algoritmos nuevos y establecidos para identificar estructuras celulares y crear modelos digitales precisos del tejido. Es especialmente útil para estudios en biología celular y del desarrollo, permitiendo a los investigadores visualizar cómo interactúan las células para formar tejidos complejos. Utiliza scripts con funciones dentro del software que permite realizar las tareas más eficientemente, Fig. 2.6.

La plataforma MT utiliza varias técnicas tradicionales de procesamiento de imágenes para lograr su objetivo de reconstrucción tridimensional precisa. Algunas de estas técnicas incluyen:

- (a) **Eliminación de Ruido (*Denoising*):** La eliminación de ruido (*denoising*) se realiza utilizando métodos tradicionales como el filtro de mediana, filtros de difusión y una umbralización para eliminar el fondo (*background*).
- (b) **Generación del Mallado de los Núcleos:** A partir de la segmentación de los núcleos (*foreground*), se realizan nuevamente operaciones de filtro mediana y se genera un mallado o triangulación de esta. Esta triangulación proporciona una base para la reconstrucción tridimensional del contenido nuclear de la imagen.
- (c) **Generación de la Máscara de Núcleos:** Muchos de los núcleos obtenidos del paso anterior se encuentran pegados en el mallado, lo que resulta en que, al generar la segmentación de los núcleos

Script	
[Channel 1 -> Prc 1] Median Smooth ()	
[Prc 1 -> Prc 1] Diffusion (100, 0.01, TRUE, 10)	
[Prc 1 -> Prc 1] Diffusion (50, 0.05, TRUE, 5)	
[Prc 1 -> Prc 1] Median Smooth ()	(a)
[Prc 1 -> Prc 2] Arithmetics (1, 7, 1, 0, 1, 100, TRUE)	
[Prc 1 -> Prc 3] Segmentation (40, 80, 1, FALSE, FALSE, TRUE, 2, FALSE)	
[Prc 1 -> Prc 4] Segmentation (40, 80, 1.2, FALSE, FALSE, TRUE, 2, FALSE)	
[->] Pause ()	
[Prc 3 -> Prc 5] Median Smooth ()	
[Prc 5 -> Prc 4] Triangulation (1, 1)	(b)
[Prc 4 -> Prc 4] Fill Triangle Mesh (1, FALSE, TRUE)	
[Prc 4 -> Prc 4] Triangulation (1, 1)	
[->] Pause ()	
[Channel 1 -> Prc 6] Median Smooth ()	
[Prc 6 -> Prc 6] Blur (0.6)	(c)
[Prc 6 -> Prc 7] Apply Segmened Mask (3, FALSE)	
[->] Pause ()	
[Channel 1 -> Prc 1] Apply ROI Mask (FALSE)	
[Prc 1 -> Prc 2] Triangulation (1, 1)	(d)
[Prc 2 -> Main Frame] Find Nuclei (1, 1)	
[->] Pause ()	
[Channel 1 -> Main Frame] Find Nuclei (1, 1)	
[Main Frame -> Proc. Frame] Split Nuclei by Curvature (550, 0.3, 0.005, FALSE, 50, 1, 1, FALSE)	(e)
[Proc. Frame -> Proc. Frame] Split Nuclei by Curvature (550, 0.3, 0.001, FALSE, 50, 1, 1, FALSE)	
[Proc. Frame -> Prc 3] Fill Nuclei ()	

Fig. 2.6: Script de segmentación de núcleos de Motion Tracking. **(a)** Preprocesamiento, eliminación del ruido (denoising) y segmentación de núcleos. **(b)** Generación del mallado de los núcleos (triangulación). **(c)** Generación de la máscara de núcleos para aplicarle el algoritmo de Watershed. **(d)** Aplicación de la máscara obtenida a través de Watershed a la triangulación de los núcleos obtenido en (b). **(e)** Splitting de la triangulación de los núcleos muy densos.

individuales, estos quedarían con el mismo Id, considerándolos como un solo núcleo. Para solucionar este problema, dentro del script se genera una máscara del contenido nuclear de la imagen, lo que permite separar los núcleos en Fiji [36] a través del algoritmo de Watershed [37].

- (d) **Obtención de mallado de núcleos:** Una vez se obtiene la máscara de los núcleos separados individualmente en Fiji, con la máscara y el mallado del contenido nuclear obtenido anteriormente, se genera un mallado de los núcleos separados individualmente.
- (e) **Separación de la Triangulación de Núcleos Muy Densos:** El último paso del script de MT se emplea en casos donde los núcleos están muy densamente agrupados y no se han logrado dividir correctamente a través del algoritmo de Watershed en Fiji. En estas situaciones, MT aplica técnicas de splitting para dividir estas regiones densas en segmentos más pequeños basándose en la forma de los núcleos. Aunque esta técnica puede ser efectiva, a veces puede resultar en una sobresegmentación de núcleos que previamente habían sido segmentados correctamente.

La mayoría de las operaciones en MT requieren ajustes específicos de los parámetros para cada imagen. A medida que aumenta el número de muestras, se vuelve cada vez más difícil y laborioso segmentar las imágenes de manera eficiente y precisa, ya que cada una necesita un ajuste individual. Este proceso de ajuste manual es ineficiente en términos de tiempo, lo que representa un desafío significativo cuando se trabaja con grandes volúmenes de datos.

2.5 Deep Learning para Investigación Biológica

La microscopía desempeña un papel indispensable en la investigación biológica. Con el desarrollo de la óptica y la informática, las tecnologías avanzadas de microscopía han abierto una nueva perspectiva a los investigadores [11]. Las nuevas generaciones de microscopios permiten adquirir más imágenes que nunca a una velocidad mucho mayor, con una mayor resolución espacial y un campo de visión más amplio. Por esto durante la última década, la cantidad y complejidad de las imágenes aumentaron exponencialmente, traduciéndose en la necesidad de análisis más complejos en las imágenes [38]. Es así, que el utilizar métodos tradicionales para analizar las imágenes es una tarea muy desafiante y compleja, y que debido a sus limitaciones, automatizar esta tarea se ha vuelto una prioridad. Es así que métodos de inteligencia artificial (AI), como el aprendizaje profundo (DL), se han desarrollado y han revolucionado las técnicas de análisis de imágenes, dando resultados excepcionales en la segmentación de núcleos y otras ramas [11].

2.5.1 Deep Learning

DL es un subconjunto del aprendizaje automático (ML) (y un subconjunto de la AI, Fig. 2.7), que consiste esencialmente en una red neuronal con tres o más capas. Estas redes neuronales, llamadas redes neuronales artificiales (ANNs) están diseñadas para imitar, en cierta medida, el funcionamiento del cerebro humano, aunque todavía están lejos de igualar su capacidad. La principal habilidad de las redes de neuronales radica en su capacidad para “aprender” patrones complejos a partir de grandes volúmenes de datos. Aunque una red neuronal con una sola capa puede realizar predicciones aproximadas, la inclusión de capas ocultas adicionales desempeña un papel crucial en la mejora y refinamiento de la precisión del modelo, permitiendo una optimización más profunda y compleja de los patrones en los datos, lo que se traduce en una capacidad predictiva más precisa y sofisticada [39].

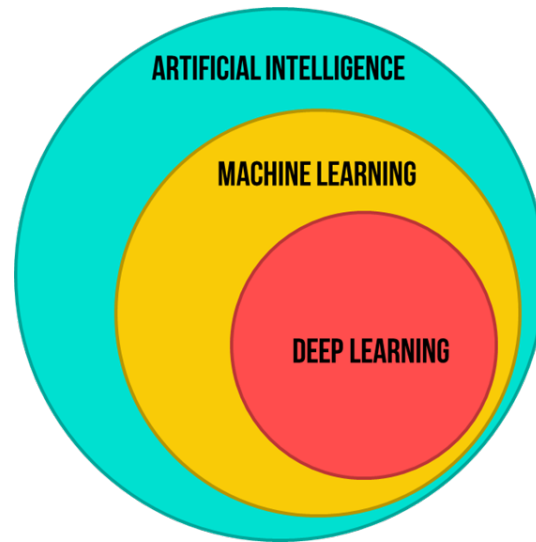


Fig. 2.7: Diagrama de Relaciones de AI, ML, y DL. Adaptado de [40].

2.5.2 Arquitectura de Redes Neuronales

Para describir las redes neuronales, empezaremos por describir la red neuronal más simple posible, la que consta de una sola “neurona”. Esta red neuronal simple se denomina perceptrón, Fig 2.8a. En este caso, la entrada (*input*) se mapea directamente a la salida (*output*) mediante una función conocida como “función de activación” o “función de transferencia”. Las conexiones de la entrada ($x_1 \dots x_N$) a la salida contienen los pesos ($w_1 \dots w_N$), con los que se multiplican las características y se añaden en el nodo de salida después de que la función de activación convierta la salida en una etiqueta de clase. Este tipo de red neuronal se conoce como red neuronal de una sola capa [41].

Las redes neuronales se modelan como una colección de neuronas individuales organizadas en capas, sin ciclos entre ellas. Están principalmente compuestas por tres capas: la de entrada, la de salida y una o más capas ocultas. La capa de entrada recibe los datos y los pasa a las capas ocultas. En las capas ocultas, se aplican pesos a las entradas, se dirigen las entradas a través de una función de activación y se envían sus salidas a la capa de salida. Esta arquitectura se conoce como propagación hacia adelante (*feed-forward*), Fig. 2.8b, porque las capas sucesivas se alimentan entre sí en dirección hacia adelante, desde la entrada hasta la salida. Después pasa a la capa de salida donde se utiliza una función de activación que convierte los valores en probabilidades. Estos valores luego pueden pasar a una función de umbral para determinar las puntuaciones de clase (por ejemplo, en clasificación) o un objetivo de valor real (por ejemplo, en regresión) [42]. Esta salida luego se compara con la salida original y se conoce el error, para actualizar los pesos en la retropropagación (*back-propagation*) y reducir el error. Este proceso continúa durante un cierto número de épocas (o iteraciones) y, finalmente, los pesos del modelo se actualizan y se

realiza la predicción.

La capacidad de aprendizaje de las redes neuronales depende en gran medida del peso entre las neuronas. Para las redes neuronales regulares, el tipo de capa más común es la capa completamente conectada, en la que las neuronas entre dos capas adyacentes están completamente conectadas entre sí. A estos tipos de redes se les llama **redes neuronales completamente conectadas** (DNN), Fig. 2.8b. Por lo tanto, la arquitectura de la red neuronal está casi completamente definida una vez que se han definido el número de capas y el número/tipo de nodos en cada capa [43].

Las redes neuronales existen en diferentes variedades y se utilizan con frecuencia en diversas aplicaciones de aprendizaje automático y aprendizaje profundo. Las redes neuronales convolucionales son las más comunes en procesamiento de imágenes.

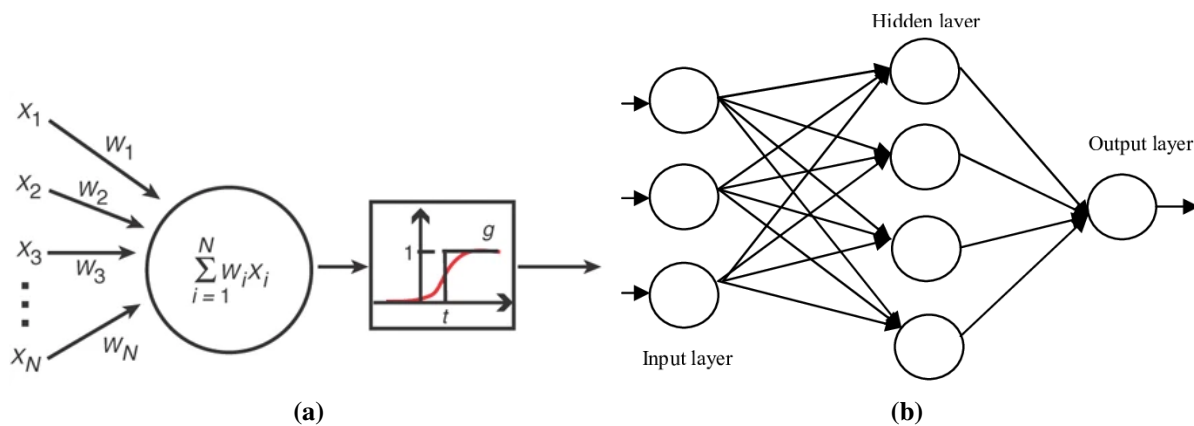


Fig. 2.8: (a) Representación gráfica del modelo de neurona de McCulloch-Pitts o perceptrón. (b) Red neuronal retroalimentada de una red totalmente conectada. La red mostrada toma tres entradas, cuatro unidades en la capa oculta y una salida. Adaptado de [44, 45].

2.5.3 Redes Neuronales Convolucionales

Redes neuronales convolucionales (CNN) es un tipo de modelo de DL diseñado para procesar datos que tienen un patrón de rejilla, como las imágenes compuesta por píxeles, y está inspirado en la organización del córtex visual de los animales [46, 47]. Está diseñado para aprender jerarquías espaciales de características de manera automática y adaptativa, desde patrones de bajo nivel hasta patrones de alto nivel. La CNN es una construcción matemática que generalmente se compone de tres tipos de capas (o bloques): capas de convolución (*convolution layers*), capas de agrupación (*pooling layers*) y capas totalmente conectadas (*fully connected layers*). Las dos primeras, capas de convolución y de agrupación, realizan la extracción de características, mientras que la tercera, una capa completamente conectada, asigna las características extraídas a una salida final, como la clasificación, Fig. 2.9.

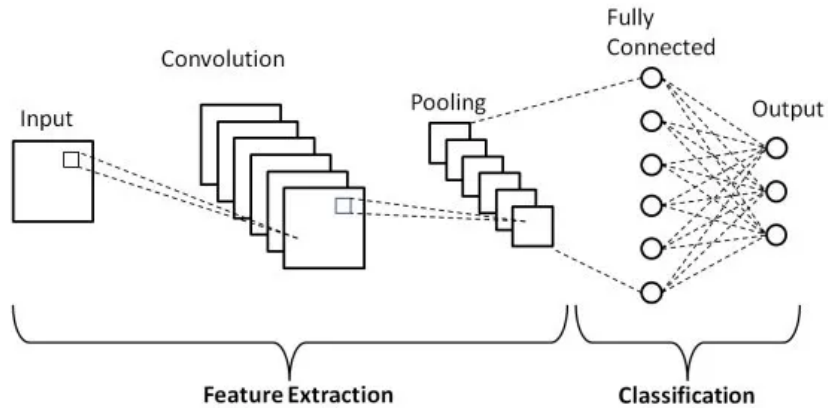


Fig. 2.9: Diagrama esquemático de una arquitectura básica de red neuronal convolucional. Adaptado de [48].

2.5.3.1 Capa de convolución

Una **capa de convolución** juega un papel clave en las CNN, y está compuesta por una serie de operaciones matemáticas, como la convolución, un tipo especializado de operación lineal. En las imágenes digitales, los valores de los píxeles se almacenan en una rejilla 2D, es decir, en una matriz de números o también llamado tensor, Fig. 2.10, y en cada posición del tensor se aplica un pequeño conjunto de parámetros llamado filtro (*kernel*), que calcula un producto elemento a elemento entre cada elemento del filtro y el tensor de entrada en cada ubicación del tensor, y luego se suman para obtener el valor de salida en la posición correspondiente del tensor de salida, obteniendo un mapa de características, Fig. 2.11.

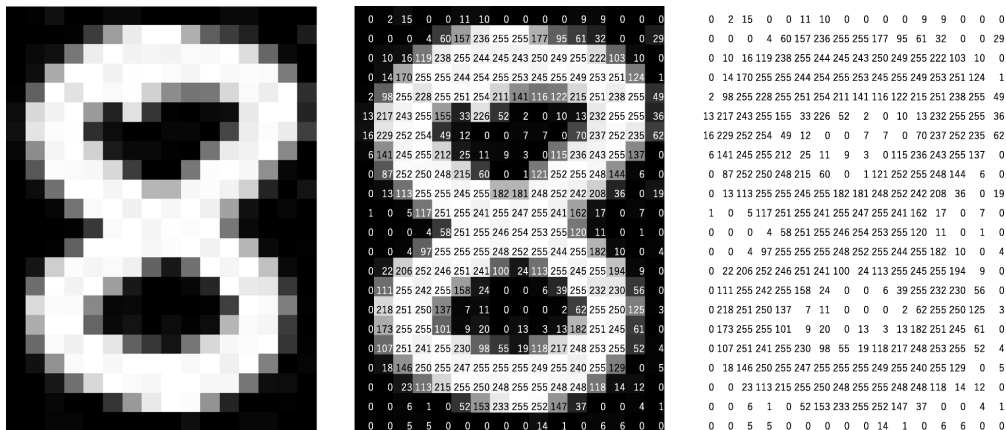


Fig. 2.10: Ejemplo de como una computadora ve una imagen como una serie de números. La matriz de la derecha contiene números entre 0 y 255, cada uno de los cuales corresponde al brillo de los píxeles en la imagen de la izquierda. Ambos están superpuestos en la imagen del medio. Adaptado de [49].

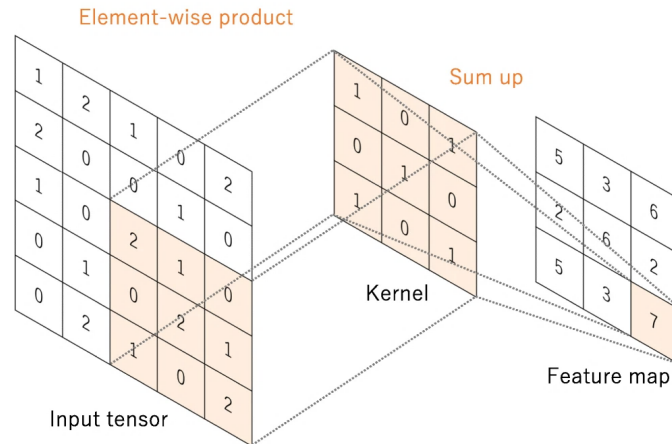


Fig. 2.11: Un ejemplo de operación de convolución con un tamaño de filtro de 3×3 . El tensor se calcula en cada ubicación y se suma para obtener el valor de salida en la posición correspondiente del tensor de salida, llamado mapa de características. Adaptado de [50].

Este procedimiento se repite aplicando múltiples filtros para formar un número arbitrario de mapas de características, que representan diferentes características de los tensores de entrada. Por lo tanto, diferentes filtros pueden considerarse como diferentes extractores de características, Fig. 2.12. Dos hiperparámetros clave que definen la operación de convolución son el tamaño y el número de filtros. El primero suele ser 3×3 , pero a veces es 5×5 o 7×7 . El último es arbitrario y determina la profundidad de los mapas de características de salida [50].

La operación de convolución descrita anteriormente no permite que el centro de cada filtro se superponga al elemento más externo del tensor de entrada, lo que reduce el ancho y la altura del mapa de características de salida en comparación con el tensor de entrada. El relleno (*padding*), usualmente el **relleno con ceros** (*zero padding*), es una técnica para abordar este problema, donde se agregan filas y columnas de ceros en cada lado del tensor de entrada, de manera que el centro de un núcleo se ajuste al elemento más externo y se mantenga la misma dimensión en el plano a través de la operación de convolución (Fig. 2.13). Las arquitecturas modernas de las CNN suelen emplear el relleno con ceros para mantener las dimensiones en el plano y así aplicar más capas. Sin el relleno con ceros, cada mapa de características sucesivo se volvería más pequeño después de la operación de convolución [50].

La distancia entre dos posiciones sucesivas del filtro se llama **paso**, lo que también define la operación de convolución. La elección común para el paso es uno, sin embargo, a veces se utiliza un paso mayor para lograr un submuestreo de los mapas de características. Una técnica alternativa para realizar el submuestreo es una operación de agrupación, como se describirá luego.

La característica clave de una operación de convolución es el uso compartido de pesos: los núcleos se comparten en todas las posiciones de la imagen. El uso compartido de pesos crea las siguientes caracte-

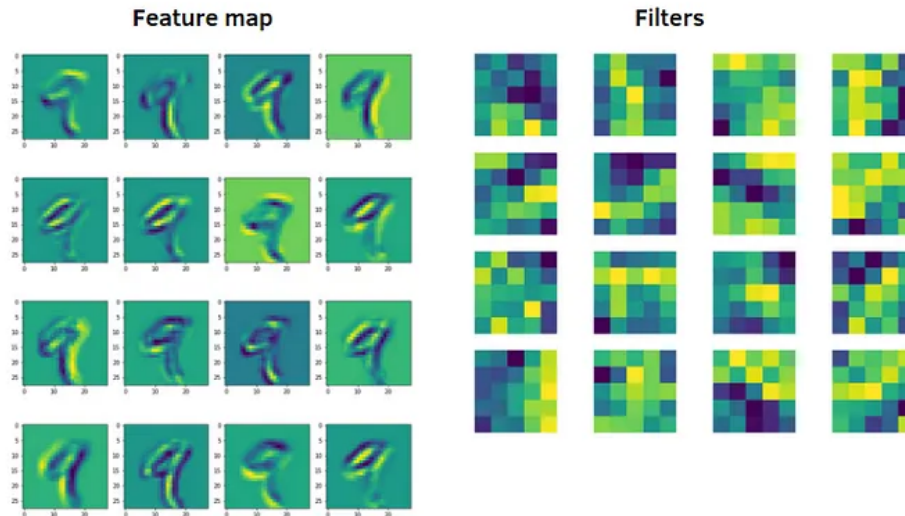


Fig. 2.12: Ejemplos de cómo los filtros en capas convolucionales extraen diferentes características de un tensor de entrada. Adaptado de [51].

rísticas de las operaciones de convolución:

1. Permitir que los patrones de características locales extraídos por los núcleos sean invariables a la traslación a medida que los núcleos se desplazan por todas las posiciones de la imagen y detectan patrones locales aprendidos.
2. Aprender jerarquías espaciales de patrones de características mediante el submuestreo en conjunto con una operación de agrupamiento, lo que resulta en capturar un campo de visión cada vez más grande.
3. Aumentar la eficiencia del modelo al reducir el número de parámetros a aprender en comparación con las redes neuronales completamente conectadas.

2.5.3.2 Capa de agrupación

Es común insertar periódicamente una **capa de agrupación** entre capas convolucionales sucesivas en una arquitectura de CNN. Su función es reducir progresivamente el tamaño espacial de la representación para disminuir la cantidad de parámetros y cálculos en la red, y así también controlar el sobreajuste. La capa de agrupación opera de forma independiente en cada “corte” de profundidad de la entrada y la redimensiona espacialmente utilizando la operación MAX. La forma más común es una capa de agrupación

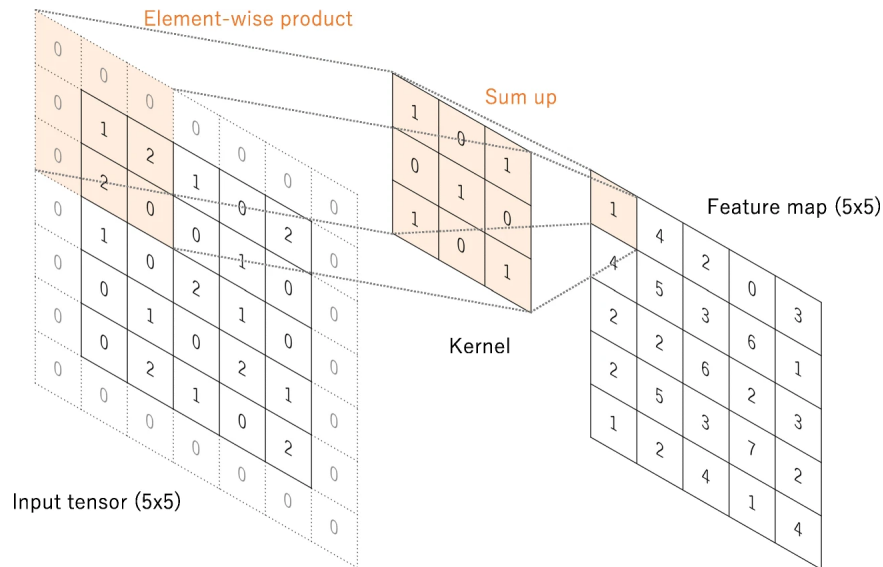


Fig. 2.13: Una operación de convolución con relleno de ceros para conservar las dimensiones en el plano. Notar que la dimensión de entrada de 5×5 se mantiene en el mapa de características de salida. En este ejemplo, se establece un tamaño de filtro y longitud de paso de 3×3 y 1, respectivamente. Adaptado de [50].

con filtros de tamaño 2×2 aplicados con un paso de 2, lo que reduce a la mitad cada corte en la entrada a lo largo del ancho y la altura, descartando el 75 % de la información. En este caso, cada operación MAX estaría tomando el máximo de 4 números (una pequeña región 2×2 en alguna corte de profundidad) [52].

2.5.3.3 Capa totalmente conectada

La **capa totalmente conectada** es simplemente una red neuronal retroalimentada (*feed forward neural networks*), cuya entrada son los mapas de características de salida de la última capa de convolución o de agrupación que generalmente se aplanan, es decir, se transforman en una matriz de números (o vector) unidimensional (1D). Esta luego se conecta a una o más capas completamente conectadas, también conocidas como capas densas, Fig. 2.15. Por tanto, sus activaciones pueden calcularse con una multiplicación de matrices más un sesgo, ecuación 2.5.1. Una vez que las características extraídas por las capas convolucionales y reducidas por las capas de agrupación son creadas, estas son mapeadas por un subconjunto de capas completamente conectadas a la salida final de la red, como las probabilidades de cada clase en las tareas de clasificación. La capa final completamente conectada generalmente tiene la misma cantidad de nodos de salida que la cantidad de clases. Cada capa completamente conectada va seguida de una función de activación.

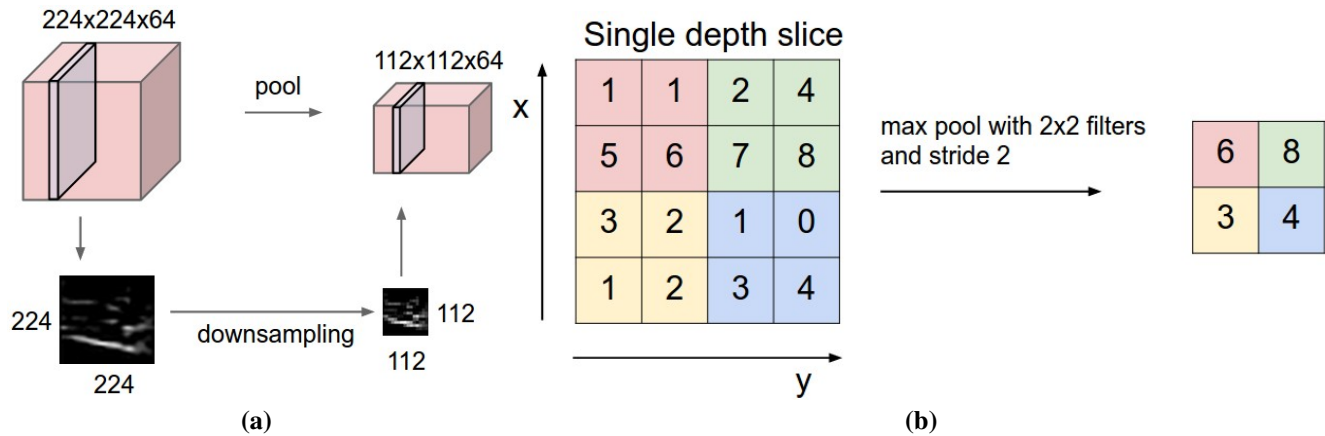


Fig. 2.14: Aplicación de una capa de agrupación máxima. La capa de agrupación reduce el tamaño espacial del volumen de manera independiente en cada corte de profundidad del volumen de entrada. **(a)** El volumen de entrada de tamaño $[224 \times 224 \times 64]$ se agrupa con un tamaño de filtro de 2 y un paso de 2, resultando en un volumen de salida de tamaño $[112 \times 112 \times 64]$, que conserva la profundidad del volumen. **(b)** La operación de reducción de resolución más común es el máximo, dando lugar a la **agrupación máxima (max pooling)**, que aquí se muestra con un paso de 2. Es decir, se toma el máximo en cada conjunto de 4 números (un pequeño cuadrado 2×2). Adaptado de [52].

$$z = g(Wx + b) \quad (2.5.1)$$

Donde, x es el vector de entrada con dimensión $[p_l, 1]$, W es la matriz de pesos con dimensiones $[p_l, n_l]$ donde p_l es el número de neuronas en la capa anterior y n_l es el número de neuronas en la capa actual, b es el vector de sesgo con dimensión $[p_l, 1]$, g es la función de activación y z es la salida.

Por último, el tener un extractor de características optimizable, hace que las CNN sean altamente eficientes para el procesamiento de imágenes, ya que una característica puede aparecer en cualquier lugar de la imagen. Además, a medida que una capa alimenta su salida en la siguiente, las características extraídas pueden volverse jerárquica y progresivamente más complejas [54].

2.5.4 Métodos de entrenamiento

Los algoritmos de aprendizaje automático pueden ser ampliamente categorizados como **supervisados** o **no supervisados** según el tipo de experiencia que se les permite tener durante el proceso de aprendizaje. La mayoría de los algoritmos de aprendizaje se pueden entender como aquellos que tienen permitido experimentar con un conjunto de datos (*dataset*) completo, donde un conjunto de datos es una colección de muchos ejemplos [43].

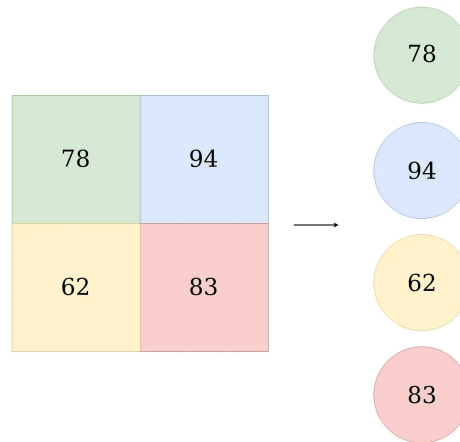


Fig. 2.15: Aplanamiento (*flattening*) de una capa convolucional o de agrupación en un vector 1D. Adaptado de [53].

2.5.4.1 Aprendizaje Supervisado

La mayoría de las aplicaciones prácticas del aprendizaje automático utilizan el **aprendizaje supervisado** (*supervised learning*), en el cual se tienen variables de entrada (x), una variable de salida (Y) y se utiliza un algoritmo que aprende la función que mapea o asigna las entradas a la salida, representada como $Y = f(X)$. El objetivo es aproximar tan bien la función de asignación que cuando se tengan nuevos datos de entrada (x), se puedan predecir las variables de salida (Y) para esos datos.

Se llama aprendizaje supervisado porque el proceso de un algoritmo aprendiendo del conjunto de datos de entrenamiento puede ser visto como un profesor supervisando el proceso de aprendizaje. Se conocen las respuestas correctas, el algoritmo hace predicciones de forma iterativa sobre los datos de entrenamiento y es corregido por el profesor. El aprendizaje se detiene cuando el algoritmo alcanza un nivel aceptable de rendimiento, en otras palabras, cuando los errores se vuelven insignificantes [55, 56].

Los problemas del aprendizaje supervisado se pueden dividir aún más en problemas de regresión y problemas de clasificación.

- **Clasificación:** Un problema de clasificación ocurre cuando la variable de salida es una categoría, como “rojo” o “azul”, ó “enfermedad” y “sin enfermedad”. A veces estas categorías se representan mediante números, pero su valor no tiene significado propio. Son simplemente etiquetas.
- **Regresión:** Un problema de regresión ocurre cuando la variable de salida es un número real, como “dólares” o “peso”.

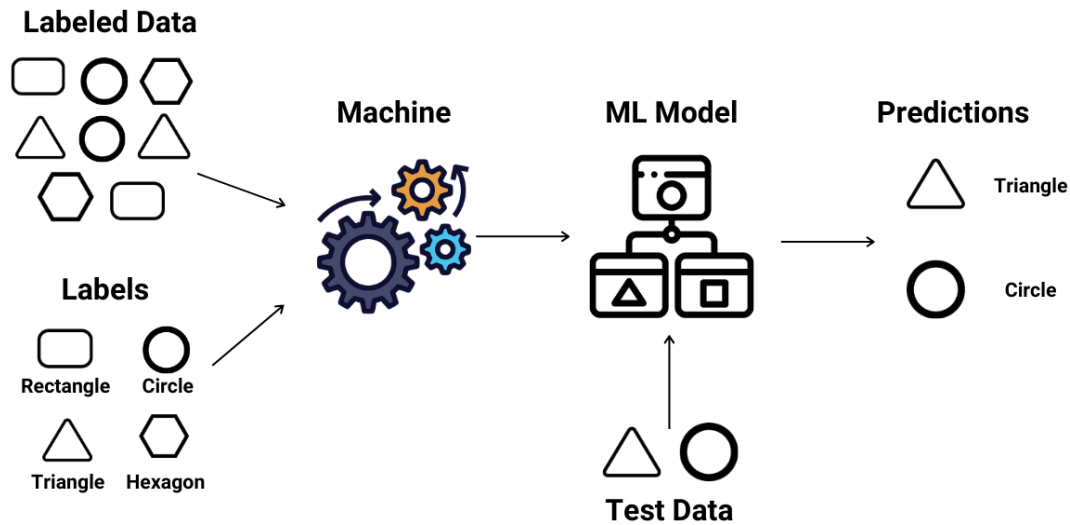


Fig. 2.16: Aprendizaje supervisado. Adaptado de [56].

2.5.4.2 Aprendizaje No Supervisado y Semi-Supervisado

El **aprendizaje no supervisado** (*unsupervised learning*) se utiliza cuando solo se tienen datos de entrada (x) y se desconocen las variables de salida y la función de asignación correspondiente. El objetivo es modelar la estructura subyacente o la distribución de los datos para obtener mayor conocimiento sobre ellos. Este tipo de aprendizaje no tiene respuestas correctas ni supervisión, y los algoritmos encuentran similitudes entre los datos de entrada según un índice de similitud, como la distancia entre dos muestras. Los problemas de aprendizaje no supervisado incluyen la **agrupación** (*clustering*), que busca descubrir grupos inherentes en los datos; la **reducción de dimensionalidad**, que convierte conjuntos de datos de dimensiones altas a dimensiones más pequeñas manteniendo la información esencial; y la **asociación**, que descubre reglas que describen grandes partes de los datos.

El **aprendizaje semi-supervisado** (*semi-supervised learning*) se utiliza cuando se tiene una gran cantidad de datos de entrada (x) y solo algunos están etiquetados (Y), situándose entre el aprendizaje supervisado y el no supervisado. Es útil en situaciones donde el etiquetado de datos es costoso o requiere mucho tiempo, mientras que los datos no etiquetados son baratos y fáciles de recopilar. Un buen ejemplo es un archivo fotográfico donde solo algunas imágenes están etiquetadas (por ejemplo, perro, gato, persona) y la mayoría no tienen etiquetas. Este método es común en problemas reales de aprendizaje automático debido a los costos y tiempos asociados con el etiquetado manual de datos.

2.5.5 Aumento de Datos

El aumento de datos es un proceso que genera artificialmente nuevos datos a partir de los datos existentes, principalmente para entrenar modelos de ML. Dado que obtener conjuntos de datos suficientemente diversos del mundo real puede ser complicado, el aumento de datos crea variaciones a partir de los datos originales para enriquecer el conjunto de datos de entrenamiento. Las soluciones de IA generativa se emplean actualmente para realizar este proceso de manera rápida y eficiente [57].

El aumento de datos es crucial para el entrenamiento de modelos de DL, ya que proporciona las siguientes mejoras:

1. **Rendimiento Mejorado:** Al aumentar el conjunto de datos con diversas variaciones, el modelo puede identificar características más variadas, generalizar mejor y mejorar su rendimiento en entornos reales.
2. **Reducción de la Dependencia de los Datos:** Optimiza la eficacia de conjuntos de datos más pequeños, disminuyendo la necesidad de recopilar grandes volúmenes de datos.
3. **Mitigación del Sobreajuste:** Evita el sobreajuste (*overfitting*) proporcionando un conjunto de datos más amplio y variado para el entrenamiento, lo que ayuda a las redes neuronales profundas a no adaptarse exclusivamente a características específicas del conjunto de entrenamiento.

Las técnicas de aumento de datos que utilizamos son las siguientes:

- **Volteo 3D:** Voltea la imagen 3D respecto a uno de los tres ejes (X, Y o Z) seleccionados aleatoriamente.
- **Transpose 3D:** Transpone la imagen 3D respecto a uno de los tres ejes seleccionados aleatoriamente.
- **Brillo:** Modifica el brillo de la imagen utilizando una transformación gamma de ley de potencia (*power-law gamma transformation*) con parámetros aleatorios.
- **Rotación y Zoom 3D:** Rota la imagen 3D en un ángulo α respecto al eje Z, α varía aleatoriamente entre 0 y 45 grados. Además, aplica un zoom con factores de escala aleatorios entre 0.8 y 1.2.
- **Zoom 3D:** Aplica un zoom con factores de escala aleatorios entre 0.8 y 1.2, en X, Y y Z.

- **Deformación Elástica:** Aplica una deformación elástica a la imagen y su objetivo utilizando un método de deformación de cuadrícula aleatoria.
- **Desenfoque:** Aplica un desenfoque gaussiano a la imagen con un sigma seleccionado aleatoriamente.
- **Añadir Ruido:** Añade ruido de tipo Poisson a la imagen.

Además, cada técnica de aumento de datos se aplica con una probabilidad del 75 % para asegurar que al menos el 25 % de los datos permanezcan originales.

2.5.6 Deep Learning para segmentación de núcleos

2.5.6.1 Segmentación de imagen

La segmentación de imágenes es el proceso de dividir una imagen digital en múltiples regiones basadas en las diferentes propiedades de los píxeles o vóxeles para extraer información significativa que facilite su análisis [58]. Los métodos de segmentación tradicionales a menudo dependen de características seleccionadas manualmente, donde los investigadores o usuarios de dichas herramientas necesitan estar activamente involucrados en la selección de esas características, es decir, cada método necesita ser ajustado específicamente para cada experimento. En contraste, los métodos basados en aprendizaje profundo requieren parámetros de entrada mínimos del usuario y no requieren ajuste fino entre experimentos, lo que los hace más sencillos de aplicar que los enfoques clásicos [59]. Además, se ha demostrado que las redes neuronales profundas son capaces de aprender una jerarquía de características cada vez más complejas directamente de datos en el dominio y realizar la clasificación de píxeles asignando a cada píxel de la imagen una clase, pudiendo distinguir objetos del fondo o un objeto de otro del mismo tipo [60]. Estas dos formas diferentes de clasificar píxeles en una imagen se denominan segmentación semántica y segmentación de instancias.

Segmentación semántica: Se llama segmentación semántica cuando cada píxel se etiqueta como perteneciente a una clase, por ejemplo, distinguiendo entre el fondo y el núcleo de las células, pero sin hacer ninguna distinción entre los núcleos. En este caso, se asignará una etiqueta al fondo, generalmente cero, y otra a todos los núcleos, generalmente uno. Cuando nuestra red neuronal realice la segmentación, asignará las mismas etiquetas a los píxeles pertenecientes al fondo y a los núcleos, respectivamente, dando

como resultado una imagen con el mismo número de etiquetas que las imágenes de verdad fundamental Fig. 2.17a.

Segmentación de instancias: La segmentación de instancias va un paso más allá de la segmentación semántica. En este caso, en lugar de asignar la misma etiqueta a todos los píxeles que pertenecen al mismo tipo de objeto, cada objeto de la misma clase recibirá una etiqueta diferente. Por lo tanto, cada instancia de clase en la imagen tiene una ID única, y la imagen resultante será una imagen donde los límites de los píxeles separan todos los objetos. Como resultado, habrá una clara distinción entre qué píxeles pertenecen a un objeto y cuáles pertenecen al fondo de la imagen, pero también habrá límites claros entre los objetos Fig. 2.17b.

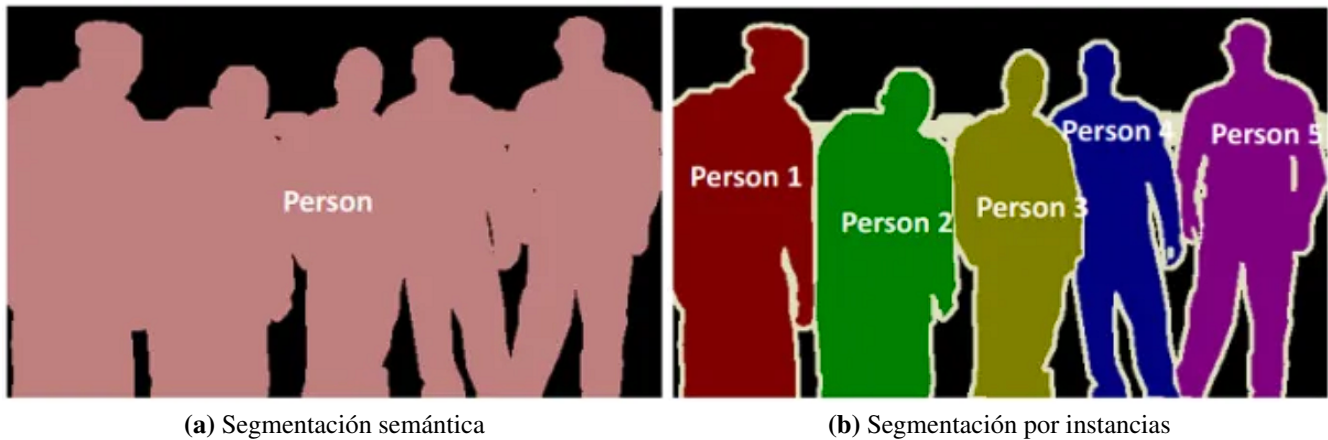


Fig. 2.17: (a) Diagrama de segmentación semántica. (b) Segmentación por instancias. La segmentación semántica trata múltiples objetos dentro de una sola categoría como una entidad, mientras que la segmentación por instancias identifica objetos individuales dentro de estas categorías. Adaptado de [61].

Algunas de las redes neuronales convolucionales más conocidas utilizadas para la segmentación de núcleos son Mask-RCNN [62], U-Net [20], StarDist [1, 2] y Cellpose [3]. Durante el desarrollo de este trabajo, me centré en utilizar modelos del estado del arte como Cellpose, U-Net 3D y Stardist 3D, y desarrollar una segmentación de instancias de núcleos de microscopía de fluorescencia para poder compararlos con el GT generado que se describirá mas adelante. Además se desarrolló un modelo basado en U-Net 3D utilizando una metodología similar a [5, 4].

2.5.6.2 Algoritmos de deep learning para segmentación de núcleos

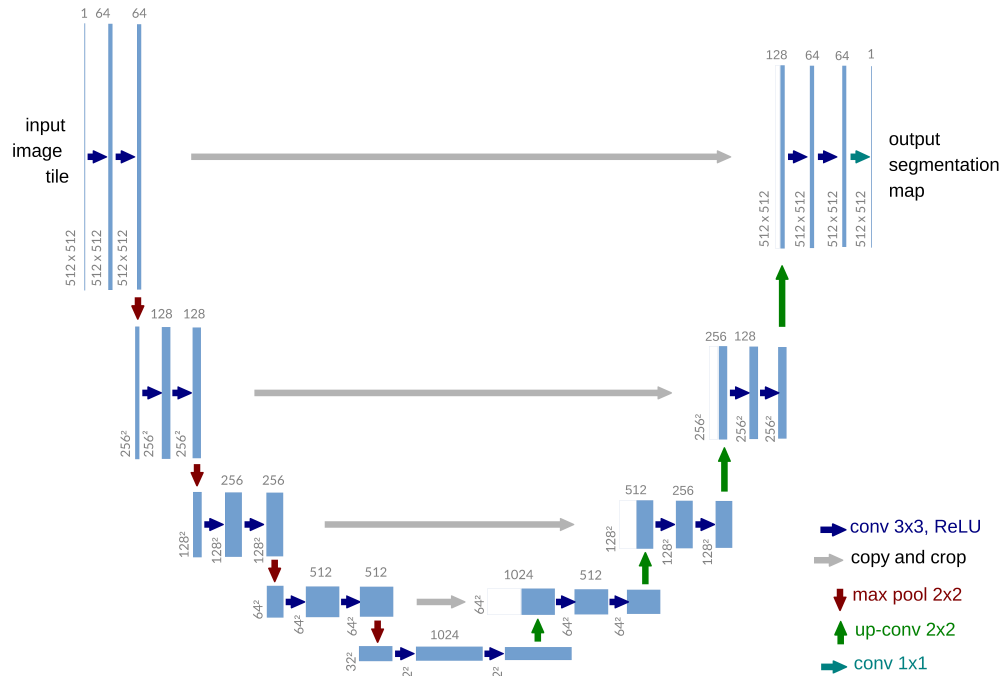


Fig. 2.18: Arquitectura U-Net. Cada caja azul corresponde a un mapa de características multicanal. El número de canales se indica en la parte superior de la caja. El tamaño en x-y se proporciona en el borde inferior izquierdo de la caja. Las cajas blancas representan mapas de características copiados. Las flechas denotan las diferentes operaciones. Adaptado de [20].

U-Net: U-Net es una de las CNN más utilizadas para realizar segmentación de imágenes y se enfoca en segmentar imágenes de microscopía biomédica y biológica. U-Net se ha aplicado a tareas generales de clasificación de píxeles en imágenes 2D y pilas de imágenes 3D con uno o múltiples canales. Además, U-Net se ha adaptado para detectar y segmentar estructuras arbitrarias en tejido biológico utilizando los datos de entrenamiento correspondientes con resultados de segmentación precisos, demostrando su capacidad para superar a algoritmos de segmentación anteriores. Algunas de las tareas donde U-Net se ha aplicado con éxito incluyen la predicción de un único punto de referencia por célula, segmentación de células, segmentación de imágenes médicas y segmentación de núcleos celulares [63, 64].

U-Net toma su nombre de la forma de su arquitectura (Fig. 2.18), que consiste en un camino de contracción, el codificador, y un camino expansivo, el decodificador. El camino de contracción, lado izquierdo de Fig. 2.18, sigue la arquitectura de una red neuronal convolucional tradicional. El codificador U-Net consiste en la repetida aplicación de convoluciones con relleno con ceros, seguidas por una unidad lineal rectificadora (ReLU) y una operación de agrupación máxima de 2×2 con paso de 2 para el submuestreo. Cada uno de estos pasos de submuestreo duplicará el número de canales de características o filtros. Por otro lado, el camino expansivo, lado derecho de Fig. 2.18, consiste en un sobremuestreo (*upsampling*)

del mapa de características seguido de una convolución de 2×2 (*up-convolution*) que reduce a la mitad el número de canales de características, al mismo tiempo que se realiza una concatenación con el mapa de características correspondiente del camino de contracción. Esta concatenación es seguida por dos convoluciones de 3×3 , cada una seguida por una ReLU. El resultado pasa a la siguiente capa del decodificador, y el proceso se repite. En la capa final, se utiliza una convolución de 1×1 para mapear cada vector de características de 64 componentes al número deseado de clases. El camino de submuestreo ayuda a U-Net a propagar información de contexto a capas de mayor resolución combinando características de alta resolución de la decodificación y la salida de las capas del codificador, creando un mapa de segmentación de alta resolución. La red neuronal U-Net es un algoritmo de aprendizaje profundo supervisado. También se ha demostrado que, cuando se implementan aumentos básicos de datos en el conjunto de datos, U-Net se puede utilizar con muy pocas imágenes y aún así tener una precisión muy alta [20].

Los resultados de U-Net pueden extenderse de la segmentación semántica a la segmentación de instancias con algún post-procesamiento o usando imágenes anotadas con tres clases (fondo, bordes y núcleos), en lugar de dos (fondo y núcleos). Aunque U-Net ya ofrece buenos resultados, se han publicado varios artículos en los que U-Net es la base de nuevos algoritmos de segmentación. Algunas de las arquitecturas desarrolladas utilizando U-Net como columna vertebral son 3D U-Net [65], U-Net++ [66], Cellpose [3] y StarDist2D y 3D [1, 2], entre otros. La mayoría de estas nuevas arquitecturas se utilizan para analizar y segmentar imágenes biomédicas y biológicas como U-Net [64].

3D U-Net es una ampliación del marco básico de U-Net que permite la segmentación volumétrica en 3D, donde todas las operaciones 2D se reemplazan por operaciones 3D [65]. Además, algunos modelos más recientes han incorporado bloques residuales y puertas de atención, respectivamente, en la arquitectura U-Net para mejorar su rendimiento. Los bloques residuales, permiten el flujo de gradiente a través de saltos de identidad, han demostrado ser efectivos en la construcción de redes más profundas y estables. Las puertas de atención, por otro lado, permiten que la red preste atención a regiones específicas de la imagen, mejorando la segmentación en áreas complejas o con bajo contraste. Estas mejoras han llevado al desarrollo de modelos como Residual U-Net [67] y Attention U-Net [68], o una combinación de ambos, Residual Attention U-Net [15], que han mostrado resultados superiores en diversas tareas de segmentación de imágenes biomédicas.

Stardist 2D y Stardist 3D: StarDist 2D y StarDist 3D son métodos de segmentación de instancias basados en U-Net. Predicen la forma del núcleo utilizando polígonos convexos estrellados y poliedros convexos estrellados para objetos 2D y 3D [1, 2], respectivamente, Fig. 2.20a. Estos métodos permiten predecir la representación de la forma del núcleo sin necesidad de refinamiento adicional. StarDist 2D predice un polígono convexo estrellado para cada píxel y utiliza supresión de no máximos (NMS) para

generar el conjunto final de polígonos [1]. StarDist 3D, por otro lado, describe la forma de los núcleos utilizando un poliedro estrellado y genera un vector escalado anisotrópicamente para considerar la anisotropía de las imágenes [2], Fig. 2.19. La superficie de los poliedros estrellados se obtiene a partir de sus vértices y el polígono convexo final se determina durante el proceso de NMS.

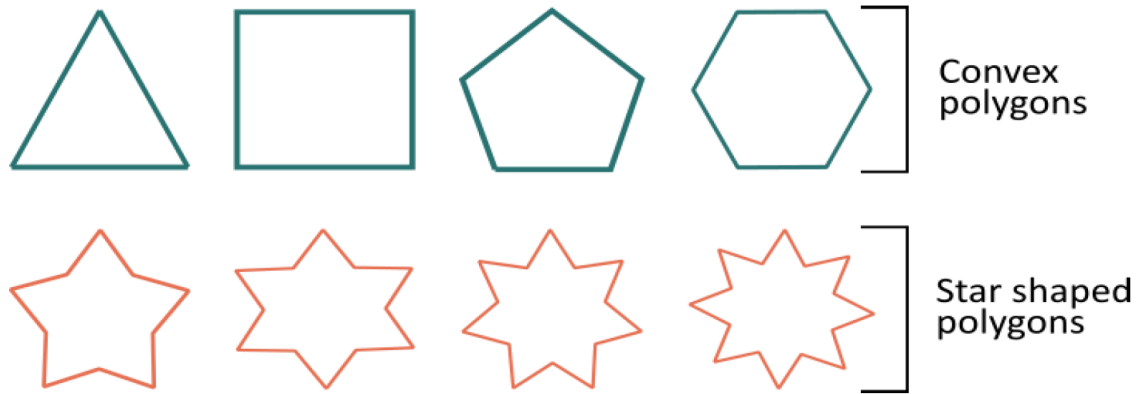


Fig. 2.19: Ejemplos de polígonos convexos y estrellados. Adaptado de [69].

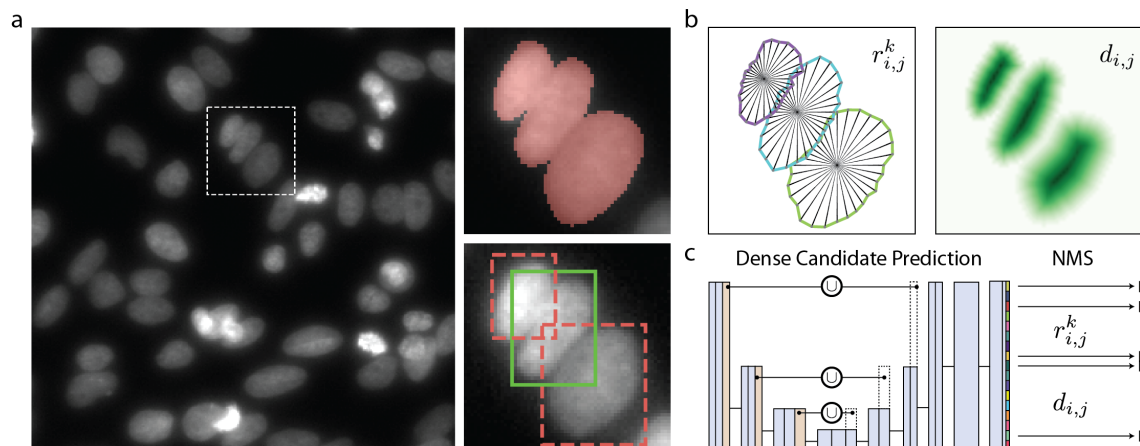


Fig. 2.20: Arquitectura de StarDist 2D. (a) Comparación entre la clasificación de píxeles y la detección de objetos utilizando cajas delimitadoras. (b) Metodología de segmentación de StarDist. (c) Red neuronal de StarDist. Adaptado de [1].

Cellpose: Cellpose [3] es una herramienta de segmentación de instancias de extremo a extremo, es decir, realiza todas las etapas del proceso de segmentación de principio a fin. Recibe una imagen cruda como entrada y produce la segmentación final de las instancias de los objetos. Además, Cellpose se especializa en la segmentación de células y núcleos [3]. Aunque su diseño está fundamentalmente pensado para imágenes 2D, puede extenderse a 3D [3].

Cellpose consta de dos partes principales: primero, se realiza una segmentación semántica y una

regresión de imágenes. La segmentación predice una probabilidad de célula, mientras que la regresión predice un campo vectorial sobre la imagen, como un “mapa de flujos” que guía los píxeles hacia sus respectivos centros, Fig. 2.21a-c. Luego, la segmentación y el campo vectorial se pasan a un segundo paso que realiza la segmentación de instancias final, Fig. 2.21e,f.

Para su funcionamiento, Cellpose requiere que las imágenes de entrada sean 2D. Esto significa que las imágenes 3D se dividen en rodajas 2D antes de ser preprocesadas y pasadas a través de un U-net Residual [70]. Cellpose utiliza la arquitectura U-net como su red base, pero con algunas alteraciones, Fig. 2.21d. Primero, los enlaces entre las rutas de contracción y expansión no se concatenan, sino que simplemente se suman para reducir el número de parámetros. En segundo lugar, se realiza un promedio global de agrupamiento (como una agrupación máxima pero tomando el promedio de los elementos en vez del valor máximo) en la salida de convolución del nivel más bajo de la ruta de contracción, creando así un vector de “estilo” que se proyecta linealmente y se concatena a cada nivel de la ruta de expansión, Fig. 2.21d. Cellpose también utiliza el ensamblaje de modelos, entrenando cuatro modelos idénticos a la vez y promediando sus resultados. Además, realiza aumentaciones en tiempo de prueba en forma de redimensionamiento, procesando múltiples aumentaciones de los datos y promediando los resultados predichos. Finalmente, Cellpose realiza una estimación de la calidad de la región de interés de la segmentación de instancias para eliminar aquellas que se consideran malas.

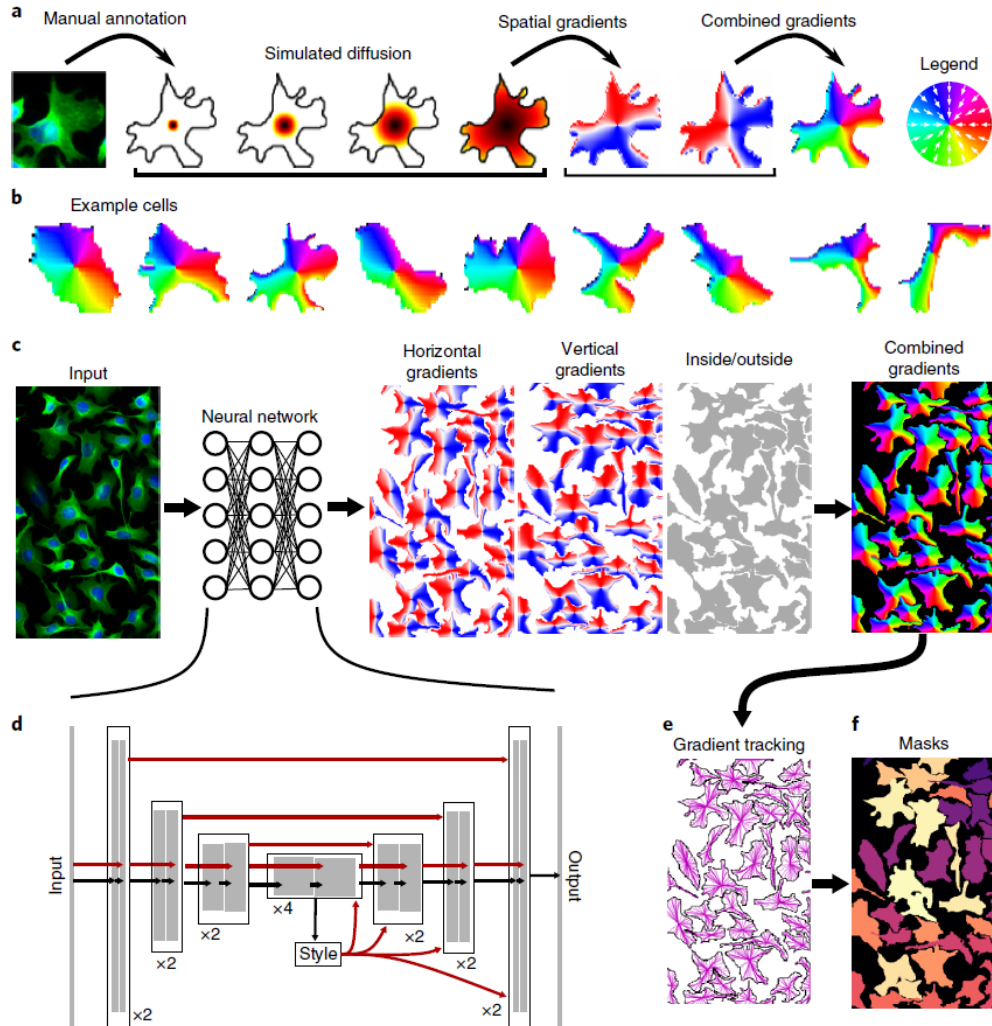


Fig. 2.21: Arquitectura de Cellpose. **(a)** Procedimiento para transformar máscaras anotadas manualmente en una representación de flujo vectorial que puede ser predicha por una red neuronal. **(b)** Ejemplo de gradientes espaciales para células del conjunto de datos de entrenamiento. **(c)** Se entrena una red neuronal para predecir los gradientes horizontales y verticales, también una máscara binaria. Los 3 mapas predichos se combinan en un campo vectorial de gradientes. **(d)** Detalles de la red neuronal que contiene una columna vertebral estándar U-Net. **(e)** Durante la predicción, los campos vectoriales predichos construyen un sistema dinámico donde cada píxel sigue los gradientes hacia un punto fijo. **(f)** Los píxeles que convergen al mismo punto fijo forman una máscara. Adaptado de [3].

2.6 Algoritmo de Descenso de Gradiente

Dado un gradiente 3D predicho, $\nabla\Phi$, se aplica el descenso de gradiente al conjunto de todas las coordenadas de la imagen en primer plano (*foreground*), $\{x_n, y_n, z_n\}$. La ecuación iterativa de actualización para el descenso de gradiente 3D con momento para el número de iteración, $t = 0, \dots, T$, donde $T = 50$

es el número total de iteraciones implementadas en la estimación del centroide, ver Fig. 2.6.1.

$$(x_n^t, y_n^t, z_n^t) \leftarrow (x_n^{t-1}, y_n^{t-1}, z_n^{t-1}) - \eta \cdot \nabla \Phi(x_n^{t-1}, y_n^{t-1}, z_n^{t-1}) + \mu \cdot \nabla \Phi(x_n^{t-2}, y_n^{t-2}, z_n^{t-2}) \quad (2.6.1)$$

Donde en Ec. 2.6.1 $\nabla \Phi$ es el mapa de gradientes 3D, η es tamaño de paso (por defecto $\eta = 0.05$), μ es el parámetro de momento que controla la influencia del gradiente de la iteración anterior, con un rango de 0-1 (por defecto $\mu = 0.8$). Se utiliza la interpolación lineal entre los píxeles del mapa de gradiente, de modo que (x_n^t, y_n^t, z_n^t) siempre existe.

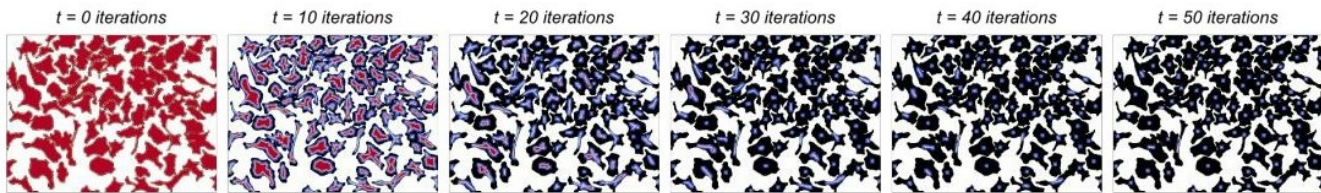


Fig. 2.22: Ejemplo de la utilización del algoritmo de descenso de gradiente para estimar los centroides de cada núcleo. Adaptado de [4].

2.7 Discusión

El desarrollo de nuevas técnicas y tecnologías ha permitido avances significativos en el campo de la biología, particularmente en la comprensión de la estructura y función de los órganos a nivel celular. Una de las herramientas más importantes en la investigación biológica es la microscopía de fluorescencia, la cual permite visualizar y analizar la distribución de moléculas específicas dentro de las células y tejidos. En los últimos años, el deep learning ha emergido como una poderosa herramienta para el análisis de imágenes biológicas. Las redes neuronales convolucionales, en particular, han demostrado ser altamente efectivas para la segmentación de imágenes y la detección de estructuras celulares. Esta técnica podría ser útil para analizar y estudiar el hígado, debido a su complejidad y difícil tarea de abordar con métodos tradicionales, como las herramientas que utiliza Motion Tracking. Este capítulo revisa el uso de técnicas de deep learning en la investigación biológica, con un enfoque en la segmentación de núcleos celulares en imágenes de microscopía de fluorescencia.

Capítulo 3. Generación de base de datos

3.1 Introducción

En esta sección se presenta todo el proceso realizado por el laboratorio para la adquisición y procesamiento de las imágenes. Además se presenta el flujo de trabajo realizados para la segmentación manual de lo núcleos 3D. Por último se presentan los resultados de la segmentación para 3 sujetos de control.

3.2 Generación de base de datos

La segmentación de núcleos es una tarea importante en el análisis celular que requiere métodos de segmentación precisos y confiables. Uno de sus mayores desafíos es segmentar núcleos celulares densamente empaquetados cuando muchos núcleos están tocándose entre sí en las direcciones X, Y y Z. Desafortunadamente, este paso sigue siendo un cuello de botella en la actualidad, principalmente debido a su complejidad y a la diferencia de velocidad entre la generación y el análisis de datos. De hecho, los métodos basados en modelos luchan tanto en el análisis de datos muy complejos como en la generalización para diferentes condiciones. Por el contrario, los enfoques de aprendizaje profundo, y especialmente los supervisados, pueden tener éxito en ambos ámbitos a expensas de contar con suficientes datos para entrenar. Como resultado, la disponibilidad de datos de verdad fundamental (GT) se ha convertido en el principal cuello de botella para los métodos basados en datos [13, 14]. Un conjunto de datos de GT de alta calidad es crucial para el éxito de cualquier algoritmo de aprendizaje profundo supervisado. En consecuencia, su creación es un proceso que consume mucho tiempo, es desafiante y propenso a errores porque depende de los esfuerzos humanos.

3.2.1 Adquisición y Procesamiento de las Imágenes

- **Muestras para Base de Datos** Para generar la base de datos de imágenes, se obtuvieron ratones C57BL/6J de día 30 postnatal y adultos de 8 y 12 semanas (C1, C2 y C3) del Centro Regional de Estudios Avanzados para la Vida (CREAV) de la Universidad de Concepción. Los animales se mantuvieron en estrictas condiciones libres de patógenos y recibieron alimentación *ad libitum*. Todos los procedimientos realizados fueron aprobados por la Comité de Ética y de Bioseguridad

de Investigación y Vicerrectoría de Desarrollo de la Universidad de Concepción. Además, este estudio se realizó de acuerdo con las directrices (ARRIVE), y todos los métodos se llevaron a cabo de conformidad con las directrices y reglamentos pertinentes.

- **Recolección de Muestras e Inmunotinción:** Los hígados de los ratones fueron fijados mediante perfusión intracardiaca con paraformaldehído al 4 % Tween-20/PBS al 0,1 % y post-fijados durante la noche con la misma solución a temperatura ambiente. Se obtuvieron secciones de hígado de 100 μm de grosor con un vibratomo. El inmunomarcaje y la limpieza óptica se realizaron como se describen en [9].
- **Imágenes:** Se tomaron imágenes de las muestras de hígado, con un tamaño de vóxel de 0.3 μm , en un microscopio multifotónico invertido de barrido láser (Zeiss LSM 780) utilizando un objetivo de multi-imersión de 40x1,2 de apertura numérica (Zeiss). El DAPI se excitó a 780 nm utilizando un láser de 2 fotones Chameleon Ti-Sapphire. Alexa Fluor 488, 568 y 647 se excitaron con líneas láser 488, 561 y 633. La señal emitida se detectó con detectores de fosforo de arseniuro de galio (GaAsp).
- **Pre-procesamiento de imágenes:** Los distintos componentes de los tejidos hepáticos (canalículos biliares - BC, sinusoides, núcleos, células estrelladas - HSC, células de Kupffer - KC y hepatocitos) se reconstruyeron a partir de pilas de imágenes fluorescentes de alta resolución, tamaño de vóxel 0.3 x 0.3 x 0.3 μm ($\approx 100 \mu\text{m}$ de profundidad). Para cubrir la totalidad de los ejes vena central - vena porta (CV-PV), se unieron mosaicos de 2x1 utilizando el complemento de unión de imágenes de Fiji [71]. Brevemente, para el pre-procesamiento de las imágenes 3D se utilizó primero el algoritmo PURE-LET [72] con el máximo número de ciclos. A continuación, se realizó una corrección del fondo y del sombreado utilizando la herramienta BaSiC [73] a lo largo de la pila. Y todos los canales se alinearon con uno de referencia utilizando la función Correct 3D Drift de Fiji.

3.2.2 Segmentación manual

La anotación manual solo requiere 3 componentes: (1) una imagen para anotar, (2) un software que pueda abrir la imagen, manejar la creación y transferencia de regiones de interés (ROIs) a otra imagen, y (3) un usuario que delinearé los diferentes objetos con ROIs. Como quería anotar núcleos en 3D, una mejora obvia para evitar una anotación completamente manual habría sido utilizar algún método tradicional para generar segmentaciones por instancia en 2D y reconectarlas en 3D ó utilizar algún software o algoritmo en 2D para generar una anotación semi-automática. Utilicé softwares como Quepath [74] o Ilastik [75] para generar segmentaciones 2D y reconectarlas en 3D.

El proceso de unir la pila de máscaras 2D requería mucho tiempo, ya que era necesario corregir manualmente las segmentaciones incorrectas, incluyendo casos de sobresegmentación y subsegmentación. Además, el seguimiento de cada núcleo dentro de la imagen resultaba complicado. Era esencial revisar las tres vistas ortogonales del núcleo (XY, XZ, YZ) para asegurarse de que la segmentación estuviera correcta.

Debido a estas dificultades, se optó por realizar una segmentación completamente manual, núcleo por núcleo, complementada con una herramienta de interpolación entre capas. Esto permitía segmentar solo algunas de las capas del núcleo, sin necesidad de segmentarlas todas.

3.2.2.1 Napari

La segmentación manual se realizó a través del software de Napari [17]. Si bien ya existían softwares en el laboratorio para generar anotaciones como Matlab [76] o Fiji [36], estos están diseñados para la segmentación 2D y en general se vuelve complicado el manejo de errores o retomar la segmentación de núcleos. Además en el caso de Matlab, después de segmentar muchos núcleos, el software se vuelve mas lento y pesado. Es por esto que se decidió utilizar Napari, ya que este puede leer imágenes en formato .tif (formato de las imágenes) y admite varios tipos de capas, entre las que destacamos las capas de imágenes y de etiquetas para realizar nuestra segmentación. También admite datos de tipo n-dimensiones, por lo que se pueden visualizar imágenes 2D y 3D con facilidad en todas sus vistas ortogonales, Fig. 3.23. Otro punto destacado es la comunicación bidireccional entre el visor y el kernel de Python, lo que resulta muy útil cuando se trabaja con imágenes en un cuaderno en Jupyter y se quieren visualizar o guardar.

Por último Napari, al ser un software abierto a la comunidad, posee una gran versatilidad debido a la

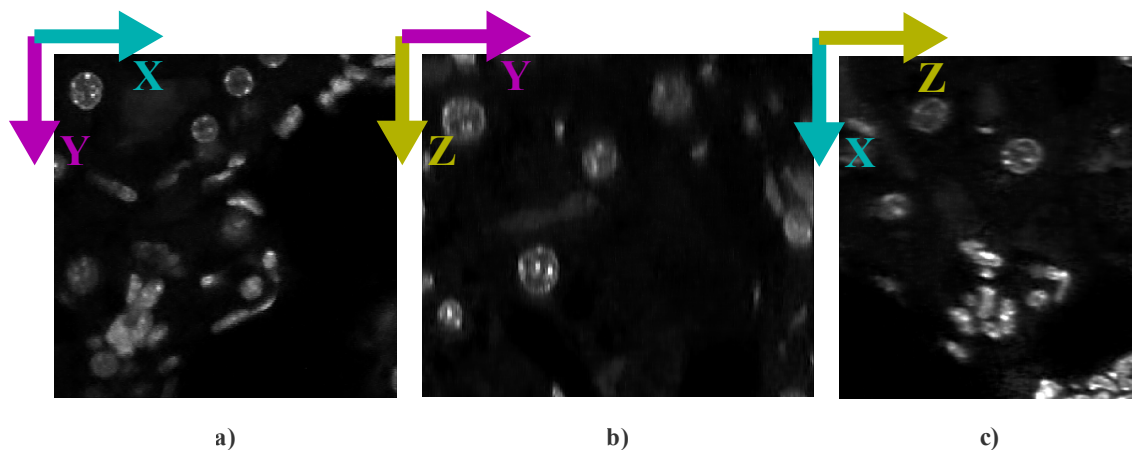


Fig. 3.23: Vistas ortogonales de una muestra núcleos hepáticos. (a) Plano YX. (b) Plano ZY. (c) Plano XZ.

existencia de muchos complementos (*plugins*) creados por esta misma comunidad con diversas funcio-

nalidades. Uno de ellos es el complemento Annotation Toolbox de Napari [77] que tiene la herramienta de *Slice Interpolator* 3.24b, la cual utilizamos para realizar la interpolación requerida para no segmentar completamente los núcleos con el fin de disminuir el tiempo que demora en segmentar un núcleo 3D. Específicamente, este complemento utiliza el método de Posición Reescalada por Velocidad Cuadrada (RPSV), que es un método un poco más sofisticado y robusto para la interpolación en comparación con métodos basados en distancia o en el contorno, utilizamos los parámetros predeterminados (300 puntos de contorno y un máx. de 20 iteraciones), Fig. 3.24c.

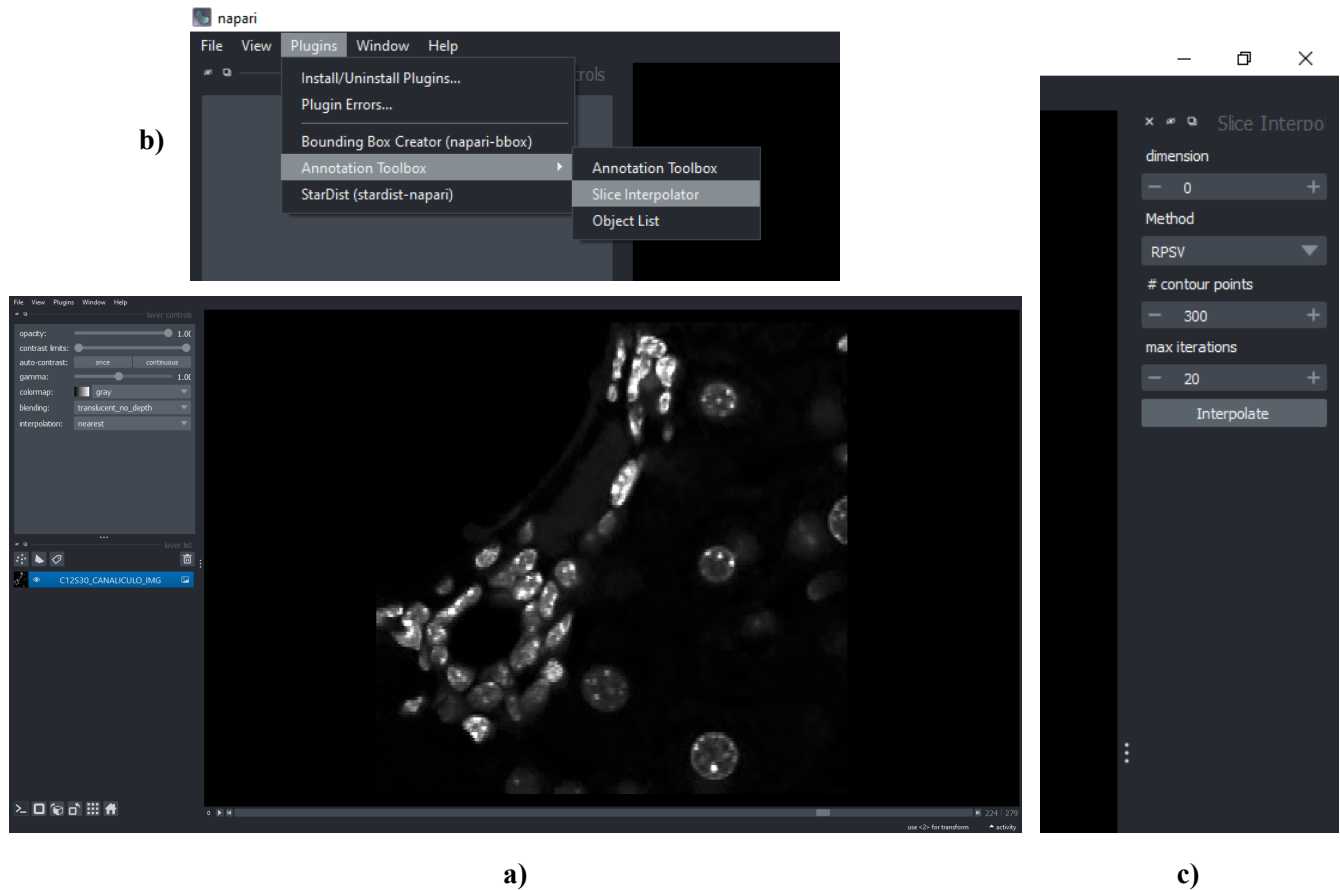


Fig. 3.24: Visualización y plugin de interpolación en Napari. (a) Visualizador de Napari con una muestra de núcleos hepáticos. (b) Plugin de Napari: Annotation-Slice Interpolator. (c) Sección de interpolación con el método RPSV.

3.2.3 Proceso de segmentación

De cada sujeto de control se seleccionaron tres zonas específicas de cada imagen, como se mencionó en 2.2.3, la región periportal, región pericentral y región central del lóbulillo hepático. Cada zona segmentada se delimitó en cubos de $280 \times 320 \times 320$ vóxeles y así obtener información suficiente de la

profundidad de la imagen y la región en la que está. Se segmentó núcleo por núcleo la imagen utilizando las capas de imagen y etiquetas de Napari, mediante el cual se fueron segmentado desde el centro aproximado del núcleo hacia los extremos, considerando un espaciado de entre 3-5 capas aproximadamente y enfocándose un poco más en los límites del núcleo, como se puede ver en Fig. 3.25a-c. Luego utilizando el *Slice Interpolator* con el método RPSV se genera la máscara tridimensional del núcleo, ver Fig. 3.25d.

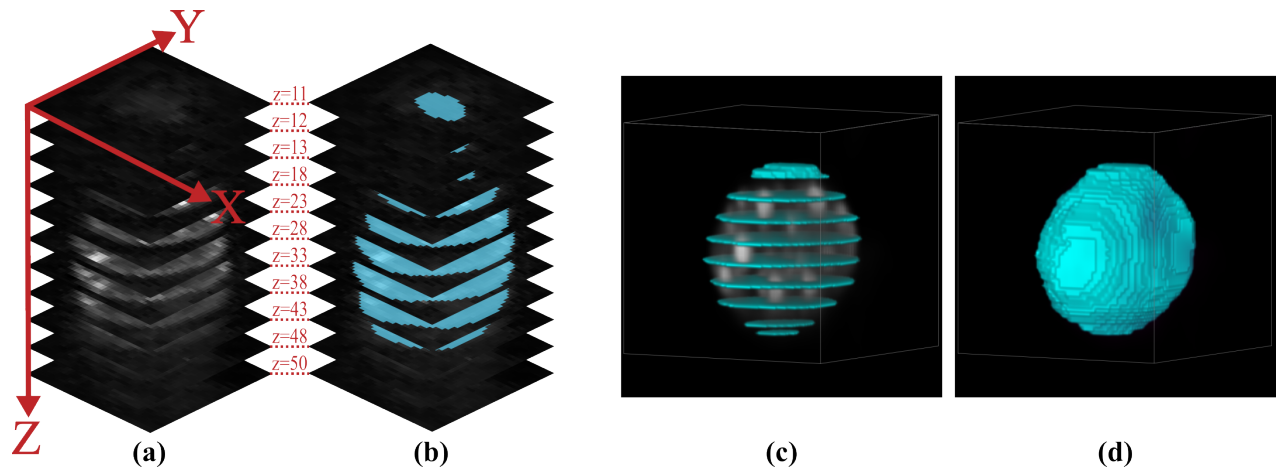


Fig. 3.25: Proceso de interpolación de máscaras de núcleos 3D. **(a)** Imagen del z-stack de un núcleo de hepático. **(b)** Máscara de un z-stack de un núcleo hepático. **(c)** Núcleo 3D con máscara sin interpolación. **(d)** Máscara 3D interpolada.

El proceso de segmentación se realizó en conjunto con estudiantes de doctorado del laboratorio del Dr. Segovia. La segmentación total de los 12 cubos tomó alrededor de un mes con 4 personas trabajando a medio tiempo (3-4 hrs. diarias/ 5 días por semana) y se segmentaron aproximadamente 2900 núcleos 3D. Los detalles se pueden ver en la Tabla 3.1 y Fig. 3.26.

Sujeto	Región Pericentral	Región Central	Región Periportal	Total
C1	261	369	570	1200
C2	207	230	425	862
C3	218	228	375	821
Total	686	827	1370	2883

Tabla 3.1: Cantidad de núcleos en cada sujeto (C1, C2 y C3) por región del lobulillo hepático.

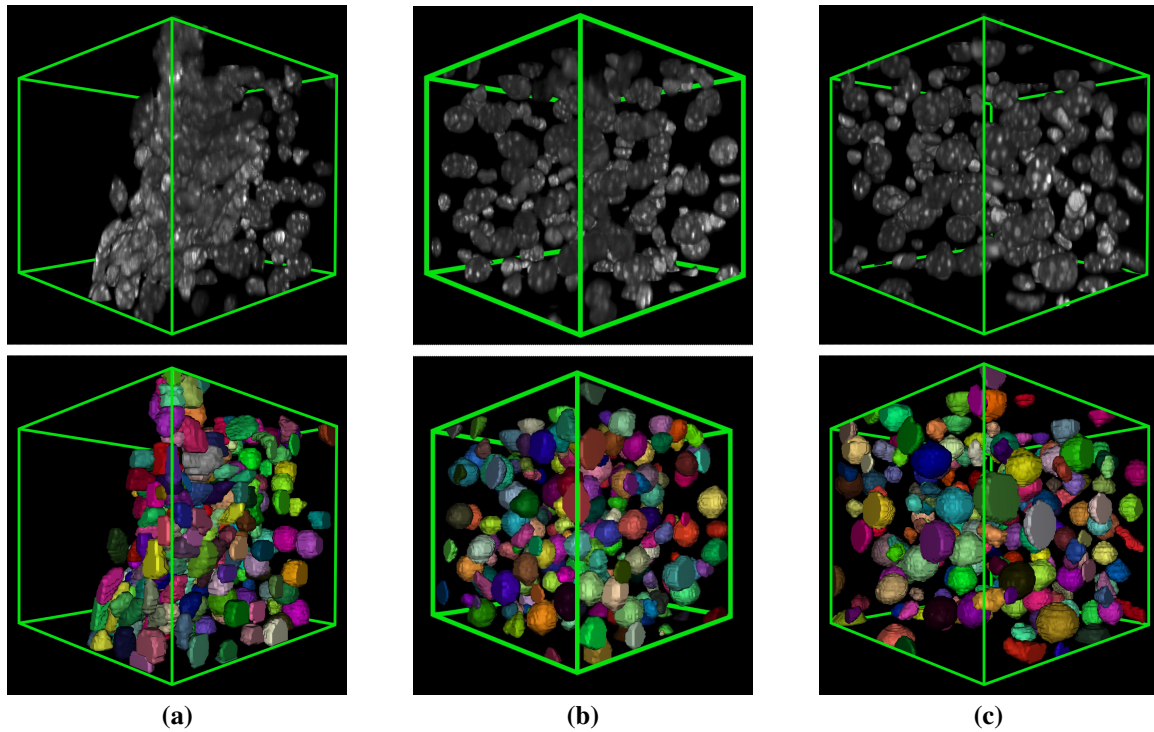


Fig. 3.26: Imágenes y máscaras de núcleos hepáticos de C3. Arriba están las imágenes DAPI de los núcleos y abajo las máscaras manualmente segmentadas. (a) Región periportal. (b) Región central. (c) Región pericentral.

3.3 Discusión

La generación de una base de datos precisa y confiable es esencial para el éxito de los algoritmos de segmentación de núcleos celulares en deep learning. En el laboratorio se realizó un esfuerzo significativo para recopilar y procesar imágenes de alta calidad de hígados de ratones, siguiendo procedimientos estandarizados y rigurosos de inmunotinción y captura de imágenes. Este proceso ha sido crucial para garantizar la precisión y consistencia de las imágenes utilizadas para la segmentación manual y el posterior entrenamiento de modelos de deep learning.

Uno de los mayores desafíos en la segmentación de núcleos 3D en el hígado es la complejidad inherente del órgano. Se encuentran diferentes tipos de núcleos hepáticos con morfologías variadas y una distribución que varía según la región del lobulillo. En áreas específicas como la región periportal, hay una alta densidad de núcleos 3D los cuales se tocan en múltiples direcciones (X, Y y Z), lo que dificulta la segmentación. Esto requirió más tiempo y precisión. Sin embargo, la segmentación manual se complementó con herramientas de interpolación, demostrando ser efectiva para generar datos de referencia de alta calidad.

Capítulo 4. Implementación

4.1 Introducción

En esta sección se detalla la arquitectura del modelo propuesto para la segmentación por instancia en imágenes de microscopía de fluorescencia de tejido hepático con núcleos 3D. Primero, se describe el sistema utilizado para el entrenamiento del modelo, incluyendo las especificaciones del hardware y software. A continuación, se discuten las consideraciones tomadas para la generación del conjunto de datos de entrenamiento, enfatizando los métodos empleados para crear parches adecuados a partir de los datos originales. También se presentan los hiperparámetros del modelo y la configuración de la función de pérdida compuesta, explicando cómo se combinan diferentes componentes de pérdida para optimizar el rendimiento del modelo. Finalmente, se exploran las técnicas de aumentación de datos aplicadas para abordar la insuficiencia de datos, destacando su importancia en la mejora de la robustez y generalización del modelo.

4.2 Arquitectura del modelo

Se utilizó un modelo 3D basado en U-Net, que incorpora bloques residuales (Fig. 4.27b) y puertas de atención (Fig. 4.27c), modificando el bloque convolucional usual (Fig. 4.27a). Dado que estos componentes solo proporcionan una segmentación semántica, para abordar el problema de la segmentación por instancia (diferenciación de núcleos individuales), se añadió un segundo decodificador a la arquitectura Res Attention U-Net, como se ilustra en la Fig. 4.28, siguiendo la metodología descrita en [78]. Además, de manera similar a Cellpose [3], convertimos las etiquetas de los núcleos en una máscara binaria y en un campo de flujo vectorial (Fig. 4.29). Este campo de flujo se descompone en gradientes en las direcciones horizontal, vertical y de profundidad (gradientes en X, Y y Z), lo cual permite estimar el centroide y los bordes de cada núcleo [5].

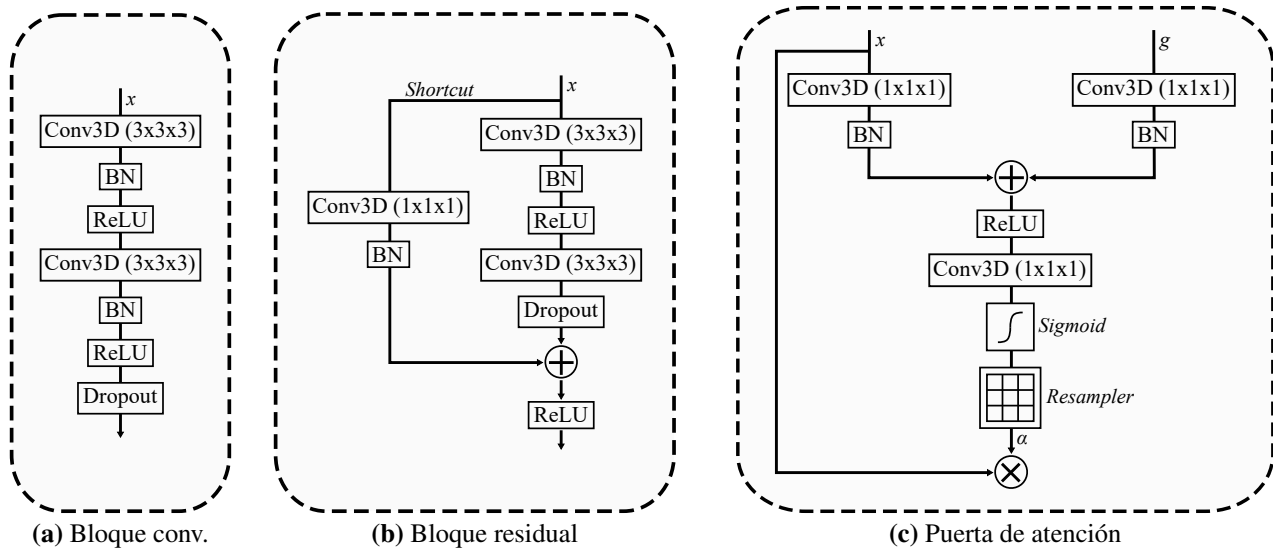


Fig. 4.27: Bloques de arquitectura convolucional. **(a)** Bloque convolucional. **(b)** Bloque residual con conexión directa (*shortcut*). **(c)** Puerta de atención con entrada dual, Sigmoid y Resampler para ajustar la atención sobre la entrada original. Además tenemos capas Conv3D, Batch Normalization (BN), ReLU y Dropout. Adaptado de [5].

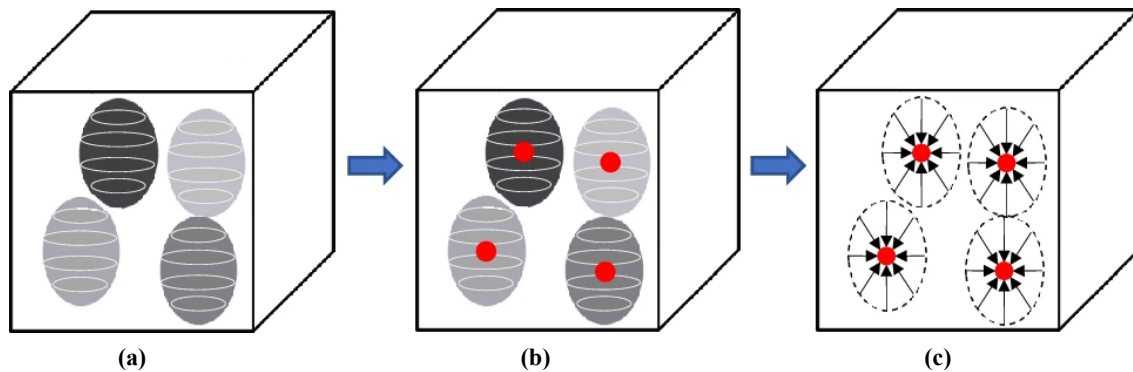


Fig. 4.29: Generación de flujo vectorial (gradiente en X, Y y Z). **(a)** Máscara de etiquetas de los núcleos. **(b)** Centroide de cada etiqueta. **(c)** Flujo vectorial en X, Y y Z. Adaptado de [5].

4.3 Entrenamiento del modelo

Para poder segmentar una imagen a través de un modelo de redes neuronales, es necesario tener varias consideraciones. Lo primero es que uno, no puede segmentar la imagen completa a través del modelo, ya sea por temas de memoria o para poder generalizar más el modelo y poder ocupar en imágenes de distinto tamaño. Así es necesario ajustar un tamaño de entrada en la red, cual puede variar por la memoria que uno tiene disponible ó de acuerdo al tipo de imagen que uno quiere procesar.

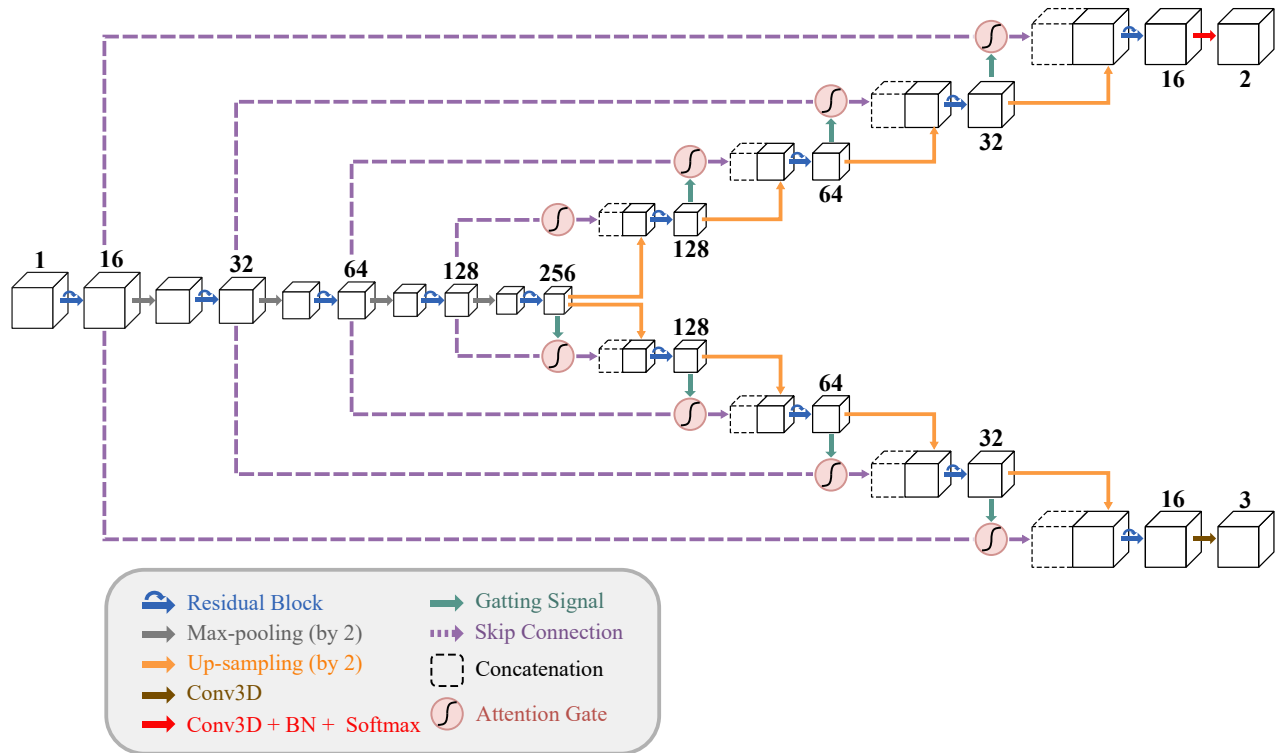


Fig. 4.28: Arquitectura del modelo 3D U-Net con bloques residuales, puertas de atención y 2 decodificadores para una multitarea. Bloques residuales (azul), capa de agrupación máxima (gris) y el sobre-muestreo (naranja). La capa de convolución 3D (café) predice la gradiente 3D de las etiquetas y la capa de convolución 3D, normalización y función softmax (rojo) predice la segmentación binaria. La señal de compuerta (verde) controla las puertas de atención, mientras que las conexiones de salto (líneas punteadas moradas) se concatenan y combinan las características de múltiples escalas. Elaboración propia.

Debido al tamaño de los núcleos, en parches de $64 \times 64 \times 64$ vóxeles apenas se pueden encontrar núcleos completos. Para abordar este problema y evitar obtener muchas imágenes con escasa o nula señal de núcleo, se utilizaron parches de $128 \times 128 \times 128$ vóxeles como entrada para evitar tener muchas imágenes con muy poca o nada de señal de núcleo en ellas.

El modelo fue entrenado en un ordenador con el sistema operativo Windows 11 Pro, utilizando un entorno virtual de Ubuntu 22.04 y TensorFlow 2.13.1 [19]. La configuración del hardware incluye un procesador AMD Ryzen Threadripper 3960X con 24 núcleos a 3.80 GHz, 256 GB de memoria RAM y una tarjeta gráfica NVIDIA GeForce RTX 3090 con 24 GB de RAM.

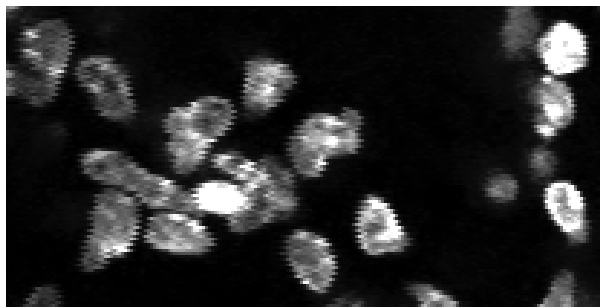
4.3.1 Pre-procesamiento

Se describen a continuación las funciones de pre-procesamiento de imágenes realizadas:

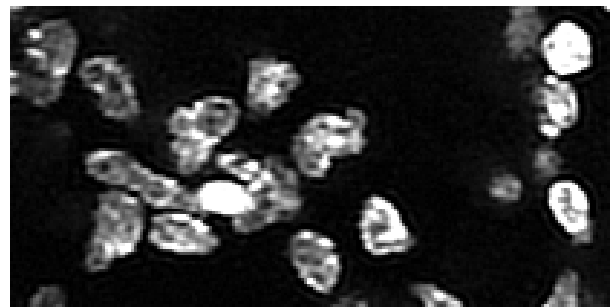
- **Normalización de Intensidad con Mejora de Contraste:** Las intensidades de las imágenes se normalizan de tal manera que 0 se establece en el percentil inferior (generalmente 2 %) y 1 se establece en el percentil superior (generalmente 99.8 %) de la intensidad de la imagen. Esto mejora el contraste de la imagen al recortar las intensidades esporádicas altas causadas por el ruido de la cámara del microscopio y al poner a cero las pequeñas intensidades de fondo no nulas comunes en la microscopía fluorescente.
- **Deconvolución de Imágenes:** Se utiliza la deconvolución ciega con el enfoque Wiener-Hunt no supervisado [79] (corte por corte en 3D), donde los hiperparámetros se estiman automáticamente utilizando un muestreador de Gibbs (implementado en Scikit-image de Python, `skimage.restoration.unsupervised_wiener`). La función de dispersión de puntos (PSF) inicial se especifica como un kernel gaussiano normalizado en suma de 15x15 píxeles ($\sigma = 1$).
- **Corrección de Iluminación Desigual:** La intensidad de la imagen de entrada imágenes 3D con iluminación desigual se corrige corte por corte con la Ec. 4.3.1.

$$I_{\text{correct}} = I_{\text{raw}} \cdot \frac{\overline{I_{\text{raw}}}}{I_{\text{bg}}} \quad (4.3.1)$$

Donde en Ec. 4.3.1, $\overline{I_{\text{raw}}}$ es la intensidad media de la imagen de entrada, I_{bg} es una estimación de la iluminación de fondo desigual, I_{bg} es la imagen de entrada con un suavizado gaussiano de un σ especificado ($\sigma = 20$).



(a) Sin pre-procesado.



(b) Con pre-procesado.

Fig. 4.30: Comparación de imágenes sin y con pre-procesado.

4.3.2 Conjunto de entrenamiento

Para el entrenamiento del modelo, utilizamos únicamente los conjuntos de datos C1 y C2, reservando C3 exclusivamente para la evaluación de los resultados.

Dado que el tamaño de cada cubo (280x320x320 vóxeles) no coincidía con el tamaño de entrada establecido (128x128x128 vóxeles), se utilizó la biblioteca Patchify [80] para generar parches del tamaño adecuado. Para poder dividir las imágenes en parches sin solapamiento, se agregó un relleno con ceros de acuerdo con la Ec. 5.2.1, obteniendo un volumen final de 384x384x384 vóxeles. Los parches se extrajeron con un paso de 128 vóxeles en los ejes X, Y y Z para garantizar que no hubiera solapamiento entre ellos (ver Tabla 4.2).

De cada imagen se extrajeron 27 parches de 128x128x128 vóxeles. Posteriormente, se filtraron aquellos parches que contenían un 10 % o menos de señal de núcleo en comparación con el máximo de señal del conjunto, con el objetivo de evitar parches con poca o ninguna información relevante. Finalmente, el 90 % de los datos se destinó al conjunto de entrenamiento, mientras que el 10 % se reservó para el conjunto de validación.

Nº parche	0	1	2
Inicio	0	128	256
Fin	128	256	384

Tabla 4.2: Inicio y fin de parches de 128 píxeles con un paso de 128 píxeles en los ejes X, Y y Z.

4.3.3 Pérdida, parámetros y entrenamientos del modelo

Se diseñó una función de pérdida basada en la utilizada en [5]. Ya que al entrenar una red con multitarea (predicción de segmentación binaria y de campo vectorial), es necesario una **función de pérdida compuesta** o **función de pérdida combinada** para integrar los distintos objetivos del entrenamiento en una sola pérdida. La matriz de campo vectorial 3D de salida se compara con la matriz de campos vectoriales de verdad fundamental, y el error entre ellos se minimiza utilizando la función de pérdida de error cuadrático medio (MSE). De manera similar, el resultado de la segmentación se compara con el volumen binario de verdad fundamental, y la diferencia se minimiza utilizando una combinación de la pérdida focal (FL) y la pérdida de Tversky (TL). Además a la ponderación de esta función de pérdida compuesta se le agregan coeficientes para ajustar la importancia de cada componente en el proceso de optimización.

$$L = \lambda_1 \cdot TL + \lambda_2 \cdot FL + \lambda_3 \cdot MSE \quad (4.3.2)$$

Donde en la ecuación 4.3.2, L es la pérdida total, TL es la pérdida de Tversky, FL es la pérdida focal y MSE es el error cuadrático medio. Los valores de los parámetros son $\lambda_1 = 1$, $\lambda_2 = 10$ y $\lambda_3 = 10$.

Para entrenar la red se utilizó un optimizador Adam con tasa de aprendizaje (*learning rate*) de 1×10^{-4} , $\beta_1 = 0.9$, $\beta_2 = 0.999$ y $\epsilon = 1 \times 10^{-7}$. La configuración del modelo se puede ver en Tabla 4.3.

Configuración del modelo	
No. de épocas	200
Tamaño de parche	128x128x128
Tamaño de lote	2

Tabla 4.3: Parámetros de configuración del modelo utilizados en el experimento.

Además, el problema de tener datos insuficientes en el entrenamiento se puede aliviar utilizando aumentación de datos. Se aplicaron operaciones de volteo 3D (*Flip 3D*), transposición de ejes, ajuste de brillo, rotación y zoom 3D, deformaciones elásticas, desenfoque gaussiano y adición de ruido con un 75 % de probabilidad de suceder para mantener al menos el 25 % de los datos originales en el conjunto de entrenamiento.

4.4 Discusión

El uso de un modelo 3D basado en U-Net con bloques residuales y puertas de atención ha demostrado ser eficaz para la segmentación semántica de imágenes biomédicas [70, 68]. Sin embargo, para abordar el problema más complejo de la segmentación por instancia, la adición de un segundo decodificador resulta crucial. Esta modificación permite una mejor diferenciación de núcleos individuales, mejorando la precisión y la utilidad de la segmentación en aplicaciones prácticas [3, 5].

La generación de un campo de flujo vectorial a partir de las etiquetas de los núcleos, y su descomposición en gradientes en X, Y y Z, proporciona una información adicional valiosa para poder estimar los centroides y bordes de los núcleos. Esta técnica, inspirada en Cellpose en 2D y recientemente por Nisnet [5] en 3D, mejora la precisión de la segmentación por instancia, permitiendo una caracterización más detallada de las estructuras celulares. La función de pérdida compuesta, que integra la pérdida de Tversky (TL), la pérdida focal (FL) y el error cuadrático medio (MSE), optimiza el entrenamiento del modelo para múltiples tareas. El uso de funciones de pérdida especializadas, como la TL para redes 3D [81] y la FL para el desequilibrio de clases, mejora la capacidad del modelo para manejar diferentes tipos de errores y desafíos durante el entrenamiento.

Además, la aumentación de datos es una estrategia clave para aliviar el problema de datos insuficientes en el entrenamiento. Esto aumenta la diversidad del conjunto de datos de entrenamiento, mejorando la capacidad del modelo para generalizar y desempeñarse bien en imágenes no vistas. En el futuro, se podría considerar la incorporación de redes generativas, como se ha visto en [5], para mejorar aún más la generalización del modelo.

Capítulo 5. Predicción y Resultados

5.1 Introducción

En este capítulo, se aborda el problema de predicción y reconstrucción de grandes volúmenes de imágenes de microscopía utilizando técnicas avanzadas de procesamiento de imágenes y aprendizaje profundo. Se presenta una metodología basada en la subdivisión de la imagen en parches adyacentes con solapamiento, seguida de la reconstrucción mediante interpolación para suavizar las transiciones entre parches. Además, se describe la metodología del post-procesado para estimar los centroides (con el algoritmo de descenso de gradiente) y generar una segmentación por instancias final de los núcleos.

5.2 Desafíos en la Predicción y Reconstrucción de Imágenes Grandes

Al predecir imágenes grandes, se enfrentan dos grandes desafíos:

- **Ajuste del Tamaño de la Entrada:** Dado que nuestra red tiene un tamaño de entrada predefinido, es necesario ajustar la imagen de entrada para que sea divisible por el tamaño esperado de la red. Esto implica que la imagen debe tener una forma específica para asegurar que se pueda dividir correctamente en parches del tamaño de entrada de la red.
- **Reconstrucción de la Imagen Original:** Otro problema no trivial es la reconstrucción de la imagen original a partir de los parches predichos. Incluso si podemos dividir la imagen en parches adyacentes y luego reconstruirla a partir de estos, es probable que se genere un *efecto de borde* o una discontinuidad en la imagen resultante, como se puede ver en Fig. 5.31a. Esto ocurre porque en los bordes de los parches al realizar las convoluciones, estos reciben menos información de contexto, lo que resulta en predicciones menos precisas y genera artefactos visibles. Este problema se ve agravado en imágenes 3D, donde la falta de información contextual en de las 3 dimensiones provocan errores más notorios y complejos en la imagen reconstruida.

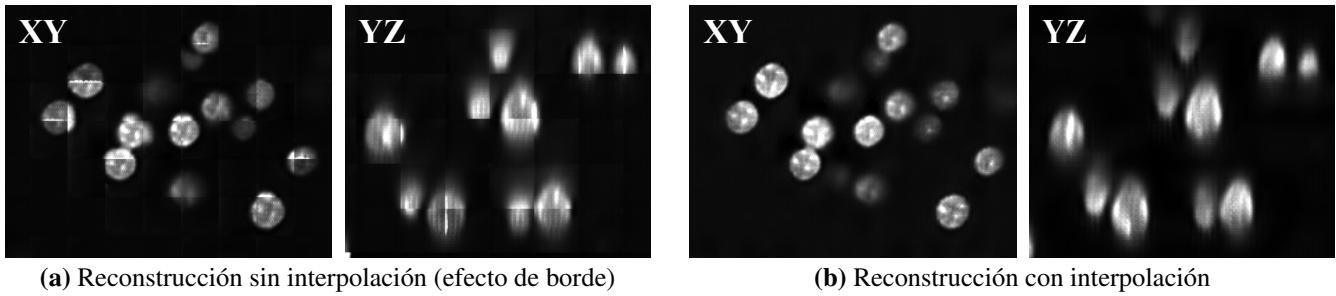


Fig. 5.31: Ejemplo de una reconstrucción de una imagen sin interpolación y con una interpolación sigmoide en el plano axial (XY) y en un plano ortogonal (YZ).

5.2.1 Preparación de la imagen

Para predecir un gran volumen de microscopía, es necesario que la imagen tenga dimensiones específicas que permitan su subdivisión en subvolúmenes o parches de tamaño de la entrada de la red, sin perder información en el proceso. Para disminuir el efecto de borde, una estrategia efectiva es utilizar un solapamiento entre cada parche adyacente y aplicar técnicas de fusión (*blending*) para suavizar la transición. Antes de subdividir la imagen, se agrega un relleno con ceros a lo largo de los ejes X, Y y Z. Este proceso, ilustrado en la Fig. 5.32, se realiza con base en las dimensiones de cada eje, el tamaño del parche y el valor del solapamiento entre parches adyacentes, calculado según la Ec. 5.2.1.

$$\text{Padding size} = \left\lceil \frac{\text{Image size} - \text{Patch size}}{\text{Patch size} \times (1 - \text{Overlap})} \right\rceil \times (\text{Patch size} \times (1 - \text{Overlap})) - \text{Image size} \quad (5.2.1)$$

Donde en Ec. (5.2.1), Image size es el tamaño de la imagen en el eje correspondiente (X, Y ó Z), Patch size es el tamaño del parche en el eje correspondiente y Overlap es el valor de solapamiento entre parches adyacentes (por ejemplo, 0.5 para un solapamiento del 50 %).

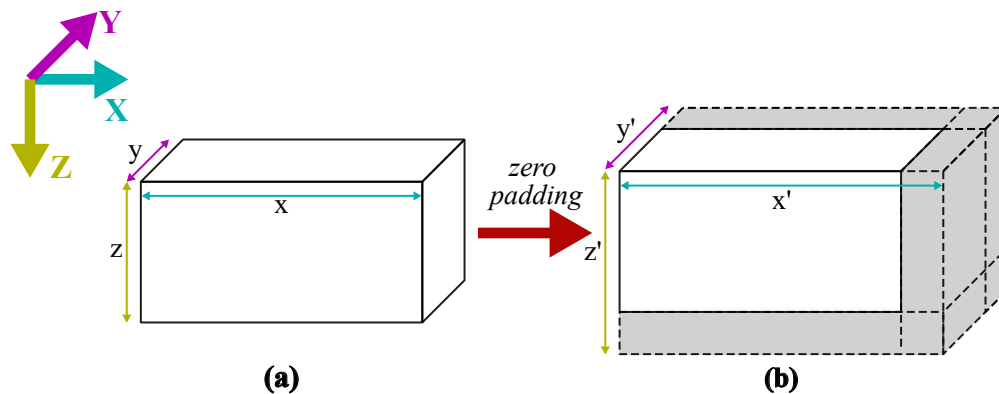


Fig. 5.32: Proceso en el que se le agrega un relleno con ceros a la imagen que queremos predecir. (a) Imagen original. (b) Imagen con relleno con ceros agregado.

5.2.2 Unión de parches

Luego, una vez se generan todos los subvolumenes de la imagen con el tamaño de la entrada, estos pasan a través de la red para generar la segmentación binaria y el campo vectorial, que se utiliza para calcular el centroide estimado.

Como se mencionó anteriormente, para poder reconstruir la imagen original a partir de los parches y reducir el efecto de borde, es necesario aplicar una técnica de fusión de parches que genere una transición suave en las áreas de solapamiento entre parches adyacentes. La idea general es que, cuando existe un solapamiento entre dos parches adyacentes (parches A y B), como se muestra en la Fig. 5.34a, en lugar de simplemente concatenar A sobre B, como se ve en la Fig. 5.34b, se genere una transición suave en la región de solapamiento. En esta transición, la región solapada se parece más a A en el lado izquierdo y más a B en el lado derecho. Esto se puede lograr mediante la Ec. 5.2.2.

$$\text{Overlap} = (1 - \alpha) \cdot A + \alpha \cdot B \quad (5.2.2)$$

Donde Ec. 5.2.2 Overlap es la región de solapamiento entre A y B , α es un coeficiente que varía entre 0 y 1, y A y B son los dos parches adyacentes, respectivamente.

Inicialmente, α se implementó como un coeficiente que varía linealmente, pero también se puede implementar como una función de transición más suave en los extremos y más rápida en el medio del solapamiento, como una función sigmoide, Fig. 5.33.

Los resultados muestran un solapamiento que transiciona progresivamente de A a B en la Fig. 5.34c,

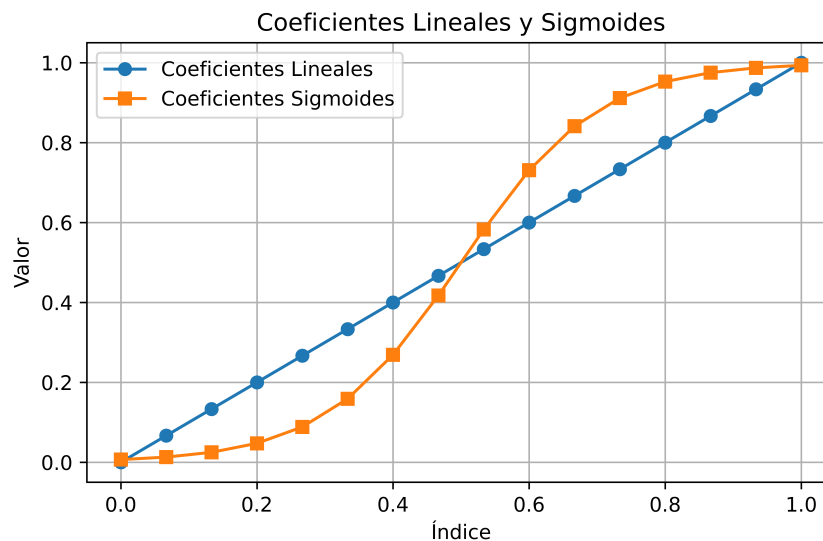


Fig. 5.33: Ejemplo de una función lineal y una función sigmoide para la transición de α con 16 coeficientes.

e, o una transición más suave en los extremos y más rápida en el centro, Fig. 5.34d, f. Por último, dado que estamos trabajando con parches 3D, es necesario realizar la interpolación en las tres direcciones (X, Y y Z) para reconstruir la máscara al tamaño original de la imagen. Esta interpolación en múltiples dimensiones garantiza una transición suave y continua entre los parches en todas las direcciones, como se ilustra en la Fig. 5.35. Los resultados de cómo mejora la predicción de una imagen se pueden ver en Fig. 5.31b

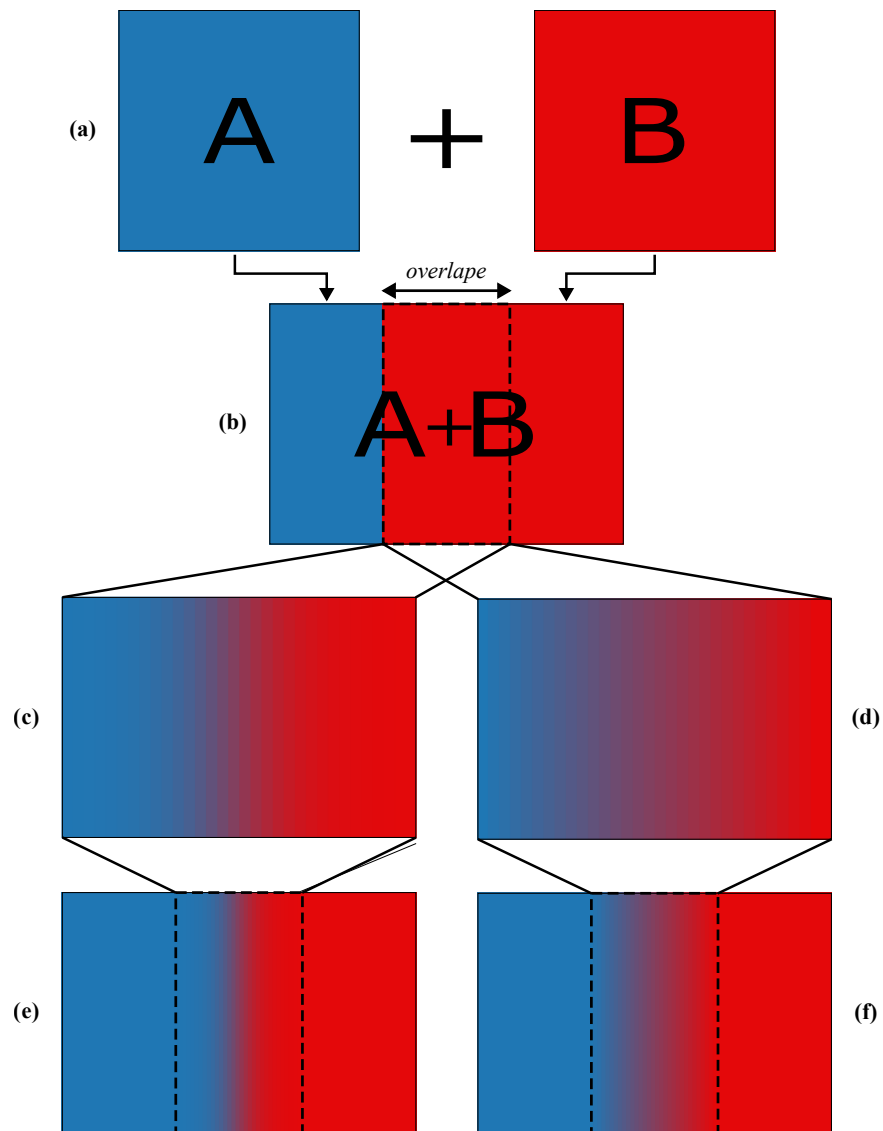


Fig. 5.34: Interpolación entre parches con solapamiento. **(a)** Parches adyacentes A y B con un solapamiento entre ellos. **(b)** Concatenación simple de A y B. **(c)** y **(d)** Regiones con solapamiento de A y B con interpolación lineal y sigmoide, respectivamente. **(e)** y **(f)** Parches A y B con la región solapada interpolada linealmente y con sigmoide, respectivamente.

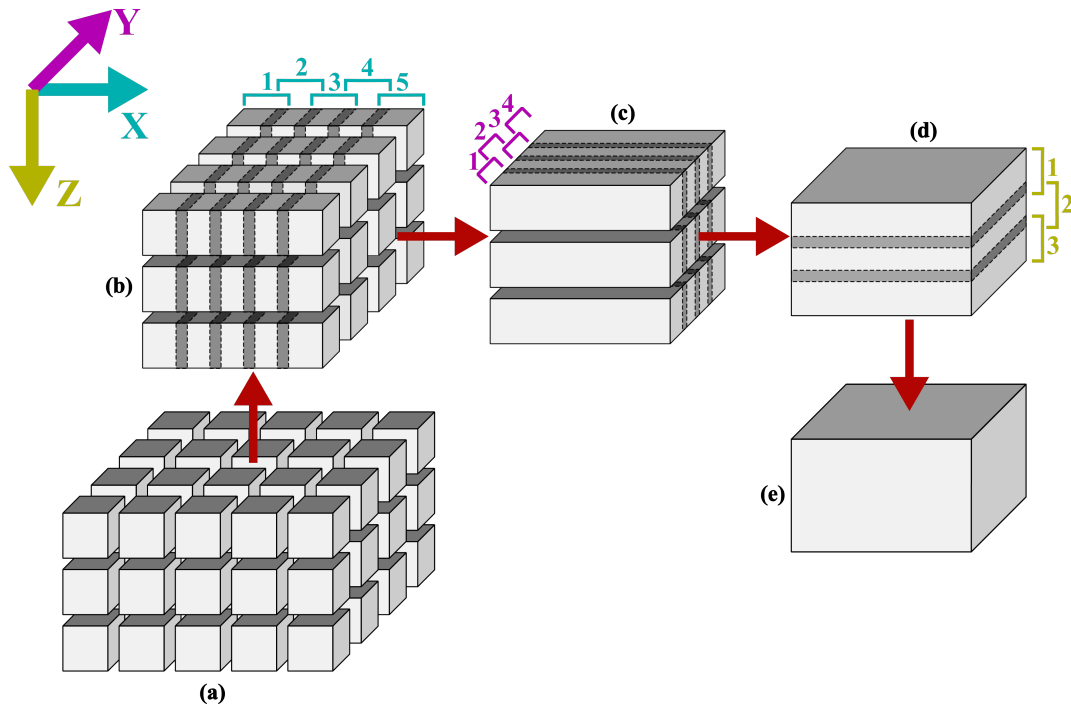


Fig. 5.35: Proceso de interpolación entre parches adyacentes con solapamiento. **(a)** Volumen de la imagen predicha separado en parches con solapamiento entre cada parche adyacente. **(b)** Interpolación de los parches con solapamiento en el eje X. **(c)** Interpolación de los parches con solapamiento en el eje Y. **(d)** Interpolación de los parches con solapamiento en el eje Z. **(e)** Reconstrucción de la imagen predicha en el volumen original.

5.3 Post-procesado

Luego de haber generado el volumen de la imagen con su máscara binaria, Fig. 5.36a,b en conjunto con la gradiente 3D, Fig. 5.36c-e, para generar una segmentación por instancias, es necesario realizar una serie de pasos luego de la predicción.

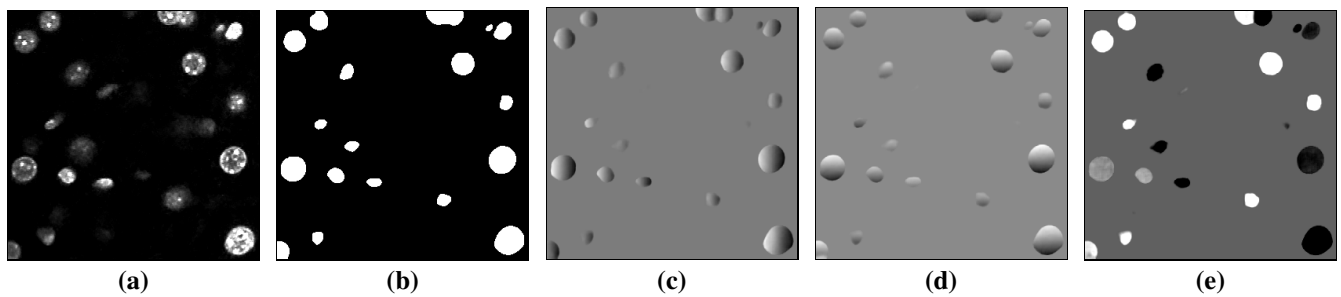


Fig. 5.36: Resultados de la red. **(a)** Imagen original. **(b)** Segmentación binaria. **(c)** Gradiente en X. **(d)** Gradiente en Y. **(e)** Gradiente en Z.

5.3.1 Metodología

Para poder obtener generar una segmentación por instancia a partir de las máscara binaria y el gradiente 3D predicho se propone utilizar la metodología de [4]. A continuación se detallan los pasos del proceso:

1. **Aproximación de coordenadas:** Se aproximan las posiciones finales de las coordenadas del primer plano, obtenidas por descenso de gradiente ($t = T$), $\{(x_n^{t=T}, y_n^{t=T}, z_n^{t=T})\}$. Se aproximan mediante el redondeo hacia abajo, es decir, $\{(\lfloor x_n^{t=T} \rfloor, \lfloor y_n^{t=T} \rfloor, \lfloor z_n^{t=T} \rfloor)\}$. Luego, los valores se recortan para que estén dentro de los límites del volumen de imagen $L \times M \times N$, es decir, $0 \leq \lfloor x_n^{t=T} \rfloor \leq L - 1$, $0 \leq \lfloor y_n^{t=T} \rfloor \leq M - 1$, $0 \leq \lfloor z_n^{t=T} \rfloor \leq N - 1$.
2. **Conteo de puntos:** Se tabula el número de puntos en cada posición de voxel, con cada punto contribuyendo con una cuenta de +1.
3. **Filtrado Gaussiano:** La imagen de conteo se filtra utilizando un filtro Gaussiano con $\sigma = 1$. Este paso ayuda a considerar la incertidumbre y a conectar espacialmente los puntos en un clúster de manera suave. Cuanto mayor sea el valor de σ , más puntos cercanos se agruparán en el mismo punto (centroide). Esto puede ser útil al segmentar estructuras alargadas.
4. **Segmentación de la imagen de conteo:** La imagen de conteo dispersa, se segmenta utilizando un umbral medio ($umbral = mean(\rho)$).
5. **Asignación de IDs:** A cada componente conexa del paso anterior se le asigna a un ID. Para el análisis de componentes se utiliza la implementación optimizada y paralela desarrollada por el Laboratorio Seung [82].
6. **Agrupación de Coordenadas:** Agrupa todas las coordenadas de la posición final el ID del centroide que por proximidad espacial está mas cerca.

$$L_{t=T}(x_n^T, y_n^T, z_n^T) = id \in \mathbb{Z}^+ \quad (5.3.1)$$

7. **Segmentación 3D:** Transfiere las etiquetas de las coordenadas finales a las coordenadas iniciales para obtener la segmentación de instancias 3D.

$$\text{Segmentación 3D} := L_{t=0}(x_n^{t=0}, y_n^{t=0}, z_n^{t=0}) = L_{t=T}(x_n^T, y_n^T, z_n^T) = id \in \mathbb{Z}^+ \quad (5.3.2)$$

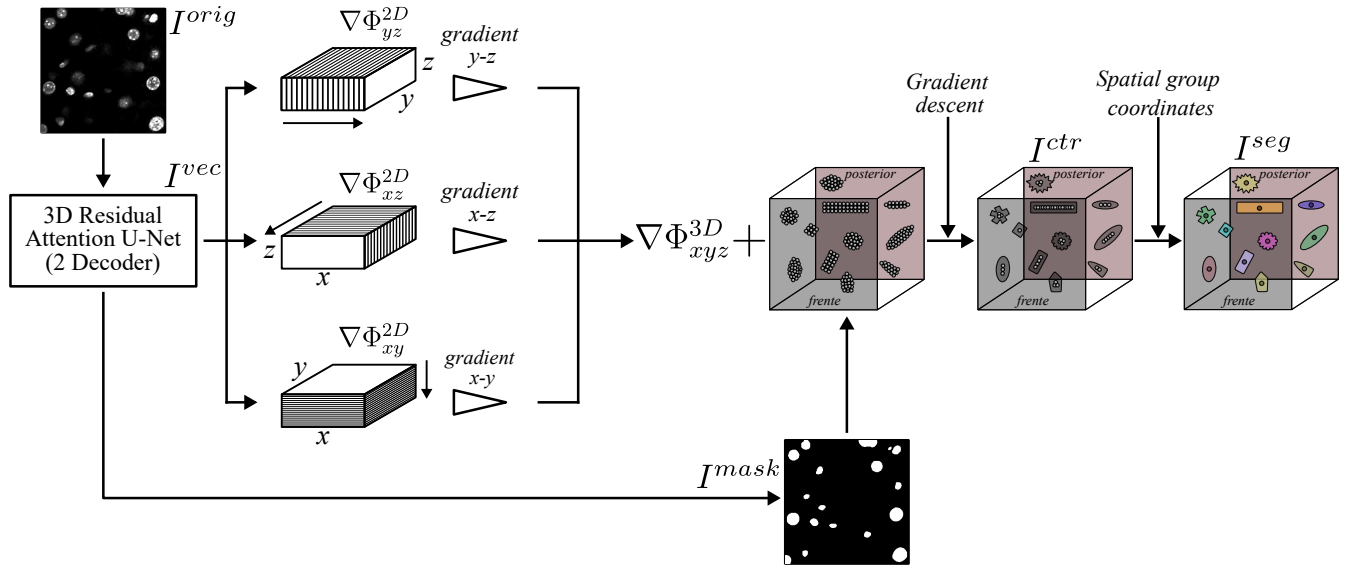


Fig. 5.37: Esquema de la metodología de post-procesamiento para obtener una segmentación por instancia de los núcleos. I^{orig} es la imagen original, I^{vec} es la gradiente 3D predicha ($\nabla\Phi_{xyz}^{3D}$). $\nabla\Phi_{yz}^{2D}$, $\nabla\Phi_{xz}^{2D}$ y $\nabla\Phi_{xy}^{2D}$ son las gradientes en los ejes X, Y y Z, respectivamente, como se muestra en la Fig. 5.36c-e. I^{mask} es la segmentación binaria predicha. I^{ctr} son los centroides obtenidos al aplicar el algoritmo de descenso de gradientes en la máscara binaria y la gradiente 3D. Finalmente, I^{seg} son los núcleos segmentados por instancia, obtenidos al agrupar las coordenadas por proximidad espacial.

5.4 Resultados

La visualización de la segmentación de C3 se puede observar en Fig. 5.38.

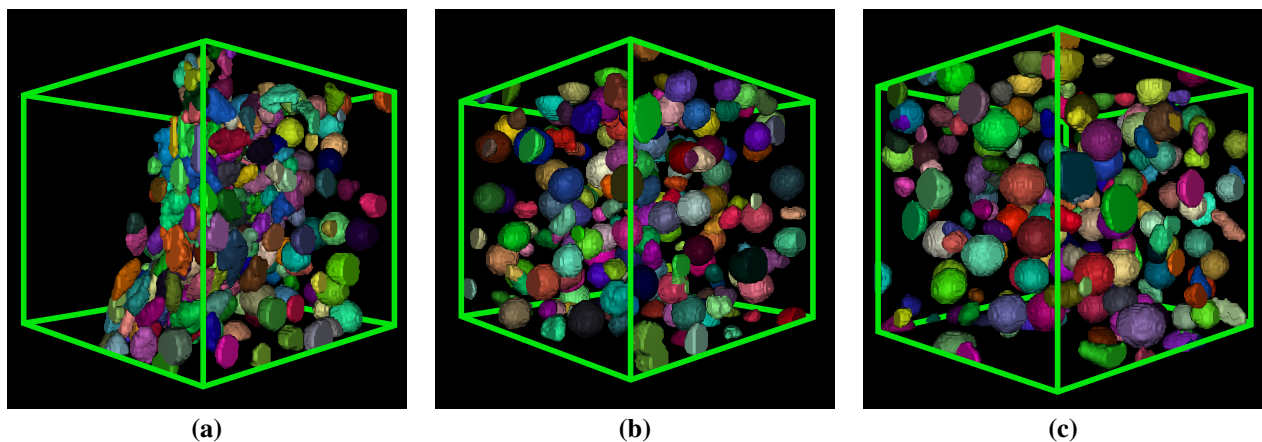


Fig. 5.38: Resultado de segmentación de C3. (a) Región periportal. (b) Región central. (c) Región pericentral.

5.5 Discusión

La predicción y segmentación de grandes volúmenes de imágenes de microscopía en 3D mediante redes neuronales profundas presenta varios desafíos técnicos y computacionales. La necesidad de ajustar las dimensiones de entrada de la red y de manejar eficazmente la reconstrucción de la imagen original a partir de los parches predichos es fundamental para obtener resultados precisos.

Uno de los principales problemas abordados en este capítulo es el **efecto de borde** que se produce cuando se segmentan grandes imágenes. Este efecto se mitiga eficazmente mediante la técnica de solapamiento e interpolación, que suaviza las transiciones entre los parches y reduce los artefactos en los límites. La utilización de funciones de transición mejora la calidad de la reconstrucción final en comparación a simplemente concatenar los parches uno al lado del otro.

La metodología implementada para generar segmentación por instancia, adaptada de [4] ha demostrado ser eficaz para separar núcleos densamente agrupados, utilizando el campo vectorial y la máscara binaria predichos por la red. La técnica permite una segmentación precisa de los núcleos, facilitando la identificación y el análisis en grandes volúmenes de datos.

Capítulo 6. Cuantificación de Resultados

6.1 Introducción

En esta sección, presentamos una evaluación del Modelo Propuesto. Lo comparamos con un método del estado del arte, Cellpose, y un método tradicional, Motion Tracking el cual además utiliza Watershed (MT+WS). Evaluamos el rendimiento de los modelos utilizando un conjunto de datos específico (C1 y C2 para el entrenamiento y C3 para la evaluación), en las distintas regiones del lobulillo hepático: periportal, central y pericentral. Se implementó una metodología de emparejamiento aprox. similar a la utilizada por StarDist [1, 2] para evaluar las segmentaciones. Se mide el rendimiento mediante métricas estándar como la precisión, el recall, la precisión media (AP) y el puntaje F1.

6.2 Cuantificación de resultados

Al igual que con nuestro método propuesto, entrenamos Cellpose con C1 y C2. Luego utilizamos la misma implementación de emparejamiento aproximado que StarDist [1, 2], para evaluar nuestros 3 métodos (MT+WS, Cellpose y el Modelo Propuesto). Utilizamos los conjuntos de datos de C3 (región periportal, central y pericentral) para evaluar. Las métricas de evaluación utilizadas incluyen la intersección sobre unión (IoU), la precisión, el recall, el F1-Score y la AP. Los resultados se pueden ver en la Tabla 6.4, los cuales se obtuvieron de los resultados por región de las Tablas A.1, A.2 y A.3. Además, se utiliza la AP para evaluar la calidad de la segmentación a diferentes umbrales de IoU y obtener una curva de precisión media representativa del desempeño del modelo en el conjunto de datos. En línea con Stardist y Cellpose, se muestra la curva AP para 11 umbrales de IoU que muestrean equitativamente el rango [0.5, 1.0] en la Fig. 6.39, los cuales se obtuvieron de las Tablas A.5, A.6 y A.6.

Método	Región Periportal				Región Central				Región Pericentral			
	Precision	Recall	F1-Score	AP	Precision	Recall	F1-Score	AP	Precision	Recall	F1-Score	AP
MT+WS	0.6836	0.2854	0.4027	0.2521	0.9282	0.7336	0.8195	0.6942	0.7514	0.6456	0.6945	0.532
Cellpose	0.5118	0.5637	0.5365	0.3666	0.8991	0.8952	0.8972	0.8135	0.8685	0.8981	0.8831	0.7906
Modelo Propuesto	0.7582	0.6509	0.7005	0.5391	0.9378	0.9214	0.9295	0.8683	0.9147	0.9369	0.9257	0.8616

Tabla 6.4: Resultados del rendimiento de los modelos en C3. El mejor desempeño con respecto a cada métrica está en negrita. Utilizamos la precisión, el recall, la precisión media (AP) y el puntaje F1 (F1-Score) para evaluar el rendimiento de la segmentación de los modelos.

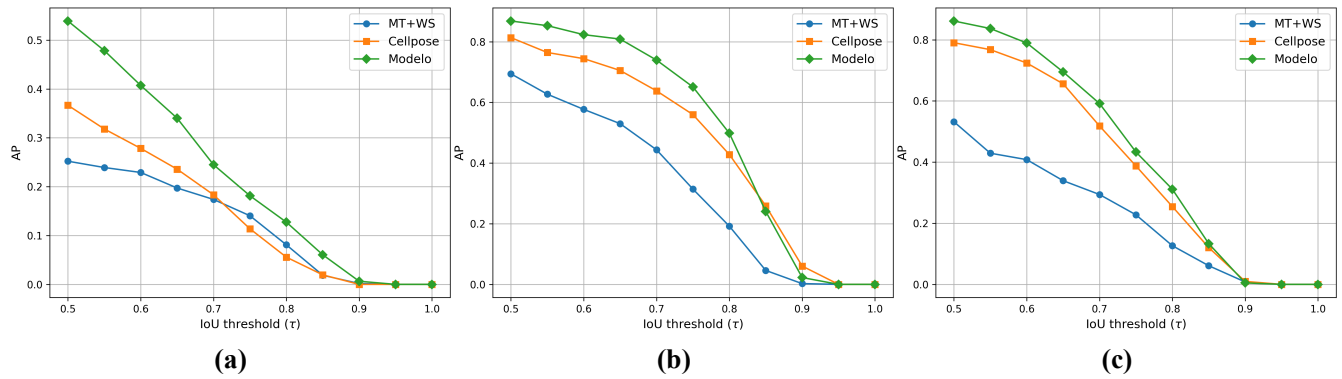


Fig. 6.39: Precisión promedio para varios umbrales de IoU (τ) de los 3 modelos en: (a) la región periportal, (b) la región central y (c) la región pericentral.

6.3 Discusión

De los resultados en Tabla 6.4 podemos observar que, efectivamente en el mejor caso (región central), Motion Tracking (MT) tiene una precisión del 90 %. Sin embargo, al analizar otras métricas, como el recall, se confirma que existe una proporción de núcleos, aproximadamente el 30 %, que no se detectan. Además, al evaluar AP, vemos que la calidad general es del 70 %, lo que representa un 20 % menos en comparación con la precisión inicial del 90 %. Esto se refleja en la calidad general de la segmentación. En la Figura 6.39, al comparar la robustez de los tres modelos, observamos que MT es el que genera segmentaciones de menor calidad en las 3 regiones, especialmente en la región periportal. En comparación con Cellpose y el modelo propuesto, MT es deficiente en la calidad general de la segmentación, siendo solo comparable en la detección de núcleos reales (precisión) en la región central e impreciso en las otras regiones, sobre todo en la región pericentral.

Al comparar nuestro modelo con Cellpose, podemos notar que está a la par con un modelo del estado del arte y lo supera ligeramente en todas las regiones. En particular, muestra una mayor precisión y F1-Score en comparación con Cellpose y MT, destacando su capacidad para manejar la segmentación de estructuras complejas en 3D. Además, es el más robusto de los tres modelos según la Figura 6.39.

A diferencia de Cellpose, nuestro modelo considera el contexto 3D durante el entrenamiento y no solo en las predicciones. Esta característica representa una ventaja, pero también una desventaja, ya que la generación de una base de datos útil para el entrenamiento y evaluación del modelo es mucho más costosa, además de necesitar muchos más recursos computacionales.

Capítulo 7. Conclusiones

7.1 Discusión

El desarrollo de nuevas técnicas y tecnologías ha permitido avances significativos en el campo de la biología, particularmente en la comprensión de la estructura y función de los órganos a nivel celular. Una de las herramientas más importantes en la investigación biológica es la microscopía de fluorescencia, que permite visualizar y analizar la distribución de moléculas específicas dentro de las células y tejidos. En los últimos años, el aprendizaje profundo han emergido como una poderosa herramienta para el análisis de imágenes biológicas. Las redes neuronales convolucionales, en particular, han demostrado ser altamente efectivas para la segmentación de imágenes y la detección de estructuras celulares. Sin embargo, la complejidad del hígado y la necesidad de una gran cantidad de datos anotados presentan desafíos significativos.

El uso de un modelo 3D basado en U-Net con bloques residuales y puertas de atención ha demostrado ser eficaz para la segmentación semántica de imágenes biomédicas. Sin embargo, la segmentación por instancia de núcleos celulares en 3D, se realizó en este caso mediante la adición de un segundo decodificador para mejorar la diferenciación de núcleos individuales. Esta técnica, combinada con la generación de un campo de flujo vectorial a partir de las etiquetas de los núcleos, permitió la segmentación por instancia, y logrando una caracterización más detallada de los núcleos celulares.

La función de pérdida compuesta, que integra la pérdida de Tversky (TL), la pérdida focal (FL) y el error cuadrático medio (MSE), optimiza el entrenamiento del modelo para múltiples tareas. El uso de funciones de pérdida especializadas mejora la capacidad del modelo para manejar diferentes tipos de errores y desafíos durante el entrenamiento. Además, la aumentación de datos es una estrategia clave para aliviar el problema de datos insuficientes en el entrenamiento, mejorando la capacidad del modelo para generalizar y desempeñarse bien en imágenes no vistas.

La predicción y segmentación de grandes volúmenes de imágenes de microscopía en 3D mediante redes neuronales profundas presenta varios desafíos técnicos y computacionales. La necesidad de ajustar las dimensiones de entrada de la red y de manejar eficazmente la reconstrucción de la imagen original a partir de los parches predichos es fundamental para obtener resultados precisos. Uno de los principales problemas abordados en este trabajo es el efecto de borde que se produce cuando se segmentan grandes imágenes. Este efecto se mitiga eficazmente mediante la técnica de solapamiento e interpolación, que suaviza las transiciones entre los parches y reduce los artefactos en los límites.

Además, el análisis de los resultados indicó que el algoritmo desarrollado no solo es robusto y preciso, sino que también puede ser utilizado en imágenes con las cuales no se entrenó. Sin embargo, se observó que la variabilidad de los núcleos en las imágenes influye en el rendimiento del algoritmo, sobre todo en las secciones donde hasta el ojo humano entrenado le dificulta segmentar. Este trabajo se basó en un conjunto limitado de sujetos, lo que sugiere la necesidad de ampliar el conjunto de datos para mejorar la generalización y robustez del modelo.

Es importante destacar que la integración de técnicas avanzadas de procesamiento de imágenes y aprendizaje profundo no solo mejoró la calidad de la segmentación, sino que permitió la automatización de procesos que tradicionalmente requerían una intervención manual intensiva, ya sea en el ajuste de parámetros o en la post corrección manual. Este avance es crucial para disminuir el tiempo empleado actualmente en el laboratorio para generar segmentaciones de núcleos, donde el aumento de las imágenes y el tiempo que demora en generar una segmentación correcta terminaba siendo un cuello de botella al querer realizar un análisis cuantitativo.

Los resultados obtenidos validan la efectividad del redes neuronales para la segmentación de imágenes de microscopía de fluorescencia confocal en tejido hepático. Este trabajo no solo aporta una herramienta poderosa para la segmentación de imágenes en el laboratorio, sino que también establece una base sólida para generar nuevos descubrimientos en esta área.

7.2 Conclusiones

El desarrollo de una base de datos específica y la implementación de un algoritmo de segmentación automática basado en redes neuronales constituyen una contribución significativa para el laboratorio y el campo del análisis de imágenes de microscopía de tejido hepático, debido a la complejidad de este tejido. Este trabajo no solo demuestra la viabilidad de utilizar técnicas de aprendizaje profundo para la segmentación de núcleos celulares, sino que también establece una base sólida para futuras investigaciones y aplicaciones en esta área. Los principales resultados son:

- Una base de datos de núcleos hepáticos 3D para el entrenamiento y evaluación de redes neuronales.
- Una segmentación precisa y eficiente de los núcleos celulares en imágenes de microscopía de tejido hepático, superando a los métodos tradicionales utilizados en el laboratorio que está a la par con modelos del estado del arte.
- La implementación exitosa de un pipeline integral que combina técnicas avanzadas y robustas de procesamiento de imágenes y aprendizaje profundo, permitiendo una automatización de procesos

que tradicionalmente requerían intervención manual intensiva.

La base de datos generada es un recurso valioso que mejora significativamente la segmentación de imágenes biomédicas, permitiendo la evaluación comparativa y el desarrollo de nuevos algoritmos. La implementación detallada del algoritmo aborda eficazmente los desafíos de segmentar grandes volúmenes de datos 3D, mejorando la calidad de la segmentación y reduciendo el tiempo y esfuerzo necesarios para la segmentación manual. Este avance es crucial para el laboratorio, ya que permite manejar grandes volúmenes de datos de manera rápida y precisa. La combinación de una base de datos de alta calidad y un pipeline de segmentación bien estructurado facilita el análisis de nuevas imágenes y promueve nuevos descubrimientos en esta área.

7.3 Trabajo a Futuro

Aunque los resultados de este trabajo son positivos, hay varios aspectos que merecen una mayor profundización. Debido a restricciones de tiempo y algunos desafíos en la implementación eficiente de ciertas ideas, algunas secciones podrían beneficiarse de mejoras adicionales. Se proponen las siguientes líneas de trabajo futuro:

- **Obtención de Datos Adicionales:** Incrementar la cantidad de datos segmentados manual o semi manual para generar una mejor segmentación de núcleos celulares. Además, emplear redes generativas para aumentar artificialmente la cantidad de datos disponibles, mejorando la capacidad del algoritmo para generalizar en diferentes condiciones y ampliando el conjunto de entrenamiento con datos sintéticos de alta calidad.
- **Optimización de la Red Neuronal:** Si bien se entrenaron distintas redes con distintos parámetros, algunas cosas como la normalización de la gradiente 3D, utilizations de funciones *leaky ReLU* ó optimizar la arquitectura de la red neuronal para reducir el consumo de memoria, no lograron probarse debido al tiempo que se tenía.
- **Aplicación en Sujetos no Control:** Extender el uso del algoritmo a imágenes de sujetos no control en imágenes de núcleos hepáticas, por ejemplo, aquellas con progresión de cáncer o en procesos de regeneración hepática, para validar su eficacia y utilidad en diferentes contextos. Esto significaría segmentar manual o semi manualmente nuevas imágenes.

Capítulo 8. Glosario

1D	Una Dimensión o Unidimensional
2D	Dos Dimensiones o Bidimensional
3D	Tres Dimensiones o Tridimensional
ADN	Ácido Desoxirribonucleico (Deoxyribonucleic Acid)
AI	Inteligencia Artificial (Artificial Intelligence)
ANNs	Redes Neuronales Artificiales (Artificial Neural Networks)
AP	Precisión Promedio (Average Precision)
ARRIVE	Investigación con Animales: Informes de Experimentos In Vivo (Animal Research: Reporting of In Vivo Experiments)
BC	Canalículos Biliares (Bile Canaliculi)
BCE	Entropía Cruzada Binaria (Binary Cross-Entropy)
CNN	Red Neuronal Convolutiva (Convolutional Neural Network)
CREAV	Centro Regional de Estudios Avanzados para la Vida
CV	Vena Central (Central Vein)
DL	Aprendizaje Profundo (Deep Learning)
DNN	Red Neuronal Completamente Conectada (Dense Neural Network)
ECM	Matriz Extracelular (Extracellular Matrix)
FL	Pérdida Focal (Focal Loss)
FN	Falsos Negativos (False Negatives)
FP	Falsos Positivos (False Positives)
GaAsp	Fosforo de Arseniuro de Galio (Gallium Arsenide Phosphide)
GT	Verdad Fundamental (Ground Truth)
HA	Arteria Hepática (Hepatic Artery)
HSC	Célula Estrellada del Hígado (Hepatic Stellate Cell)
IoU	Intersección sobre Unión (Intersection over Union)
KC	Células de Kupffer (Kupffer Cell)
MAE	error absoluto medio (Mean Absolute Error)
ML	Aprendizaje Automático (Machine Learning)
MT	Motion Tracking
NMS	Supresión de No Máximos (Non-Maximum Suppression)
PSF	Función de Dispersión de Puntos (Point Spread Function)

PV	Vena Porta (Portal Vein)
ROIs	Regiones de Interés (Regions of Interest)
RPSV	Posición Reescalada por Velocidad Cuadrada (Rescaled Position by Square Velocity)
SECs	Células Endoteliales Sinusoidales (Sinusoidal Endothelial Cells)
TL	Pérdida de Tversky (Tversky Loss)
TP	Verdaderos Positivos (True Positives)

Capítulo 9. Referencias

- [1] U. Schmidt, M. Weigert, C. Broaddus, and G. Myers, “Cell Detection with Star-Convex Polygons,” in *Medical Image Computing and Computer Assisted Intervention – MICCAI 2018*. Springer International Publishing, 2018, pp. 265–273. [Online]. Available: https://doi.org/10.1007%2F978-3-030-00934-2_30
- [2] M. Weigert, U. Schmidt, R. Haase, K. Sugawara, and G. Myers, “Star-convex Polyhedra for 3D Object Detection and Segmentation in Microscopy,” in *2020 IEEE Winter Conference on Applications of Computer Vision (WACV)*. IEEE, Mar. 2020. [Online]. Available: <http://dx.doi.org/10.1109/WACV45572.2020.9093435>
- [3] C. Stringer, T. Wang, M. Michaelos, and M. Pachitariu, “Cellpose: a generalist algorithm for cellular segmentation,” *Nature Methods*, vol. 18, no. 1, pp. 100–106, Jan. 2021.
- [4] F. Y. Zhou, C. Yapp, Z. Shang, S. Daetwyler, Z. Marin, M. T. Islam, B. Nanes, E. Jenkins, G. M. Gihana, B.-J. Chang, A. Weems, M. Dustin, S. Morrison, R. Fiolka, K. Dean, A. Jamieson, P. K. Sorger, and G. Danuser, “A general algorithm for consensus 3D cell segmentation from 2D segmented stacks,” *bioRxiv*, 2024. [Online]. Available: <https://www.biorxiv.org/content/early/2024/05/06/2024.05.03.592249.1>
- [5] L. Wu, A. Chen, P. Salama, S. Winfree, K. W. Dunn, and E. J. Delp, “NISNet3D: three-dimensional nuclear synthesis and instance segmentation for fluorescence microscopy images,” *Sci Rep*, vol. 13, no. 1, p. 9533, Jun. 2023.
- [6] S. F. Gilbert, *Developmental biology*. sinauer associates, Inc, 2010.
- [7] P. Ruusuvuori, M. Valkonen, K. Kartasalo, M. Valkonen, T. Visakorpi, M. Nykter, and L. Latonen, “Spatial analysis of histology in 3D: quantification and visualization of organ and tumor level tissue environment,” *Heliyon*, vol. 8, no. 1, 2022.
- [8] A. Kalra, E. Yetiskul, C. J. Wehrle, and F. Tuma, “Physiology, Liver,” in *StatPearls*. Treasure Island (FL): StatPearls Publishing, Jan. 2024.
- [9] H. Morales-Navarrete, H. Nonaka, A. Scholich, F. Segovia-Miranda, W. de Back, K. Meyer, R. L. Bogorad, V. Koteliensky, L. Bruschi, Y. Kalaidzidis, F. Jülicher, B. M. Friedrich, and M. Zerial,

“Liquid-crystal organization of liver tissue,” *eLife*, vol. 8, p. e44860, jun 2019. [Online]. Available: <https://doi.org/10.7554/eLife.44860>

- [10] H. Morales-Navarrete, F. Segovia-Miranda, P. Klukowski, K. Meyer, H. Nonaka, G. Marsico, M. Chernykh, A. Kalaidzidis, M. Zerial, and Y. Kalaidzidis, “A versatile pipeline for the multi-scale digital reconstruction and quantitative analysis of 3D tissue architecture,” *eLife*, vol. 4, p. e11214, dec 2015. [Online]. Available: <https://doi.org/10.7554/eLife.11214>
- [11] Z. Liu, L. Jin, J. Chen, Q. Fang, S. Ablameyko, Z. Yin, and Y. Xu, “A survey on applications of deep learning in microscopy image analysis,” *Computers in Biology and Medicine*, vol. 134, p. 104523, 2021. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0010482521003176>
- [12] F. Long, “Microscopy cell nuclei segmentation with enhanced U-Net,” *BMC bioinformatics*, vol. 21, pp. 1–12, 2020.
- [13] H. S. Behl, A. G. Baydin, R. Gal, P. H. S. Torr, and V. Vineet, “Autosimulate: (Quickly) Learning Synthetic Data Generation,” in *Computer Vision – ECCV 2020*, A. Vedaldi, H. Bischof, T. Brox, and J.-M. Frahm, Eds. Cham: Springer International Publishing, 2020, pp. 255–271.
- [14] J. Chen, L. Ding, M. P. Viana, H. Lee, M. F. Sluezwski, B. Morris, M. C. Hendershott, R. Yang, I. A. Mueller, and S. M. Rafelski, “The Allen Cell and Structure Segmenter: a new open source toolkit for segmenting 3D intracellular structures in fluorescence microscopy images,” *bioRxiv*, 2018. [Online]. Available: <https://www.biorxiv.org/content/early/2020/12/13/491035>
- [15] Z. Ni, G. Bian, X. Zhou, Z. Hou, X. Xie, C. Wang, Y. Zhou, R. Li, and Z. Li, “RAUNet: Residual Attention U-Net for Semantic Segmentation of Cataract surgical instruments,” *CoRR*, vol. abs/1909.10360, 2019. [Online]. Available: <http://arxiv.org/abs/1909.10360>
- [16] E. G. Fischer, “Nuclear morphology and the biology of cancer cells,” *Acta cytologica*, vol. 64, no. 6, pp. 511–519, 2020.
- [17] N. Sofroniew, T. Lambert, K. Evans, J. Nunez-Iglesias, G. Bokota, P. Winston, G. Peña-Castellanos, K. Yamauchi, M. Bussonnier, D. Doncila Pop, A. Can Solak, Z. Liu, P. Wadhwa, A. Burt, G. Buckley, A. Sweet, L. Migas, V. Hilsenstein, L. Gaifas, J. Bragantini, J. Rodríguez-Guerra, H. Muñoz, J. Freeman, P. Boone, A. Lowe, C. Gohlke, L. Royer, A. PIERRÉ, H. Har-Gil, and A. McGovern, “napari: a multi-dimensional image viewer for Python,” Nov. 2022, If you use this software, please cite it using these metadata. [Online]. Available: <https://doi.org/10.5281/zenodo.7276432>

- [18] G. Van Rossum and F. L. Drake, *Python 3 Reference Manual*. Scotts Valley, CA: CreateSpace, 2009.
- [19] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng, “TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems,” 2015, software available from tensorflow.org. [Online]. Available: <https://www.tensorflow.org/>
- [20] O. Ronneberger, P. Fischer, and T. Brox, “U-Net: Convolutional Networks for Biomedical Image Segmentation,” 2015.
- [21] H. Elias, “A re-examination of the structure of the mammalian liver. ii. the hepatic lobule and its relation to the vascular and biliary systems,” *American Journal of Anatomy*, vol. 85, no. 3, pp. 379–456, 1949. [Online]. Available: <https://onlinelibrary.wiley.com/doi/abs/10.1002/aja.1000850303>
- [22] I. Arias, J. Boyer, H. Alter, N. Fausto, and D. Cohen, *The Liver: Biology and Pathobiology*. Wiley, 2009. [Online]. Available: <https://books.google.cl/books?id=REmXGPDFzIoC>
- [23] W. Pawlina and M. Ross, *Histology: A Text and Atlas: With Correlated Cell and Molecular Biology*. Wolters Kluwer Health, 2018. [Online]. Available: <https://books.google.cl/books?id=5vx9DwAAQBAJ>
- [24] D. Martínez-Torres, V. Maldonado, C. Pérez-Gallardo, R. Yañez, V. Candia, Y. Kalaidzidis, M. Zerial, H. Morales-Navarrete, and F. Segovia-Miranda, “Phenotypic characterization of liver tissue heterogeneity through a next-generation 3D single-cell atlas,” *Scientific Reports*, vol. 14, no. 1, p. 2823, Feb 2024. [Online]. Available: <https://doi.org/10.1038/s41598-024-53309-4>
- [25] R. P. Cunningham and N. Porat-Shliom, “Liver Zonation - Revisiting Old Questions With New Technologies,” *Front Physiol*, vol. 12, p. 732929, Sep. 2021.
- [26] M. A. Freitas-Lopes, K. Mafra, B. A. David, R. Carvalho-Gontijo, and G. B. Menezes, “Differential Location and Distribution of Hepatic Immune Cells,” *Cells*, vol. 6, no. 4, Dec. 2017.
- [27] M. A. Omary and H. H. Patterson, “Luminescence Theory,” in *Encyclopedia of Spectroscopy and Spectrometry*, J. C. Lindon, Ed. Oxford: Elsevier, 1999, pp. 1186–1207. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/B0122266803001605>

- [28] M. J. Sanderson, I. Smith, I. Parker, and M. D. Bootman, “Fluorescence microscopy,” *Cold Spring Harb Protoc*, vol. 2014, no. 10, p. db.top071795, Oct. 2014.
- [29] W. Contributors, Nov 2023. [Online]. Available: https://en.wikipedia.org/wiki/Stokes_shift
- [30] V. Magidson and A. Khodjakov, “Chapter 23 - Circumventing Photodamage in Live-Cell Microscopy,” in *Digital Microscopy*, ser. Methods in Cell Biology, G. Sluder and D. E. Wolf, Eds. Academic Press, 2013, vol. 114, pp. 545–560. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/B9780124077614000233>
- [31] J. Sanderson, *Understanding Light Microscopy*, ser. RMS - Royal Microscopical Society. Wiley, 2019. [Online]. Available: <https://books.google.cl/books?id=9uOLDwAAQBAJ>
- [32] A. López-Macay, J. Fernández-Torres, and A. Zepeda, “Principios y aplicaciones de la microscopia láser confocal en la investigación biomédica,” *Investigación en Discapacidad*, vol. 5, no. 3, pp. 156–164, 2016.
- [33] S. W. Paddock, T. J. Fellers, and M. W. Davidson, “Introductory Confocal Concepts,” accessed: 2024-07-23. [Online]. Available: <https://www.microscopyu.com/techniques/confocal/introductory-confocal-concepts>
- [34] A3 Marketing Team, “What is fluorescence microscopy and 3D confocal imaging in Life Sciences Vision?” August 2019, accessed: 2024-07-23. [Online]. Available: <https://www.automate.org/vision/blogs/what-is-fluorescence-microscopy-and-3d-confocal-imaging-in-life-sciences-vision>
- [35] “What are TIFF files and how do you open them?” <https://www.adobe.com/cl/creativecloud/file-types/image/raster/tiff-file.html>, [Accessed 14-06-2024].
- [36] J. Schindelin, I. Arganda-Carreras, E. Frise, V. Kaynig, M. Longair, T. Pietzsch, S. Preibisch, C. Rueden, S. Saalfeld, B. Schmid, J.-Y. Tinevez, D. J. White, V. Hartenstein, K. Eliceiri, P. Tomancak, and A. Cardona, “Fiji: an open-source platform for biological-image analysis,” *Nat Meth*, vol. 9, no. 7, pp. 676–682, Jul. 2012. [Online]. Available: <http://dx.doi.org/10.1038/nmeth.2019>
- [37] S. Beucher and F. Meyer, *The Morphological Approach to Segmentation: The Watershed Transformation*. CRC Press, 01 1993, vol. Vol. 34, p. 433–481.
- [38] L. von Chamier, R. F. Laine, J. Jukkala, C. Spahn, D. Krentzel, E. Nehme, M. Lerche, S. Hernández-Pérez, P. K. Mattila, E. Karinou, S. Holden, A. C. Solak, A. Krull, T.-O. Buchholz, M. L. Jones,

- L. A. Royer, C. Leterrier, Y. Shechtman, F. Jug, M. Heilemann, G. Jacquemet, and R. Henriques, “Democratising deep learning for microscopy with ZeroCostDL4Mic,” *Nature Communications*, vol. 12, no. 1, p. 2276, Apr 2021. [Online]. Available: <https://doi.org/10.1038/s41467-021-22518-0>
- [39] J. Brownlee, “What is deep learning?” Aug 2020. [Online]. Available: <https://machinelearningmastery.com/what-is-deep-learning/>
- [40] G. Regunath, “Understanding the difference between AI, ML, and DL: Using an incredibly simple example,” Dec 2021. [Online]. Available: <https://www.advancinganalytics.co.uk/blog/2021/12/15/understanding-the-difference-between-ai-ml-and-dl-using-an-incredibly-simple-example>
- [41] C. C. Aggarwal *et al.*, “Neural networks and deep learning,” *Springer*, vol. 10, no. 978, p. 3, 2018.
- [42] S. B. Kotsiantis, I. Zaharakis, P. Pintelas *et al.*, “Supervised machine learning: A review of classification techniques,” *Emerging artificial intelligence applications in computer engineering*, vol. 160, no. 1, pp. 3–24, 2007.
- [43] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016, <http://www.deeplearningbook.org>.
- [44] A. Krogh, “What are artificial neural networks?” *Nature Biotechnology*, vol. 26, no. 2, pp. 195–197, Feb 2008. [Online]. Available: <https://doi.org/10.1038/nbt1386>
- [45] C. S. Vui, G. K. Soon, C. K. On, R. Alfred, and P. Anthony, “A review of stock market prediction with Artificial neural network (ANN),” in *2013 IEEE International Conference on Control System, Computing and Engineering*, 2013, pp. 477–482.
- [46] Z. Liu, L. Jin, J. Chen, Q. Fang, S. Ablameyko, Z. Yin, and Y. Xu, “A survey on applications of deep learning in microscopy image analysis,” *Computers in Biology and Medicine*, vol. 134, p. 104523, 2021.
- [47] L. von Chamier, R. F. Laine, J. Jukkala, C. Spahn, D. Krentzel, E. Nehme, M. Lerche, S. Hernández-Pérez, P. K. Mattila, E. Karinou, S. Holden, A. C. Solak, A. Krull, T.-O. Buchholz, M. L. Jones, L. A. Royer, C. Leterrier, Y. Shechtman, F. Jug, M. Heilemann, G. Jacquemet, and R. Henriques, “Democratising deep learning for microscopy with ZeroCostDL4Mic,” *Nature Communications*, vol. 12, no. 1, p. 2276, Apr. 2021.
- [48] V. H. Phung and E. J. Rhee, “A High-Accuracy Model Average Ensemble of Convolutional Neural Networks for Classification of Cloud Image Patches on Small Datasets,” *Applied Sciences*, vol. 9, no. 21, 2019. [Online]. Available: <https://www.mdpi.com/2076-3417/9/21/4500>

- [49] L. Deng, “The mnist database of handwritten digit images for machine learning research,” *IEEE Signal Processing Magazine*, vol. 29, no. 6, pp. 141–142, 2012.
- [50] R. Yamashita, M. Nishio, R. K. G. Do, and K. Togashi, “Convolutional neural networks: an overview and application in radiology,” *Insights into Imaging*, vol. 9, no. 4, pp. 611–629, Aug. 2018.
- [51] E. Anello, “Visualizing the feature maps and filters by Convolutional Neural Networks,” Nov 2022. [Online]. Available: <https://medium.com/dataseries/visualizing-the-feature-maps-and-filters-by-convolutional-neural-networks-e1462340518e>
- [52] A. Karpathy, J. Johnson, and F.-F. Li. (2016) CS231n: Convolutional Neural Networks for Visual Recognition. [Online]. Available: <https://cs231n.github.io/convolutional-networks/>
- [53] Arc, “Convolutional Neural Network,” Dec 2018. [Online]. Available: <https://towardsdatascience.com/convolutional-neural-network-17fb77e76c05>
- [54] F. Chollet, *Deep Learning with Python*, 1st ed. USA: Manning Publications Co., 2017.
- [55] J. Brownlee, “Supervised and unsupervised machine learning algorithms,” Oct 2023. [Online]. Available: <https://machinelearningmastery.com/supervised-and-unsupervised-machine-learning-algorithms/>
- [56] EnjoyAlgorithms, “Supervised, Unsupervised and Semi-supervised Learning,” <https://www.enjoyalgorithms.com/blogs/supervised-unsupervised-and-semisupervised-learning>, 2021.
- [57] A3 Marketing Team, “What is data augmentation?” accessed: 2024-07-23. [Online]. Available: <https://aws.amazon.com/what-is/data-augmentation/>
- [58] F. Sultana, A. Sufian, and P. Dutta, “Evolution of Image Segmentation using Deep Convolutional Neural Network: A Survey,” *Knowledge-Based Systems*, vol. 201-202, p. 106062, 2020. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0950705120303464>
- [59] R. Hollandi, N. Moshkov, L. Paavolainen, E. Tasnadi, F. Piccinini, and P. Horvath, “Nucleus segmentation: towards automated solutions,” *Trends Cell Biol*, vol. 32, no. 4, pp. 295–310, Jan. 2022.
- [60] M. Havaei, A. Davy, D. Warde-Farley, A. Biard, A. Courville, Y. Bengio, C. Pal, P.-M. Jodoin, and H. Larochelle, “Brain tumor segmentation with Deep Neural Networks,” *Medical Image Analysis*, vol. 35, pp. 18–31, 2017. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S1361841516300330>

- [61] P. Sharma, “A step-by-step guide to image segmentation techniques (part 1),” Jun 2024, [Accessed 13-06-2024]. [Online]. Available: <https://www.analyticsvidhya.com/blog/2019/04/introduction-image-segmentation-techniques-python/>
- [62] K. He, G. Gkioxari, P. Dollár, and R. Girshick, “Mask R-CNN,” 2018.
- [63] T. Falk, D. Mai, R. Bensch, Ö. Çiçek, A. Abdulkadir, Y. Marrakchi, A. Böhm, J. Deubner, Z. Jäckel, K. Seiwald, A. Dovzhenko, O. Tietz, C. Dal Bosco, S. Walsh, D. Saltukoglu, T. L. Tay, M. Prinz, K. Palme, M. Simons, I. Diester, T. Brox, and O. Ronneberger, “U-Net: deep learning for cell counting, detection, and morphometry,” *Nat Methods*, vol. 16, no. 1, pp. 67–70, Dec. 2018.
- [64] N. Siddique, S. Paheding, C. P. Elkin, and V. Devabhaktuni, “U-Net and Its Variants for Medical Image Segmentation: A Review of Theory and Applications,” *IEEE Access*, vol. 9, pp. 82 031–82 057, 2021.
- [65] Özgün Çiçek, A. Abdulkadir, S. S. Lienkamp, T. Brox, and O. Ronneberger, “3d U-Net: Learning Dense Volumetric Segmentation from Sparse Annotation,” 2016.
- [66] Z. Zhou, M. M. R. Siddiquee, N. Tajbakhsh, and J. Liang, “UNet++: A Nested U-Net Architecture for Medical Image Segmentation,” 2018.
- [67] Z. Zhang, Q. Liu, and Y. Wang, “Road Extraction by Deep Residual U-Net,” *CoRR*, vol. abs/1711.10684, 2017. [Online]. Available: <http://arxiv.org/abs/1711.10684>
- [68] O. Oktay, J. Schlemper, L. L. Folgoc, M. C. H. Lee, M. P. Heinrich, K. Misawa, K. Mori, S. G. McDonagh, N. Y. Hammerla, B. Kainz, B. Glocker, and D. Rueckert, “Attention U-Net: Learning Where to Look for the Pancreas,” *CoRR*, vol. abs/1804.03999, 2018. [Online]. Available: <http://arxiv.org/abs/1804.03999>
- [69] X. Galindo, “3D deep-learning based quantification of the nuclei distribution for light-sheet microscopy,” Ph.D. dissertation, Université de Bordeaux, 2022, thèse de doctorat dirigée par Levet, Florian Bioinformatique Bordeaux 2022. [Online]. Available: <http://www.theses.fr/2022BORD0275>
- [70] K. He, X. Zhang, S. Ren, and J. Sun, “Deep Residual Learning for Image Recognition,” 2015.
- [71] S. Preibisch, S. Saalfeld, and P. Tomancak, “Globally Optimal Stitching of Tiled 3D Microscopic Image Acquisitions,” *Bioinformatics (Oxford, England)*, vol. 25, pp. 1463–5, 05 2009.

- [72] F. Luisier, C. Vonesch, T. Blu, and M. Unser, “Fast interscale wavelet denoising of Poisson-corrupted images,” *Signal Processing*, vol. 90, no. 2, pp. 415–427, 2010. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0165168409003016>
- [73] T. Peng, K. Thorn, T. Schroeder, L. Wang, F. J. Theis, C. Marr, and N. Navab, “A Basic tool for background and shading correction of optical microscopy images,” *Nature Communications*, vol. 8, no. 1, p. 14836, Jun 2017. [Online]. Available: <https://doi.org/10.1038/ncomms14836>
- [74] P. Bankhead, M. B. Loughrey, J. A. Fernández, Y. Dombrowski, D. G. McArt, P. D. Dunne, S. McQuaid, R. T. Gray, L. J. Murray, H. G. Coleman, J. A. James, M. Salto-Tellez, and P. W. Hamilton, “QuPath: Open source software for digital pathology image analysis,” *Scientific Reports*, vol. 7, no. 1, p. 16878, Dec 2017. [Online]. Available: <https://doi.org/10.1038/s41598-017-17204-5>
- [75] S. Berg, D. Kutra, T. Kroeger, C. N. Straehle, B. X. Kausler, C. Haubold, M. Schiegg, J. Ales, T. Beier, M. Rudy, K. Eren, J. I. Cervantes, B. Xu, F. Beuttenmueller, A. Wolny, C. Zhang, U. Koethe, F. A. Hamprecht, and A. Kreshuk, “ilastik: interactive machine learning for (bio)image analysis,” *Nature Methods*, vol. 16, no. 12, pp. 1226–1232, Dec 2019. [Online]. Available: <https://doi.org/10.1038/s41592-019-0582-9>
- [76] The MathWorks Inc., “MATLAB version: 9.13.0 (R2022b),” Natick, Massachusetts, United States, 2022. [Online]. Available: <https://www.mathworks.com>
- [77] D. Bauer, “napari-nD-annotator: A toolbox for annotating objects one by one in nD,” <https://github.com/bauerdavid/napari-nD-annotator>, 2024, gitHub repository.
- [78] S. N. Rashid and M. M. Fraz, “Nuclei probability and centroid map network for nuclei instance segmentation in histology images,” *Neural Computing and Applications*, vol. 35, no. 21, pp. 15 447–15 460, Jul. 2023.
- [79] B. R. Hunt, “The Application of Constrained Least Squares Estimation to Image Restoration by Digital Computer,” *IEEE Transactions on Computers*, vol. C-22, no. 9, pp. 805–812, 1973.
- [80] Dovahcrow, “Patchify,” [Accessed 20-07-2024]. [Online]. Available: <https://github.com/dovahcrow/patchify.py?tab=readme-ov-file>
- [81] S. S. M. Salehi, D. Erdogmus, and A. Gholipour, “Tversky loss function for image segmentation using 3D fully convolutional deep networks,” *CoRR*, vol. abs/1706.05721, 2017. [Online]. Available: <http://arxiv.org/abs/1706.05721>

- [82] S. Lab, “Connected Components 3D,” 2021, accessed: 2024-07-24. [Online]. Available: <https://github.com/seung-lab/connected-components-3d>
- [83] M. Cole, *Deep Learning with C#, .Net and Kelp.Net: The Ultimate Kelp.Net Deep Learning Guide*. BPB Publications, 2019. [Online]. Available: <https://books.google.cl/books?id=DIaXDwAAQBAJ>
- [84] N. S. Keskar, D. Mudigere, J. Nocedal, M. Smelyanskiy, and P. T. P. Tang, “On Large-Batch Training for Deep Learning: Generalization Gap and Sharp Minima,” 2017.
- [85] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, “Dropout: A Simple Way to Prevent Neural Networks from Overfitting,” *Journal of Machine Learning Research*, vol. 15, no. 56, pp. 1929–1958, 2014. [Online]. Available: <http://jmlr.org/papers/v15/srivastava14a.html>
- [86] P. Everton Gomedé, “Halting in Time: Mastering Early Stopping Techniques for Optimal Machine Learning Performance,” <https://pub.aimind.so/halting-in-time-mastering-early-stopping-techniques-for-optimal-machine-learning-performance-bd7a15239a77>, Apr 2024, Medium, AI Mind, [Accessed 13-06-2024].
- [87] “CS231n Convolutional Neural Networks for Visual Recognition,” <https://cs231n.github.io/neural-networks-3/#accuracy>, [Accessed 13-06-2024].
- [88] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, “Focal Loss for Dense Object Detection,” in *2017 IEEE International Conference on Computer Vision (ICCV)*, 2017, pp. 2999–3007.
- [89] R. Padilla, S. L. Netto, and E. A. B. da Silva, “A Survey on Performance Metrics for Object-Detection Algorithms,” in *2020 International Conference on Systems, Signals and Image Processing (IWSSIP)*, 2020, pp. 237–242.
- [90] K. Ong, S.-C. Haw, and K.-W. Ng, “Deep Learning Based-Recommendation System: An Overview on Models, Datasets, Evaluation Metrics, and Future Trends,” in *Proceedings of the 2019 2nd International Conference on Computational Intelligence and Intelligent Systems*, ser. CIIS '19. New York, NY, USA: Association for Computing Machinery, 2020, p. 6–11. [Online]. Available: <https://doi.org/10.1145/3372422.3372444>
- [91] A. Rosebrock, “Intersection over union (IOU) for object detection,” Jul 2024, [Accessed 08-07-2024]. [Online]. Available: <https://pyimagesearch.com/2016/11/07/intersection-over-union-iou-for-object-detection/>
- [92] D. Müller, I. Soto-Rey, and F. Kramer, “Towards a Guideline for Evaluation Metrics in Medical Image Segmentation,” 2022. [Online]. Available: <https://arxiv.org/abs/2202.05273>

Anexo A Fundamentos en Deep Learning

A.1 Lotes y Épocas

Durante el entrenamiento de una red neuronal, en lugar de enviar toda la entrada de una vez, dividimos aleatoriamente la entrada en un número predefinido de fragmentos de tamaño igual llamados **lotes** (*batch*). Entrenar con datos en lotes hace que el modelo sea más generalizado, ya que al usar tamaños de lote pequeños, el modelo tiende a converger más rápida y eficientemente hacia un mínimo global, gracias a este tamaño reducido [83, 84]. Una iteración de entrenamiento de todos esos lotes tanto en la propagación hacia adelante como hacia atrás se define como una **época** (*epoch*). El número de épocas es crítico, aunque uno esperaría que un número mayor resultara en una mejor precisión de la red, se debe considerar el riesgo de sobreajuste (o *overfitting*) [83].

La **detención temprana** *early stopping*, Fig. A.1, es una técnica utilizada para reducir la posibilidad de tener un modelo sobreajustado sin comprometer su precisión. Esta técnica evalúa el rendimiento de la red en el conjunto de validación después de cada época (o cada N épocas). Si la red supera al mejor modelo guardado anteriormente, se guarda una copia de la red en la época actual. El modelo final es el modelo que tiene el mejor rendimiento en el conjunto de validación [85]. Otra técnica de regularización para prevenir el sobreajuste es el **dropout**, Fig. A.2, que previene el sobreajuste de la red eliminando aleatoriamente y temporalmente algunas de las neuronas de las capas ocultas junto con sus conexiones para cada época en el proceso de entrenamiento.

A.2 Tasa de aprendizaje

La **tasa de aprendizaje** (o *learning rate*), Fig. A.3, es un hiperparámetro que controla cuánto ajustamos los pesos de nuestra red con respecto a la función de pérdida. También podemos definirla como la velocidad a la que descendemos hacia los mínimos de la función de costo. Cuanto menor sea el valor, más lentamente nos desplazamos a lo largo de la pendiente descendente. Debemos elegir la tasa de aprendizaje con mucho cuidado, ya que un valor muy grande aumenta el riesgo de perder la solución óptima, y un valor muy bajo resulta en una convergencia de la red que consume mucho tiempo [41].

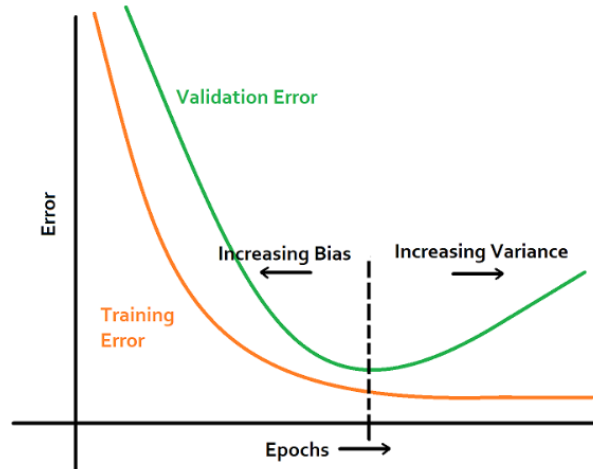


Fig. A.1: Early stopping. Si el rendimiento del modelo en el conjunto de datos de validación comienza a disminuir (por ejemplo, si la pérdida comienza a aumentar o la precisión comienza a disminuir), entonces se detiene el proceso de entrenamiento. El early stopping podría potencialmente mejorar la generalización cuando otros regularizadores están ausentes. Adaptado de [86].

A.3 Función de costo/pérdida

Las **funciones de pérdida** (o *loss function*) se utilizan para medir qué tan bien modela una red neuronal los datos. Por lo tanto, el objetivo es minimizar su valor, ya que calcula la distancia entre la salida de la red y la salida esperada derivada de los datos de entrenamiento [83]. Esto significa que si la función de pérdida tiene un valor muy alto, las predicciones hechas por nuestro modelo serán muy diferentes de los resultados reales [41, 54].

Las funciones de pérdida se dividen en dos categorías: pérdida de regresión, como el **error absoluto medio (MAE)**, que calcula el promedio de la diferencia absoluta entre el valor real y el valor predicho, Ec. A.3.1, y pérdida de clasificación, como la **entropía cruzada binaria (BCE)**, que da el valor de probabilidad entre 0 y 1 para una tarea de clasificación, donde la entropía cruzada calcula la diferencia promedio entre las probabilidades predichas y reales, Ec. A.3.2.

$$\text{MAE} = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i| \quad (\text{A.3.1})$$

Donde en Ec. A.3.1, n es el número total de muestras, y_i son los valores reales observados e \hat{y}_i son los valores predichos por el modelo.

$$\text{BCE} = -\frac{1}{n} \sum_{i=1}^n [y_i \log(\hat{y}_i) + (1 - y_i) \log(1 - \hat{y}_i)] \quad (\text{A.3.2})$$

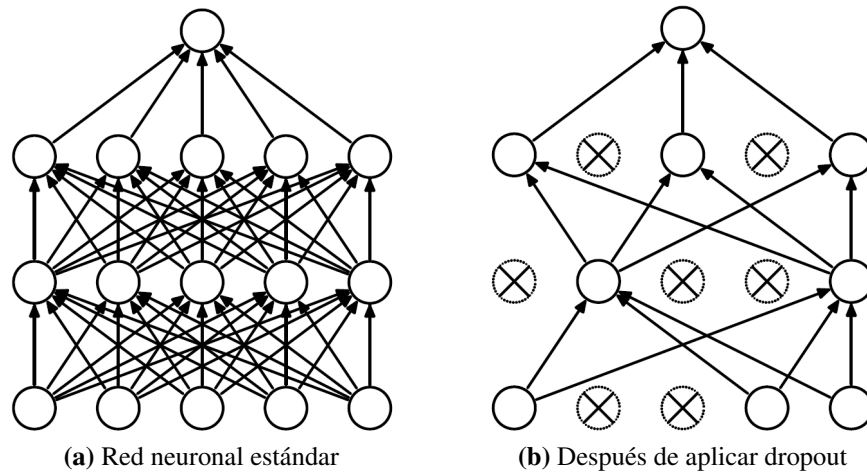


Fig. A.2: Modelo de red neuronal con Dropout. **(a):** Una red neuronal estándar con 2 capas ocultas. **(b):** Un ejemplo de una red reducida producida al aplicar dropout a la red de la izquierda. Las unidades cruzadas han sido eliminadas. Adaptado de [85].

Donde en Ec. A.3.2, n es el número total de muestras, y_i son las etiquetas binarias reales (0 o 1) e \hat{y}_i son las predicciones del modelo, que representan probabilidades.

Además, existen otras funciones de pérdida como la función de **pérdida focal** (FL) [88], que aborda el problema del desequilibrio de clases durante el entrenamiento, especialmente en tareas como la detección de objetos. Esta pérdida modifica la pérdida de entropía cruzada (A.3.2), al agregar un término que enfoca el aprendizaje en los ejemplos mal clasificados y que en términos simples, ajusta automáticamente la influencia de los ejemplos más fáciles durante el entrenamiento, permitiendo que el modelo se enfoque más rápidamente en los ejemplos difíciles y mejorando así el rendimiento general, Fig. A.4.

$$FL(p_t) = -\alpha_t \cdot (1 - p_t)^\gamma \cdot \log(p_t) \quad (\text{A.3.3})$$

Donde en Ec. (A.3.3) p_t es la probabilidad predicha para la clase verdadera, α_t es el peso de clase, que puede ser utilizado para manejar el desbalance de clases, γ es un hiperparámetro (usualmente entre 0 y 5) que modula la facilidad o dificultad de enfocarse en ejemplos de confianza alta y \log es la función logarítmica. Por último, la **pérdida de Tversky** (TL) [81] es una función de pérdida basada en el índice de Tversky, que aborda el problema del desequilibrio de datos y logra un mejor equilibrio entre precisión y recall en el entrenamiento de redes neuronales profundas completamente convolucionales en 3D. Está diseñada para medir la disimilitud entre dos conjuntos, como las máscaras binarias predichas y verdades fundamentales utilizadas en la segmentación. La pérdida de Tversky se introdujo como una extensión del índice Jaccard ampliamente utilizado (también conocido como Intersección sobre Unión, IoU) y el F1-Score.

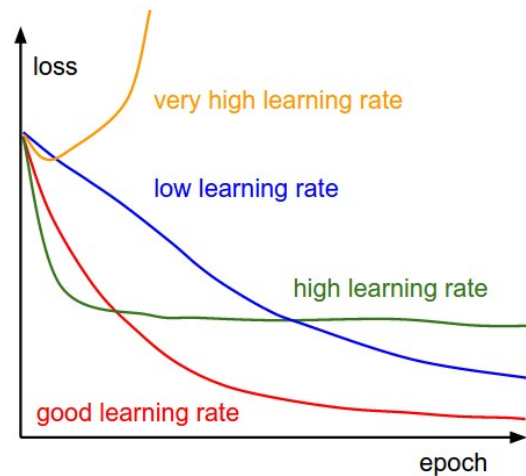


Fig. A.3: La tasa de aprendizaje es un parámetro de ajuste utilizado en el aprendizaje automático que controla qué tan rápidamente se adapta el modelo al problema. Adaptado de [87].

$$\text{Tversky}(A, B) = \frac{|A \cap B|}{|A \cap B| + \alpha|A \setminus B| + \beta|B \setminus A|} \quad (\text{A.3.4})$$

Donde en Ec. A.3.4, A y B son conjuntos o máscaras binarias, $|A \cap B|$ es la cardinalidad (cantidad de elementos) de la intersección entre A y B , $|A \setminus B|$ es la cardinalidad de los elementos en A pero no en B (diferencia de conjuntos), $|B \setminus A|$ es la cardinalidad de los elementos en B pero no en A , α y β son pesos que controlan la importancia relativa de los falsos positivos (elementos en $A \setminus B$) y los falsos negativos (elementos en $B \setminus A$), respectivamente.

A.4 Métricas para la Evaluación del Rendimiento de los Modelos

Evaluar el rendimiento de los modelos de DL es esencial para determinar su precisión. Algunas de las métricas comúnmente utilizadas se basan en el número de verdaderos positivos (TP), falsos positivos (FP) y falsos negativos (FN). Un elemento **verdadero positivo** representa un objeto correctamente predicho para pertenecer a una cierta clase (según la máscara objetivo). Un **falso negativo** es un objeto clasificado incorrectamente como no perteneciente a la clase dada. Un **falso positivo** es un objeto clasificado incorrectamente como perteneciente a una clase. En el caso de la segmentación de imágenes, un TP es una máscara predicha que se superpone a una máscara en la verdad fundamental (GT) con una cantidad superior a un cierto umbral. Un FP es una máscara predicha que no tiene correspondencia en la GT o una correspondencia que está por debajo del umbral establecido. Finalmente, los FN son objetos de la GT que no tienen un objeto correspondiente en la máscara predicha.

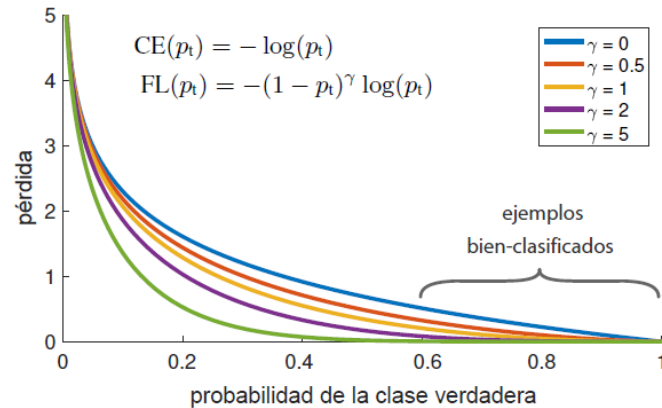


Fig. A.4: Pérdida Focal, la cual añade un factor $(1 - p_t)$ al criterio estándar de entropía cruzada. Estableciendo $\gamma > 0$, se reduce la pérdida relativa para ejemplos bien clasificados ($p_t > 0.5$), poniendo más foco en ejemplos difíciles y mal clasificados. Adaptado de [88].

La **intersección sobre unión** (IoU), Fig. A.5, es una forma estándar de definir si una detección se ha realizado correctamente. La IoU es un coeficiente de similitud entre dos conjuntos de datos y mide la superposición entre los objetos predichos y los objetos de la verdad del terreno [89], clasificando cada objeto como TP, FP o FN según un umbral de IoU previamente establecido t . Si $\text{IoU} \geq t$, entonces la predicción se considera correcta. Por el contrario, si $\text{IoU} < t$, la predicción se considera incorrecta [90].

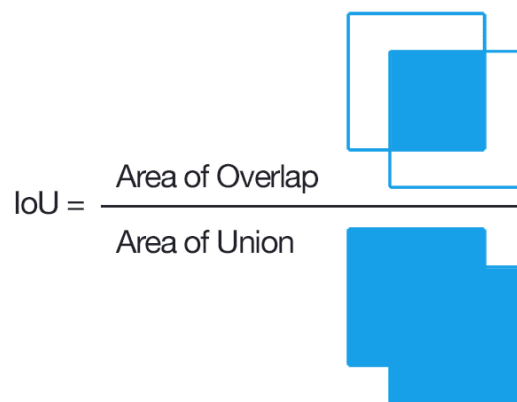


Fig. A.5: Diagrama de intersección sobre unión (IoU). Calcular la intersección sobre la unión es tan simple como dividir el área de superposición entre los cuadros delimitadores por el área de unión. Adaptado de [91].

La evaluación de muchas detecciones de objetos y modelos de segmentación de imágenes se basa principalmente en la precisión y el recall. La **precisión** refleja la proporción de objetos predichos que coinciden correctamente con el GT. El **recall**, por otro lado, calcula la proporción de GT que se identifica correctamente en los objetos predichos. Ambas métricas varían de 0 a 1, donde puntuaciones más altas indican mejores resultados del modelo [89].

$$\text{Precisión} = \frac{TP}{TP + FP} = \frac{\text{Verdaderos positivos}}{\text{Todos los objetos detectados}} \quad (\text{A.4.1})$$

$$\text{Recall} = \frac{TP}{TP + FN} = \frac{\text{Verdaderos positivos}}{\text{Todos los objetos de la verdad fundamental}} \quad (\text{A.4.2})$$

Otra métrica importante es la **medida F**, mejor conocida como **puntuación F (F1-Score)**. Se calcula a partir del recall y la precisión. Calcula la superposición entre la segmentación predicha y la verdad del terreno. Esta métrica es una de las más utilizadas para evaluar el rendimiento de una red neuronal [92]. La puntuación F se da por la siguiente fórmula:

$$\text{F1-Score} = 2 \cdot \frac{\text{Precisión} \cdot \text{Recall}}{\text{Precisión} + \text{Recall}} = \frac{2TP}{2TP + FP + FN} \quad (\text{A.4.3})$$

Por último evaluamos la calidad de la segmentación celular utilizando la **precisión promedio (AP)**, consistente con modelos populares de segmentación como StarDist[1] y Cellpose[3]. Cada máscara de célula predicha se empareja con la máscara de célula de referencia más similar, definida por IoU. Las predicciones en una imagen se evalúan a varios niveles de IoU. Con un IoU más bajo, menos píxeles en una célula predicha deben coincidir con una célula de referencia correspondiente para ser una coincidencia válida, Ec. A.4.4.

$$\text{AP} = \frac{TP}{TP + FP + FN} \quad (\text{A.4.4})$$

La curva de precisión promedio se informa para un conjunto de datos promediando la métrica de precisión promedio para cada imagen en el conjunto de datos. El emparejamiento óptimo de células predichas y de referencia es demasiado exigente computacionalmente en 3D, incluso cuando se restringe la búsqueda a los vecinos más cercanos. En línea con Cellpose, la curva AP se informa para 11 umbrales de IoU que muestrean uniformemente el rango [0.5, 1.0].

Anexo B Evaluación de Resultados por Región del Lobulillo Hepático

	FP	TP	FN	Precision	Recall	AP	F1-Score	Mean IoU True	Mean IoU Matched
MT+WS	56	121	303	0.6836	0.2854	0.2521	0.4027	0.2137	0.7489
Cellpose	228	239	185	0.5118	0.5637	0.3666	0.5365	0.3982	0.7064
Modelo Propuesto	88	276	148	0.7582	0.6509	0.5391	0.7005	0.4711	0.7238

Tabla A.1: Resultados en la región periportal de C3. Los mejores resultados con respecto a cada métrica se muestran en negrita. Utilizamos los falsos positivos (FP), verdaderos positivos (TP), falsos negativos (FN), la precisión, el recall, el puntajes F1 (F1-Score), el promedio IoU normalizado con la verdad fundamental (Mean IoU True) y el promedio IoU de los TP.

	FP	TP	FN	Precision	Recall	AP	F1-Score	Mean IoU True	Mean IoU Matched
MT+WS	13	168	61	0.9282	0.7336	0.6942	0.8195	0.5514	0.7517
Cellpose	23	205	24	0.8991	0.8952	0.8135	0.8972	0.7271	0.8123
Modelo Propuesto	14	211	18	0.9378	0.9214	0.8683	0.9295	0.7562	0.8207

Tabla A.2: Resultados en la región central de C3. Los mejores resultados con respecto a cada métrica se muestran en negrita. Utilizamos los falsos positivos (FP), verdaderos positivos (TP), falsos negativos (FN), la precisión, el recall, el puntajes F1 (F1-Score), el promedio IoU normalizado con la verdad fundamental (Mean IoU True) y el promedio IoU de los TP.

	FP	TP	FN	Precision	Recall	AP	F1-Score	Mean IoU True	Mean IoU Matched
MT+WS	44	133	73	0.7514	0.6456	0.5320	0.6945	0.4721	0.7312
Cellpose	28	185	21	0.8685	0.8981	0.7906	0.8831	0.6939	0.7726
Modelo Propuesto	18	193	13	0.9147	0.9369	0.8616	0.9257	0.7284	0.7775

Tabla A.3: Resultados en la región pericentral de C3. Los mejores resultados con respecto a cada métrica se muestran en negrita. Utilizamos los falsos positivos (FP), verdaderos positivos (TP), falsos negativos (FN), la precisión, el recall, el puntajes F1 (F1-Score), el promedio IoU normalizado con la verdad fundamental (Mean IoU True) y el promedio IoU de los TP.

Modelo	Región Periportal		Región Central		Región Pericentral	
	Mean IoU True	Mean IoU Matched	Mean IoU True	Mean IoU Matched	Mean IoU True	Mean IoU Matched
MT+WS	0.2137	0.7489	0.5514	0.7517	0.4721	0.7312
Cellpose	0.3982	0.7064	0.7271	0.8123	0.6939	0.7726
Modelo Propuesto	0.4711	0.7238	0.7562	0.8207	0.7284	0.7775

Tabla A.4: Resultados de la calidad de la segmentación en C3. El mejor desempeño con respecto a cada métrica está en negrita. Utilizamos el promedio IoU normalizado con la verdad fundamental (Mean IoU True) y el promedio IoU de los verdaderos positivos (Mean IoU Matched), para evaluar la calidad de la segmentación promedio de los núcleos en general y de los núcleos emparejados, respectivamente.

Threshold τ	0.50	0.55	0.60	0.65	0.70	0.75	0.80	0.85	0.90	0.95	1.00
MT+WS	0.2521	0.2392	0.2290	0.1972	0.1738	0.1404	0.0809	0.0186	0.0017	0.0	0.0
Cellpose	0.3666	0.3180	0.2783	0.2358	0.1833	0.1138	0.0557	0.0195	0.0000	0.0	0.0
Modelo Propuesto	0.5391	0.4784	0.4071	0.3401	0.2449	0.1814	0.1273	0.0606	0.0064	0.0	0.0

Tabla A.5: Precisión promedio para varios umbrales de IoU (τ) en la región periportal

Threshold τ	0.50	0.55	0.60	0.65	0.70	0.75	0.80	0.85	0.90	0.95	1.00
MT+WS	0.6942	0.6270	0.5769	0.5299	0.4437	0.3141	0.1919	0.0459	0.0024	0.0	0.0
Cellpose	0.8135	0.7645	0.7443	0.7052	0.6380	0.5597	0.4281	0.2590	0.0603	0.0	0.0
Modelo Propuesto	0.8683	0.8531	0.8233	0.8088	0.7395	0.6509	0.4983	0.2404	0.0225	0.0	0.0

Tabla A.6: Precisión promedio para varios umbrales de IoU (τ) en la región central

Threshold τ	0.50	0.55	0.60	0.65	0.70	0.75	0.80	0.85	0.90	0.95	1.00
MT+WS	0.5320	0.4291	0.4081	0.3392	0.2939	0.2276	0.1265	0.0609	0.0079	0.0	0.0
Cellpose	0.7906	0.7679	0.7243	0.6561	0.5181	0.3874	0.2545	0.1203	0.0096	0.0	0.0
Modelo Propuesto	0.8616	0.8370	0.7897	0.6951	0.5916	0.4330	0.3113	0.1332	0.0048	0.0	0.0

Tabla A.7: Precisión promedio para varios umbrales de IoU (τ) en la región pericentral

UNIVERSIDAD DE CONCEPCION – FACULTAD DE INGENIERIA
RESUMEN DE MEMORIA DE TITULO

Departamento	: Departamento de Ingeniería Eléctrica
Carrera	: Ingeniería Civil Biomédica
Nombre del memorista	: Jorge Andrés Vergara Estrada Generación de Base de Datos y Algoritmo de Segmentación
Título de la memoria	: Automática de Núcleos Hepáticos en Imágenes 3D de Microscopía mediante Redes Neuronales
Fecha de la presentación oral	: 28 de Agosto del 2024 Dr. Pamela Beatriz Guevara Alvez
Profesor(es) Guía	: Dr. Fabián Josue Segovia Miranda Dr. Hernán Morales Navarrete
Profesor(es) Revisor(es)	: Dr. Sebastián Eugenio Godoy Medel
Concepto	:
Calificación	:

Resumen

La comprensión de la organización de órganos y la coordinación celular es fundamental en biología celular y del desarrollo. La histología de tejidos es crucial para el diagnóstico de patologías como el cáncer. Sin embargo, las evaluaciones histopatológicas en 2D pierden el contexto tridimensional (3D) del tejido. El hígado, con múltiples funciones vitales, depende de una arquitectura 3D compleja, por lo que es esencial conocer la información espacial exacta de los núcleos y células para comprender sus funciones e interacciones. Para superar estas limitaciones, se utilizan modelos geométricos y software como Motion Tracking (MT) para generar reconstrucciones 3D de alta resolución. La segmentación de núcleos celulares enfrenta desafíos, especialmente en núcleos muy cercanos. Las redes neuronales convolucionales (CNN) son efectivas para procesar y analizar datos visuales. La Residual Attention U-Net 3D, una variante avanzada de CNN, mejora la precisión de la segmentación mediante bloques residuales y mecanismos de atención. Se generó una base de datos de núcleos 3D y se entrenó la red neuronal con un enfoque multitarea, realizando segmentación binaria y predicción de centroides. Se segmentaron aproximadamente 2900 núcleos 3D en tres sujetos de control (C1, C2 y C3). Nuestro modelo superó significativamente a MT y ligeramente a Cellpose, especialmente en la región periportal, demostrando su capacidad en regiones complejas del tejido hepático.