



**UNIVERSIDAD DE CONCEPCIÓN
FACULTAD DE INGENIERÍA
DEPARTAMENTO DE INGENIERÍA ELÉCTRICA**



CLASIFICACIÓN DE NÚCLEOS CELULARES EN 3D BASADA EN DEEP LEARNING PARA LA RECONSTRUCCIÓN DIGITAL DE TEJIDO HEPÁTICO

POR

Jean Paul Taylor Muñoz

Informe Final 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
Dra. Pamela Guevara A.
Dr. Fabián Segovia M.
Dr. Hernán Morales N.

Comisión
Dra. Rosa Figueroa I.

Julio 08, 2024
Concepción
(Chile)

© 2024 Jean Paul Taylor Muñoz

© 2024 Jean Paul Taylor Muñoz

Cualquier parte de esta memoria puede reproducirse o transmitirse por cualquier medio o procedimiento, siempre y cuando se cite adecuadamente al autor.

Agradecimientos

En primer lugar, quiero agradecer profundamente a mis padres por su inquebrantable apoyo y orientación a lo largo de todo este proceso. Sin ellos, no habría llegado hasta aquí.

También quiero expresar mi profundo agradecimiento a mi novia, quien ha sido mi sostén tanto en los momentos buenos como en los difíciles, siendo un pilar fundamental en este camino hacia el éxito.

Por último, deseo dar las gracias a todos los integrantes del laboratorio. Su compañerismo y colaboración han hecho de esta experiencia algo verdaderamente gratificante y enriquecedor.

Resumen

El hígado, crucial para la digestión y el sistema circulatorio, contiene diversos tipos de células con variaciones morfológicas y nucleares. Comprender su funcionamiento a nivel celular es esencial para el tratamiento de enfermedades hepáticas, por lo que es necesario contar con un sistema que permita diferenciar estos tipos celulares.

A pesar de los desafíos recientes, la informática ha permitido avances significativos en biología celular, como la creación de modelos 3D del lobulillo hepático a partir de imágenes de microscopía confocal. Aunque estos modelos 3D son prometedores, presentan limitaciones como la dependencia de múltiples tipos de imágenes para una clasificación precisa y la clasificación errónea debido a las limitaciones de los algoritmos actuales. En este contexto, una estrategia prometedora es el uso de Deep Learning (DL) con Redes Neuronales Convolucionales (CNN), que ha demostrado éxito en clasificación en biología e imágenes médicas. El DL puede superar las limitaciones actuales al reducir la dependencia de múltiples imágenes y al identificar patrones complejos para una clasificación más precisa.

El objetivo principal de esta memoria es utilizar Deep Learning (DL) para clasificar diferentes tipos de núcleos hepáticos en 3D mediante el entrenamiento de Redes Neuronales Convolucionales (CNN). Se creó un Ground Truth (GT) en 3D a partir de imágenes de microscopía confocal y se entrenaron tres CNN reconocidas adaptadas a 3D (ResNet50, InceptionV3 e Inception-ResNet-v2). Se evaluaron distintos prototipos de clasificación, tanto multicategoría como binarios, usando métricas como F1-Score, Recall, Accuracy y Precision para seleccionar el mejor modelo.

Como resultado, se obtuvo un algoritmo robusto para clasificar diferentes tipos de núcleos hepáticos, con resultados comparables o incluso superiores a los del software MotionTracking (MT). Se desarrollaron modelos binarios específicos para distintos tipos de núcleos hepáticos, destacando el modelo para hepatocitos, que logró una correcta clasificación del 95 % de estos núcleos tanto en tejido hepático normal como en tejido en regeneración.

En resumen, la investigación subraya la importancia de mejorar la reconstrucción en 3D del tejido hepático con menos inmunomarcadores. Esto permite una clasificación más precisa de los núcleos hepáticos, facilita el estudio de otras estructuras biológicas en el hígado, optimiza recursos y tiempos de laboratorio, y reduce la dependencia de marcadores específicos, promoviendo investigaciones más detalladas sobre las complejidades del tejido hepático.

Abstract

The liver, crucial for digestion and the circulatory system, contains various types of cells with morphological and nuclear variations. Understanding its cellular function is essential for treating liver diseases, thus a system that allows differentiation between these cell types is necessary.

Despite recent challenges, computing has enabled significant advances in cellular biology, such as the creation of 3D models of the liver lobule from confocal microscopy images. Although these 3D models are promising, they have limitations, such as dependence on multiple types of images for accurate classification and misclassification due to current algorithm limitations. In this context, a promising strategy is the use of Deep Learning (DL) with Convolutional Neural Networks (CNN), which have shown success in classification in biology and medical imaging. DL can overcome current limitations by reducing the dependency on multiple images and identifying complex patterns for more accurate classification.

The main objective of this thesis is to use Deep Learning (DL) to classify different types of liver nuclei in 3D by training Convolutional Neural Networks (CNNs). A 3D Ground Truth (GT) was created from confocal microscopy images, and three well-known CNNs adapted to 3D (ResNet50, InceptionV3, and Inception-ResNet-v2) were trained. Various classification prototypes, both multi-category and binary, were evaluated using metrics such as F1-Score, Recall, Accuracy, and Precision to select the best model.

As a result, a robust algorithm was developed to classify different types of liver nuclei, with results comparable to or even exceeding those of the MotionTracking (MT) software. Specific binary models were developed for different types of liver nuclei, with the hepatocyte model achieving a correct classification rate of 95 % for these nuclei in both normal liver tissue and regenerating tissue.

In summary, the research highlights the importance of improving 3D reconstruction of liver tissue with fewer immunomarkers. This allows for more accurate classification of liver nuclei, facilitates the study of other biological structures in the liver, optimizes laboratory resources and time, and reduces dependence on specific markers, promoting more detailed investigations into the complexities of liver tissue.

Tabla de contenidos

CAPITULO 1. INTRODUCCIÓN.....	10
1.1 INTRODUCCIÓN GENERAL	10
1.2 OBJETIVOS	11
1.2.1 <i>Objetivo General</i>	11
1.2.2 <i>Objetivos Específicos</i>	11
1.3 ALCANCES Y LIMITACIONES	12
1.4 METODOLOGÍA	12
1.4.1 <i>Recolección de Muestras e Inmunotinción</i>	12
1.4.2 <i>Obtención de las Imágenes</i>	13
1.4.3 <i>Preprocesamiento</i>	13
1.4.4 <i>Generación del Ground Truth</i>	14
1.4.5 <i>Definición Conjuntos de Datos</i>	14
1.4.6 <i>Entrenamiento de los Modelos</i>	14
1.4.7 <i>Evaluación de los Modelos y Fine-Tuning</i>	15
1.5 TEMARIO	15
CAPITULO 2. MARCO TEÓRICO	17
2.1 INTRODUCCIÓN	17
2.2 TEJIDO HEPÁTICO	17
2.2.1 <i>Función del hígado</i>	17
2.2.2 <i>Estructura del hígado</i>	17
2.2.3 <i>Células presentes en el hígado</i>	19
2.3 DEEP LEARNING	19
2.3.1 <i>Red Neuronal Convolucional</i>	20
2.3.2 <i>Arquitectura CNN</i>	20
2.3.2.1 <i>Capa de convolución</i>	21
2.3.2.2 <i>Capa de agrupación</i>	21
2.3.2.3 <i>Capa densa</i>	22
2.3.2.4 <i>Funciones de activación</i>	22

2.3.2.5	Entrenamiento de una CNN	23
2.3.2.6	Técnicas de regularización	25
2.4	MÉTRICAS DE EVALUACIÓN	25
2.4.1	<i>Precision (Precisión)</i>	26
2.4.2	<i>Recall (Sensibilidad)</i>	26
2.4.3	<i>F1-Score</i>	27
2.4.4	<i>Accuracy (Exactitud)</i>	27
2.4.5	<i>Matriz de confusión</i>	27
2.5	TRABAJOS PREVIOS	28
2.5.1	<i>Napari</i>	28
2.5.2	<i>MotionTracking</i>	29
2.5.3	<i>ResNet50</i>	29
2.5.4	<i>InceptionV3</i>	30
2.5.5	<i>Inception-ResNet-v2</i>	31
2.5.6	<i>Trabajos previos en el área de interés</i>	33
2.6	DISCUSIÓN	33
CAPITULO 3. BASE DE DATOS Y GENERACIÓN DEL GROUND TRUTH		35
3.1	INTRODUCCIÓN	35
3.2	IMÁGENES A UTILIZAR	35
3.3	GENERACIÓN DEL GROUND TRUTH	36
3.3.1	<i>Etapas de segmentación</i>	37
3.3.2	<i>Etapas de clasificación</i>	38
3.3.3	<i>Análisis cuantitativo</i>	39
3.4	DISCUSIÓN	40
CAPITULO 4. IMPLEMENTACIÓN		41
4.1	INTRODUCCIÓN	41
4.2	PROCEDIMIENTO PARA LA IMPLEMENTACIÓN DE LOS MODELOS DE CLASIFICACIÓN.	41
4.2.1	<i>Definición conjuntos de entrenamiento, validación y prueba</i>	41
4.2.2	<i>Preprocesamiento</i>	42
4.2.3	<i>Implementación de los modelos de clasificación</i>	43
4.2.3.1	Aumentación de datos	44
4.2.3.2	Primer prototipo de los modelos	45
4.2.3.3	Nuevo conjunto de entrenamiento	45

4.2.3.4	Segundo prototipo de los modelos	46
4.2.3.5	Submodelos de clasificación binaria	47
4.2.3.6	Prototipo final	47
4.3	DISCUSIÓN	48
CAPITULO 5. PREDICCIONES Y RESULTADOS.....		49
5.1	INTRODUCCIÓN	49
5.2	PRIMER PROTOTIPO.....	49
5.2.1	<i>Discusión</i>	50
5.3	SEGUNDO PROTOTIPO	50
5.3.1	<i>Discusión</i>	52
5.4	MODELOS BINARIOS	52
5.4.1	<i>Modelo Binario de Núcleos de Hepatocitos</i>	52
5.4.2	<i>Modelo Binario de Núcleos de Células Estrelladas</i>	53
5.4.3	<i>Modelo Binario de Núcleos de Células de Kupffer</i>	54
5.4.4	<i>Modelo Binario de Núcleos de Células Endoteliales</i>	55
5.4.5	<i>Discusión</i>	56
5.5	PROTOTIPO FINAL	57
5.5.1	<i>Discusión</i>	59
5.6	COMPARACIÓN CON MOTIONTRACKING	60
5.6.1	<i>Discusión</i>	61
5.7	MUESTRA DE REGENERACIÓN HEPÁTICA	61
5.7.1	<i>Discusión</i>	62
5.8	DISCUSIÓN	63
CAPITULO 6. CONCLUSIONES.....		64
6.1	DISCUSIÓN	64
6.2	CONCLUSIONES	65
6.3	TRABAJOS FUTUROS	66
CAPÍTULO 7. GLOSARIO		68
CAPITULO 8. REFERENCIAS		69
ANEXO A - RESULTADOS PRIMER PROTOTIPO		74
ANEXO B - RESULTADOS SEGUNDO PROTOTIPO.....		79

ANEXO C - RESULTADOS MODELOS BINARIOS	84
C.1 MODELO BINARIO DE NÚCLEOS DE HEPATOCITOS	84
C.2 MODELO BINARIO DE NÚCLEOS DE CÉLULAS ESTRELLADAS.....	88
C.3 MODELO BINARIO DE NÚCLEOS DE CÉLULAS DE KUPFFER	93
C.4 MODELO BINARIO DE NÚCLEOS DE CÉLULAS ENDOTELIALES	98
ANEXO D - RESULTADOS PROTOTIPO FINAL Y COMPARACIÓN MT	104
D.1 PROTOTIPO FINAL	104
D.2 COMPARACIÓN MT	105

Índice de Tablas

1.1	Anticuerpos, colorantes y reactivos utilizados para tinción y experimentos.	13
3.2	Frecuencia de clase por región y muestra.	40
4.3	Distribución de clases por conjunto.	42
4.4	Distribución de clases para el nuevo conjunto de entrenamiento y sus subconjuntos balanceados.	46
5.5	Métricas de calidad para los modelos de clasificación multiclase de núcleos hepáticos.	51
5.6	Métricas de calidad para los diferentes modelos de clasificación de binaria de núcleos hepáticos en el conjunto de prueba.	56
5.7	Métricas de calidad para las diferentes clasificaciones finales en el conjunto de prueba.	59
5.8	Métricas de calidad para las diferentes clasificaciones en la muestra nueva.	61
5.9	Rendimiento por predicciones positivas y negativas del modelo binario de núcleos de hepatocitos en la muestra de regeneración hepática.	62

Índice de Figuras

2.1	Lobulillo hepático y su composición.	18
2.2	Reconstrucción en 3D del eje CV-PV.	18
2.3	Arquitectura básica de una red neuronal convolucional.	20
2.4	Ejemplo de la operación de convolución.	21
2.5	Ejemplo de operaciones Max Pooling y Average Pooling.	22
2.6	Matriz de Confusión para 5 Clases.	28
2.7	Matriz de confusión para clasificación binaria.	28
2.8	Arquitectura ResNet50.	30
2.9	Bloque residual.	30
2.10	Arquitectura InceptionV3.	31
2.11	Arquitectura Inception-ResNet-v2 y de su bloque Stem.	32
2.12	Arquitectura de los bloques Inception-ResNet y Reduction.	32
3.1	Ubicación relativa de extracción de núcleos para cada muestra.	36
3.2	Ground Truth de segmentación en 3D para un cubo visualizado en Napari.	37
3.3	Canales utilizados para clasificar junto con sus tinciones.	38
3.4	Ground Truth de clasificación de núcleos en 3D para un cubo visualizado en Napari.	39
4.1	Diagrama del preprocesamiento utilizado para entrenar las CNN.	43
4.2	Diagrama básico de los modelos de clasificación.	44
4.3	Imagen de un núcleo de hepatocito junto con el resultado de las operaciones de aumentación de datos.	45
5.1	Algoritmo final de la clasificación de núcleos hepáticos.	59
A.1	Curvas de pérdida y F1-Score del modelo de 5 clases con ResNet50.	74
A.2	Curvas de pérdida y F1-Score del modelo de 5 clases con InceptionV3.	75
A.3	Curvas de pérdida y F1-Score del modelo de 5 clases con Inception-ResNet-v2.	75
A.4	Matriz de confusión en el conjunto de prueba con el modelo de 5 clases con ResNet50.	76
A.5	Matriz de confusión en el conjunto de prueba con el modelo de 5 clases con Incep- tionV3.	77

A.6	Matriz de confusión en el conjunto de prueba con el modelo de 5 clases con Inception-ResNet-v2.	78
B.1	Curvas de pérdida y F1-Score del modelo de 4 clases con ResNet50.	79
B.2	Curvas de pérdida y F1-Score del modelo de 4 clases con InceptionV3.	79
B.3	Curvas de pérdida y F1-Score del modelo de 4 clases con Inception-ResNet-v2.	80
B.4	Matriz de confusión en el conjunto de prueba con el modelo de 4 clases con ResNet50.	81
B.5	Matriz de confusión en el conjunto de prueba con el modelo de 4 clases con InceptionV3.	82
B.6	Matriz de confusión en el conjunto de prueba con el modelo de 4 clases con Inception-ResNet-v2.	83
C.1	Curvas de pérdida y exactitud binaria del modelo binario de núcleos de hepatocitos con ResNet50.	84
C.2	Curvas de pérdida y exactitud binaria del modelo binario de núcleos de hepatocitos con InceptionV3.	85
C.3	Curvas de pérdida y exactitud binaria del modelo binario de núcleos de hepatocitos con Inception-ResNet-v2.	85
C.4	Matriz de confusión en el conjunto de prueba con el modelo binario de núcleos de hepatocitos con ResNet50.	86
C.5	Matriz de confusión en el conjunto de prueba con el modelo binario de núcleos de hepatocitos con InceptionV3.	87
C.6	Matriz de confusión en el conjunto de prueba con el modelo binario de núcleos de hepatocitos con Inception-ResNet-v2.	88
C.7	Curvas de pérdida y exactitud binaria del modelo binario de núcleos de células estrelladas con ResNet50.	89
C.8	Curvas de pérdida y exactitud binaria del modelo binario de núcleos de células estrelladas con InceptionV3.	89
C.9	Curvas de pérdida y exactitud binaria del modelo binario de núcleos de células estrelladas con Inception-ResNet-v2.	90
C.10	Matriz de confusión en el conjunto de prueba con el modelo binario de núcleos de células estrelladas con ResNet50.	91
C.11	Matriz de confusión en el conjunto de prueba con el modelo binario de núcleos de células estrelladas con InceptionV3.	92
C.12	Matriz de confusión en el conjunto de prueba con el modelo binario de núcleos de células estrelladas con Inception-ResNet-v2.	93

C.13	Curvas de pérdida y exactitud binaria del modelo binario de núcleos de células de Kupffer con ResNet50.	94
C.14	Curvas de pérdida y exactitud binaria del modelo binario de núcleos de células de Kupffer con InceptionV3.	94
C.15	Curvas de pérdida y exactitud binaria del modelo binario de núcleos de células de Kupffer con Inception-ResNet-v2.	95
C.16	Matriz de confusión en el conjunto de prueba con el modelo binario de núcleos de células de Kupffer con ResNet50.	96
C.17	Matriz de confusión en el conjunto de prueba con el modelo binario de núcleos de células de Kupffer con InceptionV3.	97
C.18	Matriz de confusión en el conjunto de prueba con el modelo binario de núcleos de células de Kupffer con Inception-ResNet-v2.	98
C.19	Curvas de pérdida y exactitud binaria del modelo binario de endoteliales con ResNet50.	99
C.20	Curvas de pérdida y exactitud binaria el modelo binario de núcleos de células endoteliales con InceptionV3.	99
C.21	Curvas de pérdida y exactitud binaria el modelo binario de núcleos de células endoteliales con Inception-ResNet-v2.	100
C.22	Matriz de confusión en el conjunto de prueba con el modelo binario de núcleos de células endoteliales con ResNet50.	101
C.23	Matriz de confusión en el conjunto de prueba con el modelo binario de núcleos de células endoteliales con InceptionV3.	102
C.24	Matriz de confusión en el conjunto de prueba con el modelo binario de núcleos de células endoteliales con Inception-ResNet-v2.	103
D.1	Matriz de confusión de la predicción optimizada del modelo final de 4 clases en el conjunto de prueba.	104
D.2	Matriz de confusión de la predicción de los modelos binarios finales combinados en el conjunto de prueba.	105
D.3	Matriz de confusión de la clasificación de MotionTracking en la muestra nueva. . .	106
D.4	Matriz de confusión de la predicción modelo final de 4 clases en la muestra nueva. .	107
D.5	Matriz de confusión de la predicción optimizada del modelo final de 4 clases en la muestra nueva.	108
D.6	Matriz de confusión de la predicción de los modelos binarios finales combinados en la muestra nueva.	109

Capítulo 1. Introducción

1.1 Introducción General

Entender cómo funcionan los tejidos y comprender la interacción entre las células de estos es un proceso complejo y que requiere mucho trabajo, esto inclusive para las células de formas comunes y que se organizan entre sí de una forma uniforme u ordenada. La histología, mediante la observación por microscopía óptica de cortes finos (5-10 μm) en dos dimensiones (2D), permite descifrar las estructuras celulares y cómo se organizan dentro de la célula. Si bien permite obtener información valiosa, en casos donde los tejidos están conformados por estructuras complejas en tres dimensiones (3D) que se relacionan entre sí, como el tejido hepático, este método tradicional no permite obtener información tan valiosa [1].

El tejido hepático es esencial y multifuncional, participando tanto en el metabolismo como en la producción de proteínas [2]. Está compuesto por diferentes tipos de células que interactúan entre sí, como los hepatocitos, que forman los ductos biliares por donde se libera la bilis [3], y las células estrelladas, que juegan un papel importante en la regulación del flujo sanguíneo en los sinusoides [4]. Además, las células de Kupffer se encargan de eliminar sustancias extrañas o peligrosas de la sangre portal [4], mientras que las células endoteliales desempeñan un papel crucial en la regulación de lípidos y vitaminas [5]. Dado que el tejido hepático es multifuncional y sus células interactúan estrechamente, estudiar su funcionamiento en 2D resulta ineficiente. La orientación de los hepatocitos puede limitar el estudio de ciertas estructuras y sus interacciones con otras células dentro del tejido hepático.

En este contexto, para entender el correcto funcionamiento de diferentes situaciones y enfermedades presentes en el tejido hepático, es necesario realizar un estudio adecuado no solo de las células presentes en el tejido, sino también de los núcleos. Una de las herramientas que permite realizar este análisis en 3D es el software MotionTracking [1]. A fines del año 2015, se presentó MT como un método versátil para la reconstrucción digital en 3D de tejidos. Este software cuenta con algoritmos para preprocesamiento, segmentación, clasificación y reconstrucción en 3D. MT ha sido una herramienta clave y se ha utilizado con gran éxito para comprender el tejido hepático [6] [7]. Gracias a sus capacidades avanzadas, MT permite obtener información estructural, estadística y organizacional de cada célula, núcleo y componente del tejido [1] [6].

Aunque MT produce resultados de clasificación de alta calidad, enfrenta desafíos significativos al clasificar solo con la imagen de los núcleos. Para lograr una clasificación precisa, el software necesita

imágenes adicionales de otros componentes del tejido hepático. Además, el algoritmo de clasificación utilizado está optimizado para imágenes que no han sido sometidas al preprocesamiento actual aplicado a las imágenes de microscopía, lo que resulta en etiquetas erróneas para diversos núcleos. El software también requiere múltiples imágenes diferentes a las del núcleo para realizar una clasificación adecuada, lo que limita el número de estructuras que se pueden estudiar en el tejido. Otra limitación de MT es que la clasificación se basa en características manuales, y el modelo implementado para la clasificación, LDA, es bastante simple. Estas dos limitaciones pueden hacer que se omitan características importantes para una clasificación precisa de los distintos tipos de núcleos en el tejido hepático [1]. Para superar estas limitaciones, se propone implementar un algoritmo basado en DL que realice la clasificación de los núcleos hepáticos utilizando únicamente la imagen de los núcleos.

El DL ofrece una base sólida para la clasificación al reconocer patrones y representar características jerárquicas [8], evitando los desafíos asociados con modelos probabilísticos como los utilizados en MT. Se plantea desarrollar un algoritmo de DL para mejorar la clasificación de núcleos hepáticos con el objetivo de lograr un método eficiente en 3D utilizando un GT como referencia y únicamente la imagen de los núcleos.

Esta estrategia, combinada con el software MT, tiene como objetivo generar una reconstrucción digital precisa y de alta calidad del tejido hepático, respaldando así la reconstrucción tridimensional del tejido hepático en el ámbito de la investigación biomédica.

1.2 Objetivos

1.2.1 Objetivo General

Clasificar los núcleos de diferentes tipos de células presentes en imágenes de microscopia confocal con inmunotinción del tejido hepático utilizando Deep Learning para la reconstrucción digital en 3D del tejido hepático.

1.2.2 Objetivos Específicos

- Generar un conjunto de imágenes de núcleos etiquetados de forma manual de referencia (Ground Truth), utilizando el software Napari.
- Implementar, adaptar, entrenar y evaluar modelos de clasificación de núcleos para etiquetar núcleos

hepáticos en imágenes segmentadas, incluyendo tanto la identificación del tipo de núcleo como su ubicación espacial.

- Evaluar el rendimiento de los modelos de clasificación mediante métricas de rendimiento, ajustar los hiperparámetros para optimizar el desempeño y comparar el mejor modelo con las clasificaciones generadas por el software MotionTracking, así como con una muestra anómala de tejido hepático.

1.3 Alcances y Limitaciones

- i. Se colaborará con el laboratorio de Cell and Tissue Architecture Lab, bajo la co-supervisión del profesor Fabián Segovia. Los alumnos del laboratorio procesarán las imágenes de microscopía y aplicarán una deconvolución para mejorar la calidad de los datos.
- ii. Aunque la cantidad de cada tipo de núcleo en las imágenes a utilizar no es insuficiente, la notable disparidad en la cantidad de núcleos de cada tipo puede resultar en una pérdida de generalidad del modelo.
- iii. Se utilizará software y hardware de alto rendimiento del laboratorio de Cell and Tissue Architecture Lab, para lograr una implementación y optimización adecuadas.
- iv. El rendimiento del modelo dependerá en gran parte de la calidad de la segmentación de los núcleos.

1.4 Metodología

Las imágenes utilizadas en esta memoria fueron obtenidas con microscopía confocal de inmunotinción del hígado de ratones C57BL/6J de diferentes edades: un ratón de 30 días postnatales, muestra 1), y dos ratones adultos de 8 semanas y 12 semanas, muestra 2 y 3, respectivamente. Aspectos asociados a la ética y obtención de los ratones se pueden consultar en [6].

1.4.1 Recolección de Muestras e Inmunotinción

Para obtener las muestras de cada sujeto se fijaron sus hígados mediante perfusión intracardiaca con paraformaldehído al 4 % en solución salina tamponada con fosfato (PBS) conteniendo un 0.1 %

de Tween20, y posteriormente se fijaron durante la noche con la misma solución a temperatura ambiente. Se obtuvieron cortes de $100\ \mu\text{m}$ para cada sujeto con un vibratome. La inmunomarcación Tabla 1.1 y el aclaramiento óptico fueron efectuados con SeeDB [9] como se describe en [10].

Tabla 1.1: Anticuerpos, colorantes y reactivos utilizados para tinción y experimentos.

Tipo de reactivo (especie) o recurso	Designación	Fuente o referencia	Identificadores	Dilución o concentración	Descripción
Anticuerpos primarios	anti-Flk1 (policlonal de cabra)	R&D System	AF644	(1:100)	Marcador sinusoidal
	anti-CD13 (monoclonal de rata)	Novus	NB100-64843	(1:500)	Marcador de canalículos
	anti-F4/80 (monoclonal de rata)	Abcam	ab6640	(1:400)	Marcador de células F4/80 positivas
	anti-Desmin (policlonal de conejo)	Abcam	ab15200	(1:200)	Marcador de células estrelladas
Anticuerpos secundarios	Asno anti-Cabra Alexa Fluor 647	Invitrogen	A-21447	(1:1000)	
	Asno anti-Rata CF 568	Biotium	20092	(1:1000)	
	Asno anti-Conejo Alexa Fluor Plus 647	Invitrogen	A32795	(1:1000)	
	Faloidina Alexa Fluor 647	Invitrogen	A22287	(1:100)	Borde celular
Colorantes pequeños	Faloidina Alexa Fluor 488	Invitrogen	A12379	(1:100)	Borde celular
	DAPI (4',6-Diamidino-2-Fenilindol, Dihidrocloreuro)	Invitrogen	D1306	(1:1000)	Núcleos
Reactivo	Clodronate y control (PBS) liposoma	LIPOSOMA	CP-010-010	(5mg/ml)	

Fuente: Obtenida del material suplementario en [6].

1.4.2 Obtención de las Imágenes

Las muestras de hígado de cada sujeto fueron capturadas en un microscopio de barrido láser multiphotón invertido (Zeiss LSM 780) con un tamaño de voxel de $0.3\ \mu\text{m}$, utilizando un objetivo multi-inmersión de apertura numérica 1.2 de 40x (Zeiss). El 4',6-diamidino-2-fenilindol (DAPI) fue excitado a 780 nm utilizando un láser de dos fotones Chameleon Ti-Sapphire. Los fluorocromos Alexa Fluor 488, 555 y 657 fueron excitados con líneas láser de 488, 561 y 633 nm, respectivamente, y detectados con detectores de fosforo de arseniuro de galio (GaAsp). Se capturó un eje Vena Central (CV) - Vena Porta (PV) de un lobulillo aleatorio por ratón ($500 \times 250 \times 100\ \mu\text{m}$) [6].

1.4.3 Preprocesamiento

Para abarcar toda la zona de interés, el eje CV-PV, se realizaron múltiples capturas del tejido organizadas en mosaicos 2×1 que fueron superpuestos para generar la imagen final utilizando el complemento Image Stitching de Fiji [11]. A la imagen resultante se le aplicaron una serie de preprocesamientos. En primer lugar, se redujo el ruido utilizando el algoritmo PURE-LET [12] con el máximo número de ciclos; luego, se realizó una corrección del fondo y el sombreado a lo largo del conjunto con la herramienta BaSiC [13]. Posteriormente, se alinearon los canales utilizando una referencia y la herramienta Correct 3D Drift de Fiji [11]. Finalmente, como un paso extra, se aplicó una operación de deconvolución para corregir la anisotropía de la imagen con MATLAB R2022b.

1.4.4 Generación del Ground Truth

Tras el preprocesamiento adecuado de las imágenes, se generó un GT de los núcleos segmentados de forma manual utilizando el software Napari [14] y el complemento napari-nD-annotator [15]. Una descripción más detallada de todo este proceso se presenta en capítulos posteriores. Además, utilizando las imágenes de otras estructuras de interés, se generó un GT manual de los núcleos clasificados en cinco clases distintas: hepatocitos, células estrelladas, células de Kupffer, células endoteliales y una categoría "sin clasificar". Esta categoría hace referencia a los núcleos de colangiocitos presentes en los ductos biliares, así como los núcleos de células presentes en las sinusoides que no son células endoteliales ni de Kupffer. Este GT manual no se creó utilizando la totalidad de las imágenes de los sujetos. En su lugar, se seleccionaron tres cubos de dimensiones: 280 píxeles en el eje z y 320 píxeles en los ejes x e y. Estos cubos se extrajeron de diferentes partes del eje CV-PV para garantizar su representatividad. Se seleccionó uno perteneciente a la zona pericentral, abarcando parte de la vena central; otro perteneciente al centro del tejido; y uno final perteneciente a la zona portocentral, pero enfocado principalmente en abarcar los ductos biliares. Estos cubos fueron extraídos de los tres sujetos, siendo el último el único que carece de ducto biliar.

1.4.5 Definición Conjuntos de Datos

Los conjuntos de entrenamiento y prueba se definieron utilizando el GT generado. En el conjunto de entrenamiento se incluyeron las muestras 1 y 2, mientras que la muestra 3 se reservó exclusivamente para probar los modelos. Esta distribución se seleccionó porque la muestra 3 representa un estadio intermedio entre las otras muestras. Además, las muestras 1 y 2 contienen una mayor cantidad de núcleos y la presencia del ducto biliar, lo cual se considera crucial para la generalización de los modelos.

1.4.6 Entrenamiento de los Modelos

Con estos conjuntos se llevará a cabo el entrenamiento y prueba de los modelos de CNN destacados en la literatura para la clasificación de imágenes, como lo son ResNet50 [16], InceptionV3 [17] e Inception-ResNet-v2 [18]. Este entrenamiento se realizará con Python 3.9 [19] y TensorFlow 2.13.1 [20] en un computador de sistema operativo Windows con una GPU Nvidia GeForce RTX 3090 con 25 GB de memoria, una CPU AMD Ryzen Threadripper 3960X de 24 núcleos - 3.80 GHz y 256 GB de memoria RAM.

1.4.7 Evaluación de los Modelos y Fine-Tuning

Se evaluarán los modelos utilizando métricas de rendimiento como la exactitud, la sensibilidad, la precisión, el F1-Score y las matrices de confusión. A partir de estos resultados, se compararán los diferentes prototipos de modelos de clasificación de núcleos hepáticos, cada uno con un enfoque específico derivado de problemáticas observadas en prototipos anteriores. Basado en esta comparación, se definirá un prototipo final y se presentará el algoritmo final diseñado para la clasificación de núcleos hepáticos. Además, se comparará la clasificación del prototipo final con la clasificación realizada por el software de alto rendimiento MotionTracking mediante las métricas previamente mencionadas y en una muestra de regeneración hepática.

Finalmente, se llevará a cabo un análisis exhaustivo de los resultados obtenidos en comparación con el software MotionTracking y en una muestra de regeneración hepática, con un enfoque en los motivos detrás de estos resultados. Se explorarán posibles trabajos futuros y se discutirán estrategias para mejorar aún más la clasificación de núcleos hepáticos en 3D.

1.5 Temario

- i. Capítulo 1 Introducción: En esta sección se presenta una introducción general al trabajo, indicando los objetivos y alcances del proyecto.
- ii. Capítulo 2 Marco teórico: Se realiza una revisión bibliográfica sobre Deep Learning, con énfasis en Redes Neuronales Convolucionales. Se analizan sus arquitecturas básicas, conceptos y componentes determinantes para el funcionamiento y entrenamiento de una red. Se presentan las métricas de rendimiento a utilizar y se examinan modelos de renombre en el campo, junto con investigaciones previas relacionadas con la clasificación de áreas similares.
- iii. Capítulo 3 Base de datos y generación del Ground Truth: Se describen las imágenes utilizadas, resaltando las diferencias entre ellas y explicando la selección de un subconjunto. Se detalla el proceso de creación manual del Ground Truth para la segmentación y clasificación de núcleos hepáticos. Se realiza un análisis cuantitativo del Ground Truth de clasificación, seguido de una discusión sobre los hallazgos obtenidos.
- iv. Capítulo 4 Implementación: Se presentan los diversos pasos realizados para la implementación de los diferentes modelos destinados a la clasificación de núcleos hepáticos.

- v. Capítulo 5 Resultados: Se exponen los resultados de los distintos modelos de clasificación y su comparación, además de una comparación con la clasificación realizada con el software Motion-Tracking y con una muestra de regeneración hepática.
- vi. Capítulo 6 Conclusión: Se exponen las conclusiones obtenidas tras el desarrollo y evaluación del algoritmo diseñado, resumiendo los principales hallazgos y destacando tanto los logros alcanzados como las limitaciones identificadas durante el proceso. Además, se consideran posibles líneas de investigación futura, sugiriendo áreas de mejora, expansión o investigación adicional.

Capítulo 2. Marco Teórico

2.1 Introducción

En este capítulo se presenta un estudio breve del tejido hepático y su composición. Se describe el DL, centrándose en el funcionamiento y los componentes de las CNN y su entrenamiento. A continuación, se ofrece una breve descripción de las métricas de rendimiento para evaluar la clasificación binaria y multicaso. El capítulo continúa con la descripción de un software de apoyo para la segmentación y clasificación manual de núcleos, y concluye con una revisión de algunos modelos destacados en el área de clasificación de imágenes y en el área de interés.

2.2 Tejido hepático

2.2.1 Función del hígado

Es un órgano esencial encargado de la detoxificación, el metabolismo y la síntesis de proteínas. Su importancia no solo radica en sus funciones, sino también en su relación directa con el sistema circulatorio, específicamente con la vena porta y la vena hepática, ya que permite el flujo sanguíneo hacia y desde el hígado [2]. Sus células principales, los hepatocitos, producen la bilis, que es transportada a través de los canalículos biliares e interviene en procesos digestivos asociados al metabolismo. La sangre es transportada por las sinusoides, que permiten el intercambio transvascular. Esto facilita que los hepatocitos realicen sus funciones vitales, ya que la sangre en las sinusoides es rica en nutrientes [4].

2.2.2 Estructura del hígado

Existe una gran variedad de modelos que definen la estructura hepática, pero para nuestro caso se considera el modelo clásico, que define al lobulillo hepático como la unidad funcional [2].

Esta estructura funcional corresponde normalmente a un hexágono, con la vena central en su eje central. En cada vértice, se encuentra una tríada porta, compuesta por la vena y arteria hepáticas, además del ducto biliar, como se aprecia en la Fig. 2.1. Entre estas estructuras se encuentran células como los hepatocitos, las células estrelladas y las células de Kupffer. La conexión entre la vena central y la porta está

dada por las sinusoides, estructuras que atraviesan los hepatocitos en todas direcciones. Los canalículos biliares también atraviesan los hepatocitos, recogiendo la bilis producida, que luego se transporta hacia el ducto biliar [4].

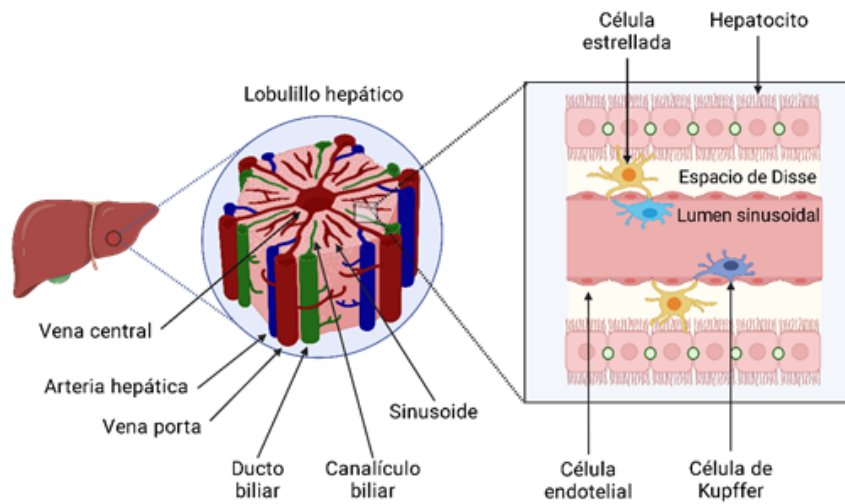


Fig. 2.1: Lobulillo hepático y su composición.

Una subunidad del lobulillo, la cual es la sección donde se obtiene las imágenes utilizadas en esta memoria, corresponde al área entre el centro del lobulillo y un vértice, es decir, el espacio entre la CV y la PV, el cual es referido como eje porto-central o eje CV-PV. Una representación de este se observa en la Fig. 2.2.

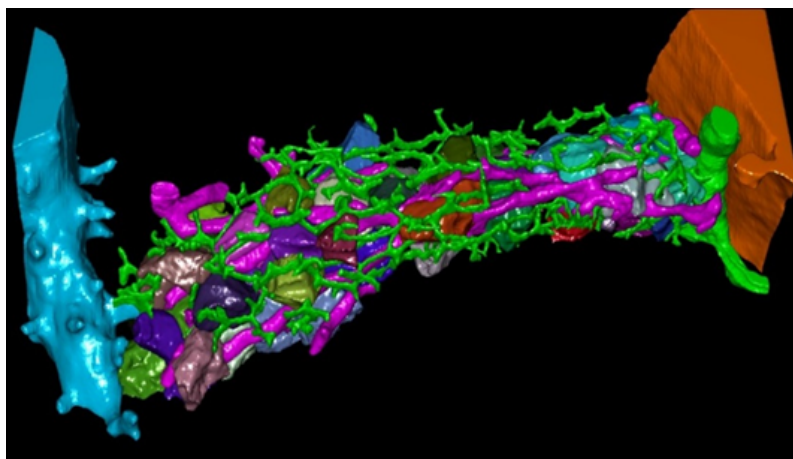


Fig. 2.2: Reconstrucción en 3D del eje CV-PV.

Reconstrucción del eje porto-central donde se aprecia la vena central de color celeste y la vena porta de color naranja. Además, se aprecian los canalículos biliares en verde, las sinusoides en magenta y los hepatocitos de diferentes colores a lo largo del eje porto central [1].

2.2.3 Células presentes en el hígado

1. **Hepatocitos:** Conforman la mayoría del volumen del tejido hepático. Estas células son polares, con caras apicales que se organizan para constituir los canalículos biliares. En estos canalículos, liberan la bilis, que luego será transportada hacia el conducto biliar. Por otro lado, sus caras basales están en contacto con las sinusoides, que son responsables de llevar la sangre entre las venas central y porta [3].
2. **Células estrelladas:** También conocidas como células Ito, son células especializadas ubicadas en la parte exterior del endotelio sinusoidal. Estas células almacenan grasa y vitamina A. Su abundancia de microtúbulos y microfilamentos citoplasmáticos vinculados a la pared sinusoidal, junto con su capacidad para mostrar actividad contráctil, sugiere que desempeñan un papel importante en la regulación del flujo sanguíneo en las sinusoides. Su forma no convexa se debe principalmente a la presencia de múltiples y delgados procesos citoplasmáticos que se extienden desde el soma central para abrazar extensamente las superficies del endotelio que rodea las sinusoides [4].
3. **Células de Kupffer:** Forman parte de la pared sinusoidal y desempeñan un papel crucial en la eliminación de partículas, células y sustancias perjudiciales o extrañas en la sangre portal. Son macrófagos que, al estar anclados a la superficie endotelial interna de la sinusoide, están expuestos al torrente sanguíneo. La irregularidad de estas células se debe a las microvellosidades, filopodios y lamelipodios en su superficie, junto con procesos que las conectan al endotelio o cruzan el lumen hacia el extremo opuesto de la sinusoide, confiriéndoles una forma no convexa distintiva [4].
4. **Células Endoteliales:** Son células planas con núcleos curvados y delgadas ramas nucleares. Son cruciales para el equilibrio de lípidos y vitaminas, y también secretan citocinas y factores de crecimiento [5].
5. **Colangiocitos:** Son células epiteliales que forman el revestimiento interno del árbol biliar. Su tamaño y forma dependen del diámetro del conducto biliar. Contienen cilios primarios que detectan los cambios en el flujo biliar, lo que produce alteraciones en su secreción [5].

2.3 Deep Learning

El DL, una subárea del Machine Learning (ML), se distingue por la profundidad y complejidad de su arquitectura, basada en Redes Neuronales Artificiales. En el entrenamiento del DL se utilizan dos mecanismos principales: el Aprendizaje Supervisado y el Aprendizaje No Supervisado. La diferencia

clave radica en que el Aprendizaje Supervisado utiliza resultados reales conocidos como GT, mientras que el Aprendizaje No Supervisado se enfoca en tareas de agrupación sin contar con un GT [21].

2.3.1 Red Neuronal Convolutiva

Las CNN son algoritmos fundamentales en el DL. Lo que diferencia a las CNN de otros enfoques convencionales, como el ML, es su capacidad para procesar directamente imágenes como entradas del modelo. En lugar de depender de un sistema de pesos, las CNN utilizan la operación de convolución, que emplea filtros para extraer características relevantes de los datos, actuando como pesos en el proceso. Una ventaja clave de las CNN es su capacidad para aprender automáticamente características importantes de los datos, eliminando la necesidad de selección manual. Esta característica las hace efectivas en el procesamiento de imágenes, ya que pueden adquirir conocimiento y experiencia a través del entrenamiento [22].

2.3.2 Arquitectura CNN

La arquitectura básica de una CNN se compone de una serie de capas diseñadas para extraer automáticamente características de una imagen. La entrada de la CNN es la propia imagen, y su objetivo es generar una salida que represente la clasificación de la imagen. Entre la entrada y la salida se encuentran diferentes tipos de capas, que incluyen capas de convolución, capas de agrupamiento y capas totalmente conectadas o densas. Estas capas se organizan en bloques, y la CNN en su conjunto se forma al combinar varios de estos bloques. La interacción entre estas capas es unidireccional, ya que la salida de una capa se convierte en la entrada de la siguiente [22] [23]. La Fig. 2.3 ilustra la arquitectura básica de una CNN.

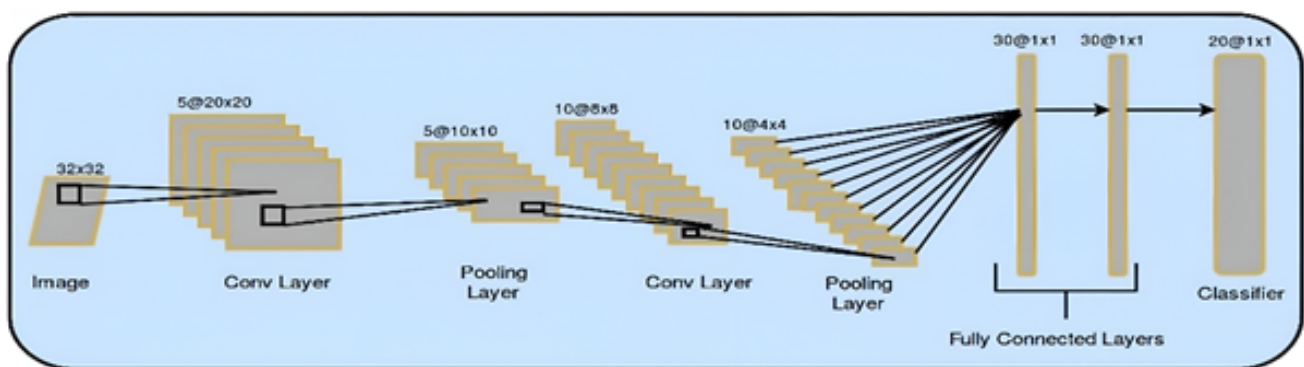


Fig. 2.3: Arquitectura básica de una red neuronal convolutiva.

Esta consta de las capas de convolución y agrupación, las cuales conforman la etapa de extracción de características, seguidas por las capas totalmente conectadas, las que corresponden a la etapa de clasificación [23].

2.3.2.1 Capa de convolución

En esta capa, se emplea un filtro de convolución sobre una matriz de números, que en este contexto representan las imágenes. Esta matriz se denomina tensor. La función de esta capa es extraer características del tensor, y el resultado se conoce como mapa de características [24]. Este proceso se ilustra en la Fig. 2.4.

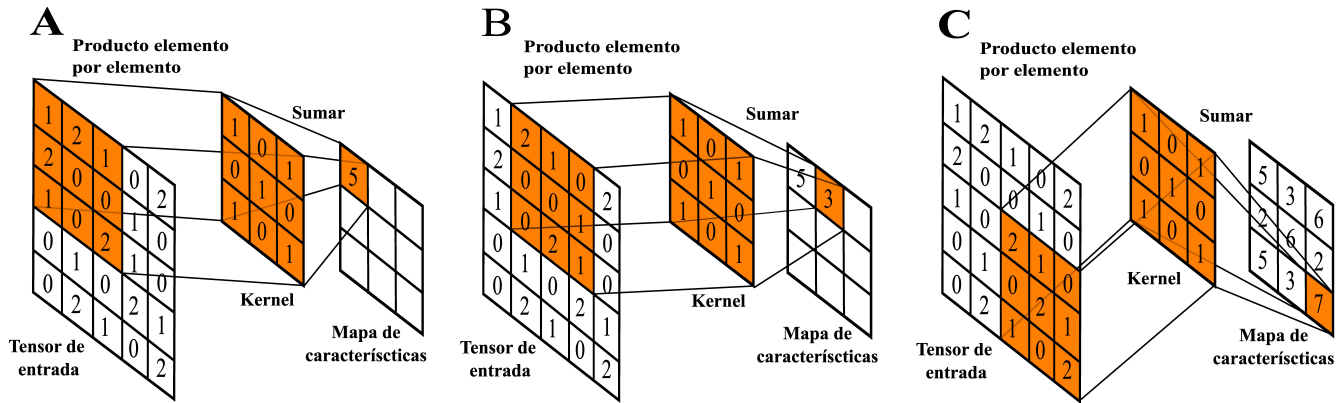


Fig. 2.4: Ejemplo de la operación de convolución.

Ejemplo de operación de convolución para un kernel de dimensión 3x3 sobre un tensor de dimensión 5x5. A) Convolución con el primer elemento del tensor. B) Convolución con el segundo elemento del tensor. C) Resultado final de la convolución. Figura adaptada de [24].

2.3.2.2 Capa de agrupación

Esta capa se conoce como capa de pooling debido a su función de seleccionar un único valor de un conjunto de valores adyacentes en un arreglo. Este proceso se realiza mediante operaciones predefinidas, como el average pooling, que calcula el promedio de los valores seleccionados, o el max pooling, que elige el valor máximo [24]. El propósito de esta capa es reducir la dimensionalidad del mapa de características para evitar la pérdida de información importante y disminuir el costo computacional. Este proceso se ilustra en la Fig. 2.5.

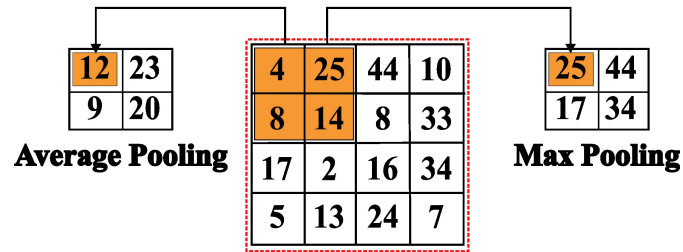


Fig. 2.5: Ejemplo de operaciones Max Pooling y Average Pooling.
Adaptado de [25].

2.3.2.3 Capa densa

Esta capa, también denominada fully-connected, se obtiene tras aplanar el arreglo de entrada, reduciendo su dimensionalidad de dos a una, es decir, convirtiéndolo en un vector. Su función es establecer conexiones entre las características extraídas para generar la salida del sistema, es decir, la clasificación. La capa final, es una capa densa que tiene el mismo número de nodos que clases a clasificar [24].

2.3.2.4 Funciones de activación

Estas facilitan la clasificación mediante la correcta separación de los datos después de operaciones como convoluciones o aplanamientos. Esto se logra reduciendo los costos computacionales y aumentando la no linealidad del problema. La función más comúnmente utilizada para esto es la Unidad Lineal Rectificada (ReLU), como se muestra en la ecuación 2.3.1. Sin embargo, al llevar todos los valores negativos a cero, esta función sufre del problema de neuronas muertas. Esto ocurre cuando una neurona en la red produce valores negativos constantes que, al ser llevados a cero, desactivan la neurona. Para solucionar este problema, se propuso la Unidad Lineal Rectificada con Fuga (Leaky ReLU), que se muestra en la ecuación 2.3.2. Esta función permite que las neuronas que producen valores negativos tengan una pequeña pendiente, evitando que se apaguen y permitiendo que continúen aprendiendo [26].

Por otra parte, en la capa final de clasificación es común utilizar funciones de activación como la sigmoide 2.3.3 para la clasificación binaria y softmax 2.3.4 para la clasificación multiclase. Estas funciones normalizan los valores de los nodos en las últimas capas de la red y los representan como probabilidades, las cuales se utilizan para obtener la clasificación final [24] [27].

$$F(z) = \begin{cases} 0, & \text{si } z < 0 \\ z, & \text{si } z \geq 0 \end{cases} \quad (2.3.1)$$

$$F(z) = \begin{cases} 0.01z, & \text{si } z < 0 \\ z, & \text{si } z \geq 0 \end{cases} \quad (2.3.2)$$

$$\sigma(z) = \frac{1}{1 + e^{-z}} \quad (2.3.3)$$

$$\sigma(z_i) = \frac{e^{z_i}}{\sum_{j=1}^K e^{z_j}} \quad \text{para } i = 1, 2, \dots, K \quad (2.3.4)$$

Donde z : valor observado (en ecuación 2.3.1 y 2.3.2) y probabilidad predicha (en ecuaciones 2.3.3 y 2.3.4); i : dato observado; K : número total de datos.

2.3.2.5 Entrenamiento de una CNN

El proceso de entrenamiento de una CNN considera la configuración de los hiperparámetros. Los 5 principales hiperparámetros y una pequeña descripción de estos se realiza a continuación [25] [28]:

1. Epochs: Epochs (o épocas) es un hiperparámetro que determina cuantas veces se realiza el proceso de entrenamiento. Cada epoca implica que todas las imágenes de entrenamiento se utilizan una vez. Su valor óptimo varía según contexto.
2. Batch Size: El hiperparámetro batch size o (tamaño de lote) configura la cantidad de datos que se consideran antes de actualizar los pesos de la red. Usualmente se ajusta a potencias de dos para facilitar el procesamiento. Un batch es un grupo de datos que se pasan a través de la red juntos, antes de actualizar los pesos.
3. Optimizador: Componente clave que trabaja en conjunto con la función de pérdida para actualizar los pesos de la red con el objetivo de minimizar el error. Utiliza el gradiente de la función de pérdida para ajustar los pesos y lograr la convergencia a un valor de pérdida mínimo. Un optimizador común en las tareas de clasificación es Propagación de la Media Cuadrática (RMSprop), el cual divide la tasa de aprendizaje por un promedio de los cuadrados de los gradientes que decae exponencialmente.

4. Funciones de pérdida: Esta calcula el valor de error, que mide la diferencia entre la salida de la red y el valor real o GT. En otras palabras, cuantifica qué tan lejos está la predicción del modelo de la realidad. Existen diferentes tipos de funciones de pérdida según el tipo de clasificación que se está realizando. Para la clasificación binaria se tiene la Entropía Cruzada Binaria y para multiclase se tiene la Entropía Cruzada Categórica.
5. Learning-rate: Hiperparámetro crítico y determina qué tan rápido o lento aprende el modelo durante el proceso de entrenamiento. Su valor es configurado en un rango específico, usualmente entre $1e^{-1}$ y $1e^{-7}$. Por esto, si su valor es muy pequeño, el entrenamiento puede ser extremadamente lento, mientras que uno demasiado grande generaría dificultad en la convergencia de un mínimo valor de pérdida. Encontrar el valor óptimo depende del contexto y su complejidad, pero su obtención es crucial.

Las expresiones matemáticas del optimizador y las funciones de pérdida se presentan a continuación.

El optimizador RMSprop está denotado por la ecuación:

$$\theta_{t+1} = \theta_t - \frac{\eta}{\sqrt{E[g^2]_t + \epsilon}} g_t \quad (2.3.5)$$

Donde t denota la iteración actual; θ representa el peso; η la tasa de aprendizaje; $E[g^2]_t$, g_t la gradiente de la función de pérdida con respecto a los parámetros en la iteración actual; ϵ es una pequeña constante para prevenir la división por 0, generalmente es 1×10^{-7} o 1×10^{-8} .

Para la clasificación binaria, se tiene la Entropía Cruzada Binaria, denotada por:

$$Loss = -(y \log(p) + (1 - y) \log(1 - p)) \quad (2.3.6)$$

Donde y es el valor binario indicando la clase observada, y p es la probabilidad de que la clase observada corresponda a la clase real.

Para la clasificación multiclase, se tiene la Entropía Cruzada Categórica, denotada por:

$$Loss = - \sum_{c=1}^M y_{o,c} \log(p_{o,c}) \quad (2.3.7)$$

Donde o es la clase observada, c la clase real, p la probabilidad de que o corresponda a c , y es el valor binario que indica si o y c coinciden, y M el número de clases.

2.3.2.6 Técnicas de regularización

Existen técnicas de regularización que ayudan a manejar los problemas de sobreajuste o subajuste, lo cual mejora la capacidad de generalización del modelo. A continuación, se presentan algunas de estas técnicas [25] [29].

1. Dropout: Consiste en la desconexión aleatoria de conexiones en la red neuronal, regulada por un parámetro numérico que varía de 0 a 1, representando el porcentaje de neuronas que se inhiben. Al aplicar Dropout, se omite un porcentaje de pesos durante el entrenamiento, lo que resulta en la disminución del aprendizaje específico y en un aumento de la capacidad de generalización del modelo.
2. Weight decay: Esta técnica reduce la magnitud de los pesos del modelo mediante una penalización proporcional añadida a la función objetivo durante el entrenamiento. Ajusta los pesos para minimizar los errores, al mismo tiempo que los mantiene pequeños, lo que contribuye a disminuir la complejidad del modelo. Las técnicas más comunes son L1 para el subajuste y L2 para el sobreajuste.
3. Batch Normalization: Esta consiste en ajustar las entradas de cada capa de una red neuronal para que tengan una distribución estable. Esto se logra calculando la media y la varianza de un grupo de datos (mini-batch) y luego ajustando todas las entradas del grupo en función de estos valores. Además, se utilizan parámetros especiales para escalar y trasladar las entradas, lo que contribuye a estabilizar el entrenamiento de la red neuronal.
4. Data Augmentation: Consiste en aumentar artificialmente el número de muestras en el conjunto de datos mediante transformaciones como traslación, escalado, rotación, reflexión y transformaciones afines, con el propósito de mejorar la generalización del modelo y fortalecer su robustez.

2.4 Métricas de evaluación

Si bien existen diferentes métricas de rendimiento, para evaluar una clasificación, ya sea multiclase o binaria, se definen las siguientes métricas de rendimiento que son capaces de entregar una evaluación

detallada y global del funcionamiento de la clasificación [30]:

1. True positives o verdaderos positivos (TP): clases identificadas como positivas cuando la etiqueta real también es positiva.
2. True negatives o verdaderos negativos (TN): clases identificadas como negativas cuando la etiqueta real también es negativa.
3. False positives o falsos positivos (FP): clases identificadas como positivas cuando la etiqueta real es negativa.
4. False negatives o falsos negativos (FN): clases identificadas como negativas cuando la etiqueta real es positiva.

En base a estas 4 métricas se definen las métricas de calidad utilizadas para evaluar el rendimiento de la clasificación de los modelos.

2.4.1 Precision (Precisión)

La precisión corresponde a la proporción de casos positivos predichos por el modelo que son verdaderamente positivos. Es decir, es la proporción de núcleos predichos como pertenecientes a una clase específica que realmente son núcleos de esa clase. Esta métrica se define mediante la siguiente ecuación:

$$Precision = \frac{TP}{TP + FP} \quad (2.4.1)$$

2.4.2 Recall (Sensibilidad)

La sensibilidad, también conocida como recall o tasa de verdaderos positivos, corresponde a la proporción de casos positivos reales que fueron correctamente identificados por el modelo. Es decir, es la proporción de núcleos de una clase específica que fueron correctamente predichos como pertenecientes a esa clase por el modelo. Mientras más alto sea el valor, mejor. Esta métrica se define mediante la siguiente ecuación:

$$Recall = \frac{TP}{TP + FN} \quad (2.4.2)$$

2.4.3 F1-Score

El F1-score combina precisión y recall en un solo valor para proporcionar un equilibrio entre la identificación correcta de positivos y la minimización de falsos positivos y falsos negativos. Su valor va de 0 a 1. Cuando es 1, tanto la precisión como el recall son perfectos, es decir, no existen falsos positivos ni falsos negativos. Cuando es 0, significa que la precisión o el recall son 0. A medida que este número aumenta, indica un mejor equilibrio entre la minimización de falsos positivos y falsos negativos. Al considerar ambas métricas igualmente importantes, penaliza de manera equitativa los errores en ambas direcciones. El F1-score se define mediante la siguiente ecuación:

$$F1 - Score = \frac{2 * Precision * Recall}{Precision + Recall} \quad (2.4.3)$$

2.4.4 Accuracy (Exactitud)

Indica el porcentaje de clasificaciones correctas entre la totalidad de datos. Denotado por:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (2.4.4)$$

A lo largo de esta memoria se utilizarán las versiones ponderadas de cada métrica, es decir, la versión *weighted*. Esto implica calcular la métrica para cada clase y luego encontrar el promedio ponderado según el número de instancias verdaderas de cada clase. Esta elección se debe a que en el tejido hepático hay un gran desbalance de clases y, por lo tanto, los conjuntos de datos utilizados para obtener estas métricas estarán desbalanceados.

La utilización de métricas ponderadas permite obtener una evaluación más representativa del desempeño del modelo en presencia de este desbalance, garantizando que las clases menos representadas no se vean eclipsadas por las clases más predominantes.

2.4.5 Matriz de confusión

Una matriz de confusión expone un análisis visual de la clasificación, esto con sus clases reales en filas y las predicciones del modelo en columnas. Dependiendo de la clase observada, los valores para TP), TN), FP y FN pueden ser obtenidos según se muestra a continuación:

Para una clasificación multiclase, para las clases C_1 , C_2 , C_3 , C_4 y C_5 se tiene su representación en la Fig. 2.6.

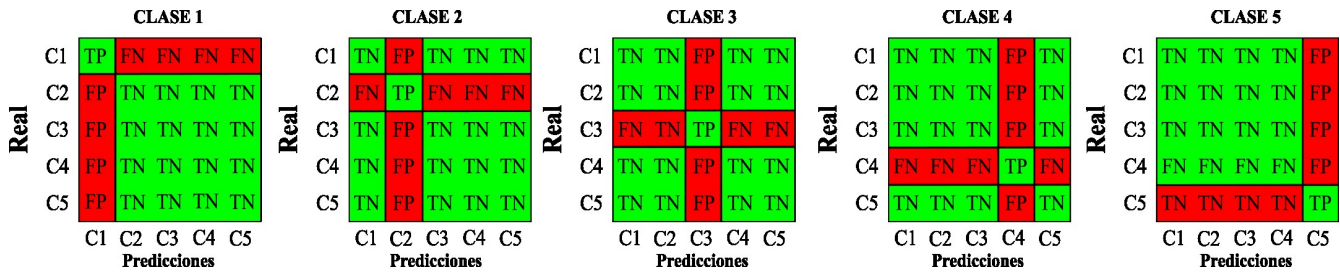


Fig. 2.6: Matriz de Confusión para 5 Clases.

Para una clasificación binaria, para las clases C_0 y C_1 se tiene su representación en la Fig. 2.7.

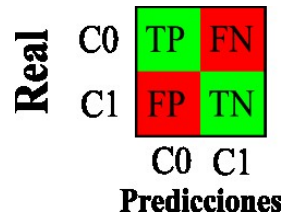


Fig. 2.7: Matriz de confusión para clasificación binaria.

2.5 Trabajos previos

2.5.1 Napari

Napari [14] es un visor de imágenes multidimensionales e interactivo para Python, diseñado específicamente para la exploración, anotación y análisis de grandes imágenes multidimensionales. Entre sus características destacadas, Napari permite la visualización de volúmenes 3D y la generación de capas adicionales de las mismas dimensiones para etiquetas. Estas capas permiten dibujar, borrar y rellenar volúmenes 3D, asignando diferentes números y colores de etiquetas.

En nuestro caso, Napari se utilizó en combinación con el plugin nD-Annotator [15], que agrega la capacidad de interpolar máscaras de etiquetas iguales. Esta funcionalidad facilita significativamente el trabajo de segmentación manual al permitir la interpolación precisa entre planos de etiquetas, mejorando la eficiencia en la anotación y segmentación de grandes conjuntos de datos.

2.5.2 MotionTracking

MotionTracking es un software especializado en el procesamiento de imágenes 2D y 3D, ofreciendo herramientas avanzadas para segmentación, clasificación y reconstrucción de elementos dentro de las imágenes. Su enfoque principal se centra en el análisis cuantitativo de células, núcleos y estructuras biológicas. Una de sus funciones destacadas es la clasificación automática de núcleos 3D, crucial para etiquetar y clasificar células según las características de sus núcleos.

Esta herramienta fue introducida en versiones más recientes del software. Aunque ya existían capacidades previas para la clasificación nuclear y celular en imágenes 2D, la extensión de este proceso a imágenes 3D presentaba desafíos debido a limitaciones en el número de marcadores fluorescentes que pueden ser utilizados simultáneamente dentro del software. Para abordar esta limitación, se desarrolló un algoritmo basado en características morfológicas, empleando clasificadores supervisados como el Análisis Discriminante Lineal (LDA) y el Clasificador de Redes Bayesianas (BNC).

Estos clasificadores demostraron una alta precisión, alcanzando aproximadamente el 100 % en la identificación de núcleos de hepatocitos, y alrededor del 95.4 % (LDA) y 92.6 % (BNC) en otros núcleos hepáticos [1].

A pesar de los resultados prometedores, la eficacia de MotionTracking puede verse afectada en imágenes tisulares menos regulares o que sean diferentes a las utilizadas en el estudio, lo que requiere ajustes manuales adicionales que pueden resultar en un consumo de tiempo significativo. Este aspecto será explorado en capítulos posteriores.

2.5.3 ResNet50

El aumento de profundidad de una red neuronal, es decir, el número de capas, típicamente se comporta en forma directamente proporcional con la precisión en el conjunto de entrenamiento. No obstante, en redes muy profundas, existe un punto en el cual esta mejora se estabiliza y puede incluso disminuir, fenómeno conocido como "degradación del rendimiento".

La arquitectura ResNet50 aborda el desafío del "*problema de degradación*" en redes neuronales profundas al introducir conexiones de "atajo" que permiten un entrenamiento más efectivo incluso en redes con muchas capas. Estas conexiones, junto con el principio de función residual, facilitan la propagación de la información a través de las capas. ResNet50 utiliza el modelo Bottleneck, que incluye tres capas en las conexiones de "atajo", en contraste con las dos capas utilizadas en otras versiones como ResNet-18 y

ResNet-34. Aunque todas las variantes de ResNet ofrecen mejoras en el problema de degradación, ResNet50 se destaca por su capacidad para mantener y mejorar la precisión incluso con un gran número de capas [16]. La arquitectura de ResNet se aprecia en la Fig. 2.8 y en la Fig. 2.9 el modelo bottleneck.

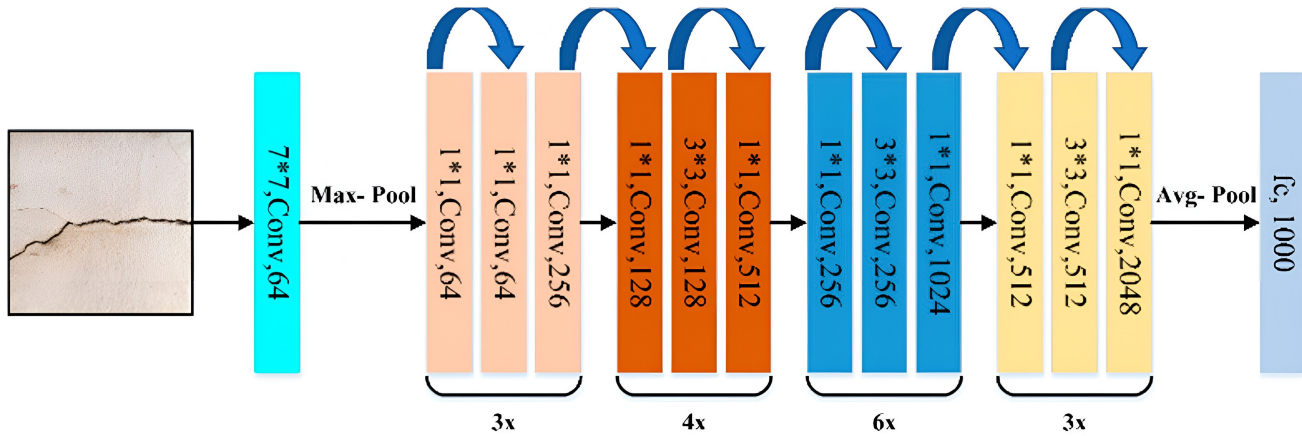


Fig. 2.8: Arquitectura ResNet50.

Obtenido de [16].

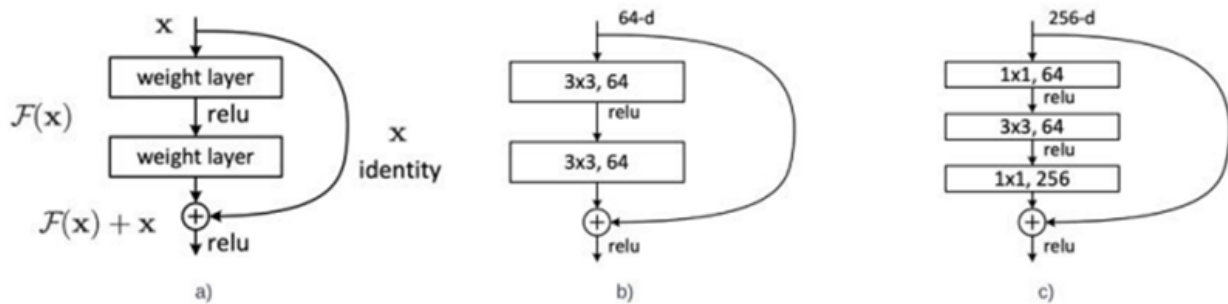


Fig. 2.9: Bloque residual.

a) bloque residual genérico. b) bloque residual de dos capas. c) bloque residual de tres capas (Bottleneck) [16].

2.5.4 InceptionV3

La arquitectura Inception se fundamenta en el módulo Inception, que aplica una variedad de filtros de convolución y capas de pooling a diferentes partes de una misma entrada, concatenando los resultados para obtener una salida. Esta estrategia permite explorar diversas combinaciones de filtros y tamaños, adaptándose a diferentes situaciones sin necesidad de especificar previamente qué configuración funcionará mejor. Sin embargo, esta flexibilidad conlleva un alto costo computacional, que se aborda mediante la inclusión de convoluciones de tamaño 1x1 antes de las operaciones principales, reduciendo drástica-

mente la carga computacional al disminuir la dimensión del input. Los bloques de la arquitectura Inception, como InceptionV3, están compuestos por múltiples módulos Inception apilados sucesivamente, con modificaciones como factorizaciones de filtros y la inclusión de clasificadores auxiliares y técnicas de regularización para mejorar el rendimiento y evitar el sobreajuste [17]. La arquitectura específica de Inception-V3 se ilustra en la Fig. 2.10.

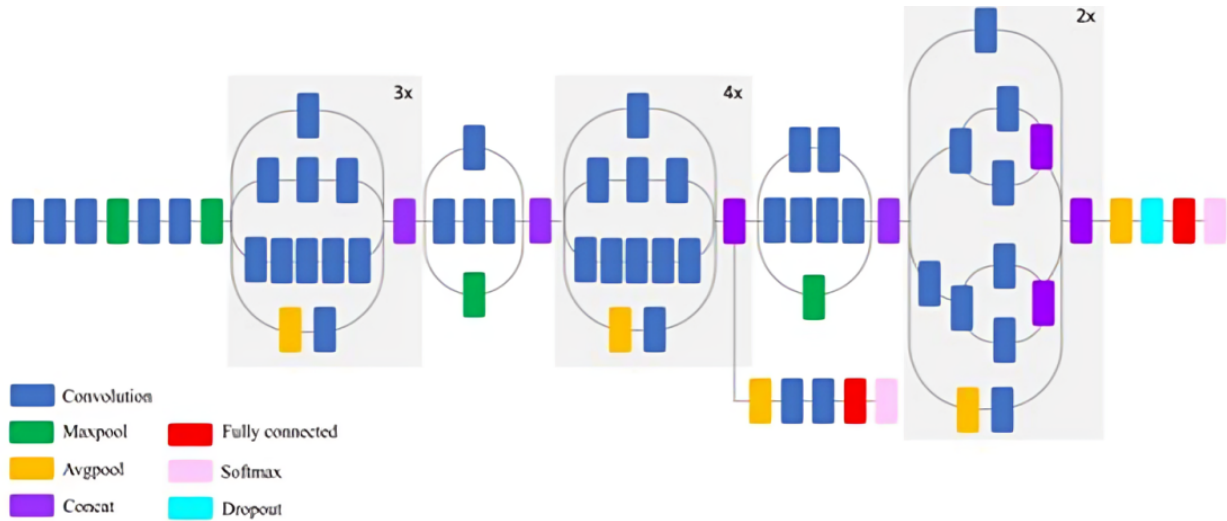


Fig. 2.10: Arquitectura InceptionV3.

Obtenido de [17].

2.5.5 Inception-ResNet-v2

Esta arquitectura combina lo mejor de las redes ResNet e Inception para ofrecer una solución robusta en el aprendizaje profundo.

Inception utiliza múltiples ramas con filtros de diferentes tamaños (1x1, 3x3, 5x5), que se fusionan mediante concatenación. Esto permite que el modelo capture una variedad de características a diferentes escalas, mejorando su capacidad interpretativa. Por otro lado, ResNet se enfoca en facilitar el entrenamiento de redes profundas mediante el uso de conexiones residuales. Estas conexiones permiten que el gradiente fluya de manera más efectiva a través de la red, lo que ayuda a prevenir problemas de degradación en redes profundas. Inception-ResNet-v2 integra estos conceptos al incluir bloques Inception-ResNet-A, B y C. Estos bloques están diseñados para estabilizar la red al reducir la escala de los residuos antes de añadirlos, combinando la eficacia de las conexiones residuales con la flexibilidad de las ramas de filtro múltiple [18].

La arquitectura de Inception-ResNet-v2 se ilustra en la Fig. 2.11, mientras que la composición deta-

llada de los bloques Inception-ResNet-A, B y C se puede ver en la Fig. 2.12 .

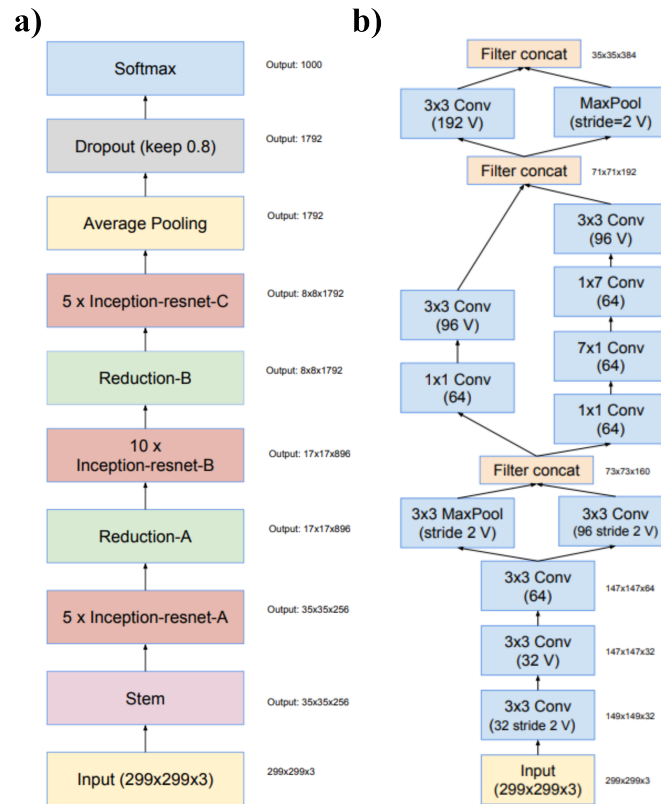


Fig. 2.11: Arquitectura Inception-ResNet-v2 y de su bloque Stem.

a) Arquitectura de Inception-ResNet-v2; b) Arquitectura del bloque Stem de Inception-ResNet-v2 [18].

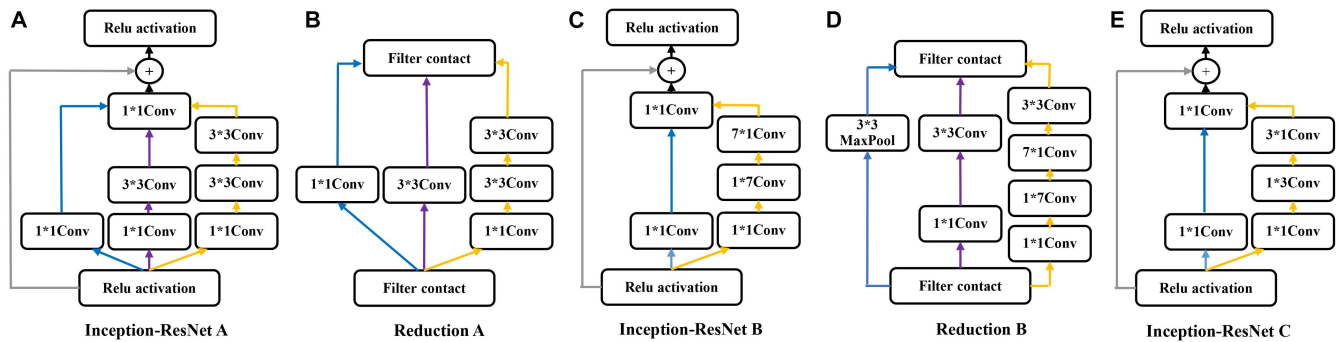


Fig. 2.12: Arquitectura de los bloques Inception-ResNet y Reduction.

a) Arquitectura bloque Inception-ResNet A; b) Arquitectura bloque Reduction A; c) Arquitectura bloque Inception-ResNet B; d) Arquitectura bloque Reduction B; e) Arquitectura bloque Inception-ResNet C [31].

2.5.6 Trabajos previos en el área de interés

Se ha demostrado que tanto ResNet50 como InceptionV3 son algoritmos eficaces para la clasificación de una variedad de imágenes en 2D, logrando una precisión comparable a la de los expertos médicos. Estos algoritmos han obtenido los mejores resultados en la clasificación de imágenes de tomografías ópticas (OCT), resonancia magnética (MRI) y tomografía computarizada (CT). No solo se destacan como clasificadores, sino también como extractores de características [32].

Por otra parte, el algoritmo EmbryoNet utiliza ResNet18 para clasificar una serie de imágenes de embriones obtenidas de películas en campo claro en un rango de 2 a 27 horas post-fertilización. Con esta red, se obtiene una exactitud en algunos casos mayor o equiparable a la de los expertos [33].

Además, según la literatura, los modelos ResNet e Inception son ampliamente reconocidos por su excelente rendimiento en diversas aplicaciones de clasificación de imágenes, abarcando campos como imágenes médicas, biológicas e histológicas, además de su uso original en conjuntos de datos como ImageNet [34], que se centra en una amplia gama de imágenes cotidianas como animales, edificaciones y paisajes [35].

Finalmente, considerando el rendimiento probado de estos modelos, se exploró su posible aplicación y adaptación a nuestro contexto tridimensional, especialmente en el ámbito biomédico. Esto se logró utilizando la biblioteca ‘classification-models-3D’ [36] de Python [19], que ofrece implementaciones de modelos clásicos adaptados para el análisis en 3D.

2.6 Discusión

Basándonos en la literatura revisada y lo expuesto en este capítulo, es evidente que los modelos ResNet e Inception destacan como opciones preferentes para la clasificación 3D de núcleos hepáticos debido a su eficacia en diversos campos relacionados. EmbryoNet utiliza ResNet18 por la necesidad de un modelo ligero y rápido. Sin embargo, en nuestra situación no estamos limitados por estas restricciones, por lo que la elección de ResNet50 parece más adecuada al ofrecer un equilibrio entre complejidad y eficiencia.

También consideramos la adopción de InceptionV3 como una opción viable, dada su robustez y rendimiento notable. Dado el buen desempeño individual de estos modelos, el uso de InceptionResNetV2, que combina las mejores características de ambos, podría proporcionar resultados aún más prometedores.

La elección de estos modelos nos permite evaluar la problemática en un espectro amplio. Tendríamos un modelo base como ResNet50, un modelo intermedio como InceptionV3 y un modelo de vanguardia como Inception-ResNet-v2. Esto nos permitirá analizar correctamente el funcionamiento de estas CNN en el complejo tejido hepático, que cuenta con núcleos de diversas formas y con una considerable similitud entre ellos.

El uso de métricas de rendimiento proporcionará un análisis más robusto del problema y nos permitirá elegir el mejor modelo para nuestra problemática. En cuanto a los datos necesarios para entrenar estos modelos, Napari y sus complementos nos facilitarán esta tarea, permitiéndonos realizar una segmentación y clasificación precisas para obtener GT de alta calidad.

Capítulo 3. Base de datos y generación del Ground Truth

3.1 Introducción

Este capítulo explora el proceso de generar el GT a partir de imágenes multicanal de microscopía confocal, obtenidas mediante la inmunotinción del eje CV-PV de tres muestras. Se seleccionaron meticulosamente nueve cubos de 280x320x320 píxeles de estas imágenes de la zona pericentral, periportal y central del eje CV-PV, con el fin de representar correctamente el tejido.

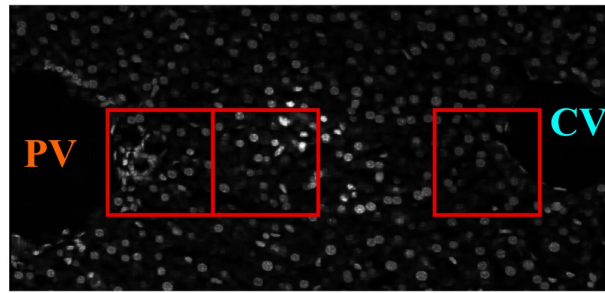
Se incluye una breve descripción de las imágenes y un detallado proceso de generación del GT dividido en una etapa de segmentación y otra de clasificación. Se finaliza con un análisis cuantitativo enfocado en la cantidad y tipos de núcleos presentes en cada cubo, así como en las relaciones entre estos tipos de núcleos. Este análisis proporciona una comprensión más profunda de la distribución de los núcleos y cómo abordar desbalances al entrenar y probar los modelos.

3.2 Imágenes a utilizar

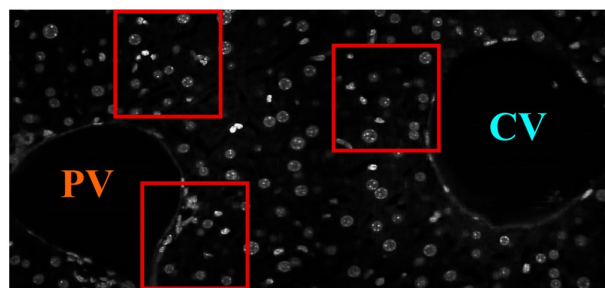
Se utilizaron imágenes de microscopía confocal multicanal que incluían canales específicos como DAPI para los núcleos, F/480 para las células de Kupffer, Desmina para las células estrelladas y Faloidina para los bordes celulares. De estas imágenes, se seleccionaron tres cubos por muestra de las zonas central, periportal y pericentral, con el fin de obtener una representación adecuada de la imagen completa. En la Fig. 3.1 se puede apreciar la ubicación desde donde fueron extraídos los cubos.

De los cubos obtenidos, solo los de las muestras 1 y 2 mostraron la presencia de ducto biliar en la zona portocentral, lo que implica la presencia de colangiocitos, un punto de interés en el estudio. En la muestra 3, se observó una cantidad significativa de núcleos en los sinusoides que no pudieron identificarse, posiblemente correspondientes a linfocitos o células sanguíneas, también relevantes para la investigación.

a) Muestra 1 (P30 1)



b) Muestra 2 (12S30)



c) Muestra 3 (8S13)

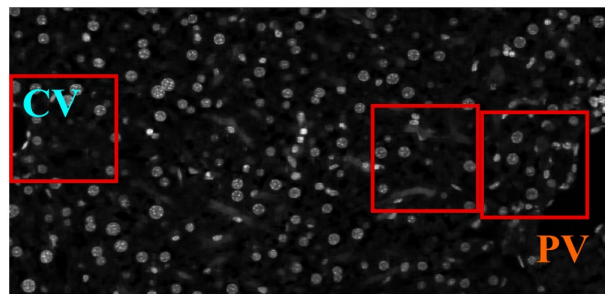


Fig. 3.1: Ubicación relativa de extracción de núcleos para cada muestra.

3.3 Generación del Ground Truth

La generación del GT fue un proceso extenso, realizado en colaboración con tres miembros del laboratorio Cell and Tissue Architecture Lab, y tomó aproximadamente 4 semanas, tal como estaba previsto.

Este proceso se dividió en dos etapas y se llevó a cabo utilizando el software Napari [14] y su complemento napari-nD-annotator [15].

3.3.1 Etapa de segmentación

Durante esta etapa, se procedió a segmentar los núcleos presentes en cada cubo, asignándoles etiquetas únicas sin continuidad entre cubos. Se utilizó Napari [14] para cargar las imágenes y generar una capa de Layers, donde cada píxel de los núcleos fue etiquetado. Para agilizar el proceso, se empleó la herramienta Slice-Interpolator del complemento napari-nD-annotator [15], que permitía generar el núcleo completo por interpolación, pintando cada 3-4 planos para asegurar la inclusión de los extremos y la parte media del núcleo. Se utilizó el método de interpolación conocido como "posición reescalada por velocidad cuadrada"(RPSV), que preserva el contorno medio con precisión visual y ofrece una interpolación superior al mantener la disposición relativa, rotación y escala de los contornos originales.

Una vez segmentados todos los cubos, se empleó la biblioteca Connected Components 3D de Python [19] para asignar etiquetas únicas a cada núcleo en cada cubo. Este paso fue crucial para evitar la asignación de la misma etiqueta a dos núcleos diferentes o la interrupción de la continuidad de las etiquetas dentro del mismo cubo debido a la extensión de los núcleos. La Fig. 3.2 muestra un ejemplo de uno de los cubos segmentados, ilustrando el resultado de este procedimiento.

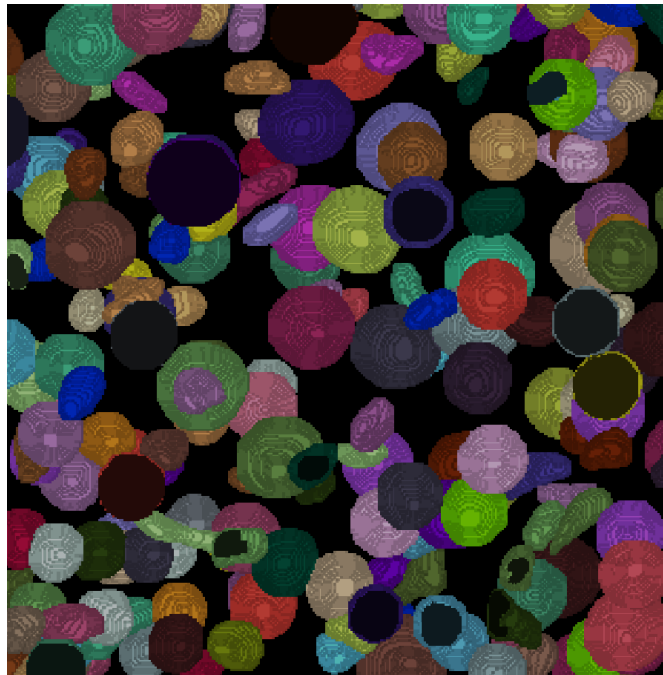


Fig. 3.2: Ground Truth de segmentación en 3D para un cubo visualizado en Napari.

Cubo segmentado de la región central de la muestra 2. Cada color representa un núcleo independiente.

3.3.2 Etapa de clasificación

Durante esta etapa, se definieron inicialmente las categorías de núcleos a clasificar, estableciendo cinco clases principales: hepatocitos, células estrelladas, células de Kupffer, células endoteliales y una categoría "sin clasificar". Esta última incluía los colangiocitos y los núcleos presentes en las sinusoides que no se ajustaban a las clases mencionadas anteriormente.

Para la clasificación, se utilizaron imágenes de canales adicionales de las muestras, que incluían DAPI para los núcleos, F/480 para las células de Kupffer, Desmina para las células estrelladas y Faloidina para los bordes celulares. Una representación de estas tinciones en un cubo de la muestra 2 se puede observar en la Fig. 3.3.

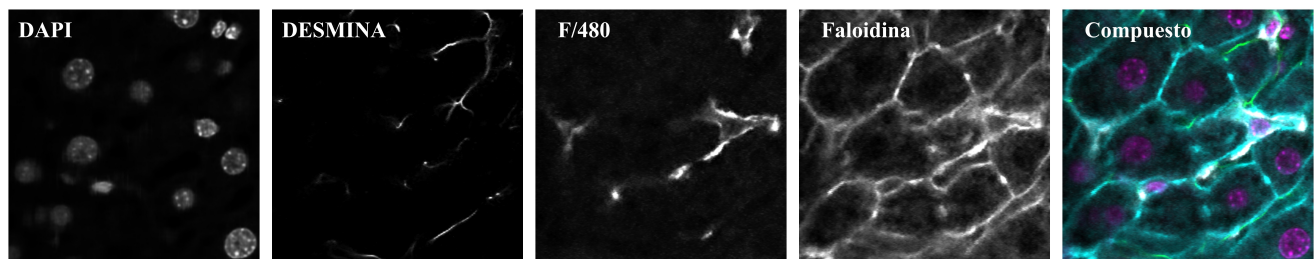


Fig. 3.3: Canales utilizados para clasificar junto con sus tinciones.

A partir de estas imágenes y mediante la superposición entre canales, se clasificaron los núcleos en las cinco categorías establecidas. Este proceso se realizó utilizando Napari [14], donde se duplicó el GT de segmentación del cubo y se aplicó la opción de relleno 3D para asignar a cada núcleo su clase correspondiente. La Fig. 3.4 muestra uno de los cubos clasificados.

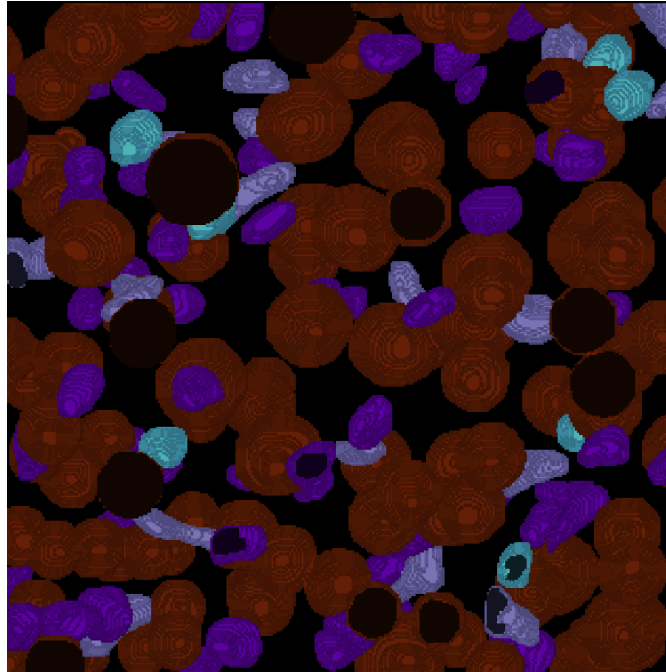


Fig. 3.4: Ground Truth de clasificación de núcleos en 3D para un cubo visualizado en Napari.

Cubo clasificado de la región central de la muestra 2. En la imagen, los núcleos están coloreados de la siguiente manera: café para los hepatocitos, cian para las células estrelladas, lavanda para las células de Kupffer, morado para las células endoteliales.

3.3.3 Análisis cuantitativo

Se realizó un análisis cuantitativo de los GT para determinar la cantidad total de núcleos y la distribución por clases en los cubos. Este análisis tenía como objetivo identificar posibles desbalances en los datos. Se llevó a cabo un conteo de las clases para cada región, desglosado por sujeto, como se detalla en la Tabla 3.2.

Tabla 3.2: Frecuencia de clase por región y muestra.

Muestra	Región	Hepatocitos	C. Estrelladas	C. de Kupffer	C. Endoteliales	Sin Clasificar	Total
1 (P30)	Pericentral	154	23	12	64	6	259
	Central	252	18	22	70	0	362
	Periportal	245	19	13	161	131	569
	TOTAL	651	60	47	295	137	1190
2 (12S30)	Pericentral	111	18	8	64	2	203
	Central	120	9	20	64	0	213
	Periportal	89	20	16	161	128	414
	TOTAL	320	47	44	289	130	830
3 (8S13)	Pericentral	119	24	6	67	0	216
	Central	113	4	18	60	28	223
	Periportal	84	40	18	202	0	344
	TOTAL	316	68	42	329	28	783
TOTAL							2803

3.4 Discusión

Los resultados del capítulo confirman un evidente desequilibrio de clases tanto a nivel individual como regional. La muestra 2 se destaca como la más desbalanceada, mostrando la menor representación de la clase Sin Clasificar, como se observa en la Tabla 3.2. Además, se revela que la muestra 1 posee la mayor cantidad de núcleos, seguida por la muestra 2 y, finalmente, la muestra 3.

El desbalance de clases se concentra en las células estrelladas, las células de Kupffer y la clase Sin Clasificar para los tres sujetos, mientras que los hepatocitos y células endoteliales son predominantes. Para entrenar y probar el modelo, se seleccionaron las muestras 1 y 2 para el entrenamiento, y la muestra 3 para la prueba. Esta decisión se justifica por la presencia exclusiva de ductos biliares y colangiocitos en las muestras 1 y 2, aspectos relevantes para el estudio. Además, se garantiza una distribución similar entre el conjunto de prueba y entrenamiento para las clases de células estrelladas y células de Kupffer.

Capítulo 4. Implementación

4.1 Introducción

En este capítulo, se detalla la implementación de modelos diseñados específicamente para clasificar diferentes tipos de núcleos hepáticos. Se han realizado modificaciones en 3D para los modelos ResNet50, InceptionV3 e Inception-ResNet-v2 utilizando TensorFlow 2.13.1 [20].

Primero se definen los datos que se utilizarán para entrenar, validar y probar los modelos. Luego, se detallan los pasos del preprocesamiento, y se continúa con el procedimiento para la implementación de los diferentes prototipos. En la implementación, se especifica primero la aumentación de datos a utilizar para el entrenamiento de los modelos. A continuación, se aborda un primer prototipo que realiza la clasificación de cinco clases previamente definidas con una base de datos desbalanceada. Posteriormente, se especifica un nuevo conjunto de entrenamiento para balancear las clases de la base de datos, creando subconjuntos balanceados para utilizar en modelos binarios.

Se prosigue con la implementación del segundo prototipo, que realiza una clasificación multiclase de cuatro clases, y se continúa con la implementación de cuatro modelos binarios para cada una de las clases del segundo prototipo. Finalmente, se describe cómo se obtiene el prototipo final y cómo será evaluado en nuevas muestras, permitiendo analizar su robustez.

4.2 Procedimiento para la implementación de los modelos de clasificación.

4.2.1 Definición conjuntos de entrenamiento, validación y prueba

Como se mencionó en el capítulo anterior, las muestras 1 y 2 se asignaron para el entrenamiento de los modelos, reservando la muestra 3 para las pruebas.

Para las muestras de entrenamiento, se utilizó el 90 % de los núcleos para el proceso de entrenamiento y el 10 % restante para la validación del modelo. La muestra 3 se destinó íntegramente a la fase de prueba. La distribución de clases y la cantidad de núcleos empleados en cada conjunto se detallan en la Tabla 4.3.

Tabla 4.3: Distribución de clases por conjunto.

Conjunto	Clase 1: Hepatocito	Clase 2: Estrellada	Clase 3: Kupffer	Clase 4: Endotelial	Clase 5: Sin Clasificar	Total
Entrenamiento	870	100	81	534	233	1818
Validación	101	7	10	50	34	202
Prueba	316	68	42	329	28	783

4.2.2 Preprocesamiento

Para trabajar con las CNN, es crucial aislar los núcleos en el volumen 3D antes de entrenar y clasificar. Esto evita posibles errores de clasificación al considerar la imagen completa como una sola clase, lo cual podría confundir a la red.

Antes de pasar las imágenes a la red, se aplicó un preprocesamiento a cada cubo obtenido en el capítulo anterior. Este proceso se dividió en tres bloques. Primero, se normalizó la imagen completa según la ecuación 4.2.1:

$$Normalizar(Imagen) = \frac{I - I_{min}}{I_{max} - I_{min}} \quad (4.2.1)$$

Donde I es la imagen original, I_{min} y I_{max} son los valores mínimo y máximo de intensidad en la imagen, respectivamente. Esta normalización asegura que los valores de intensidad estén en el rango $[0, 1]$, lo que facilita el entrenamiento de la red.

Después de obtener las máscaras de segmentación, se calcula el centroide de cada núcleo como el promedio de las coordenadas de los puntos en la máscara. Luego, se crea un cubo de $64 \times 64 \times 64$ alrededor de cada centroide, asegurando que el núcleo esté centrado dentro del cubo. En este proceso, se eliminan los núcleos que no corresponden a la etiqueta actual dentro del cubo, garantizando así que cada núcleo quede aislado. Este procedimiento se ilustra en la Fig. 4.1.

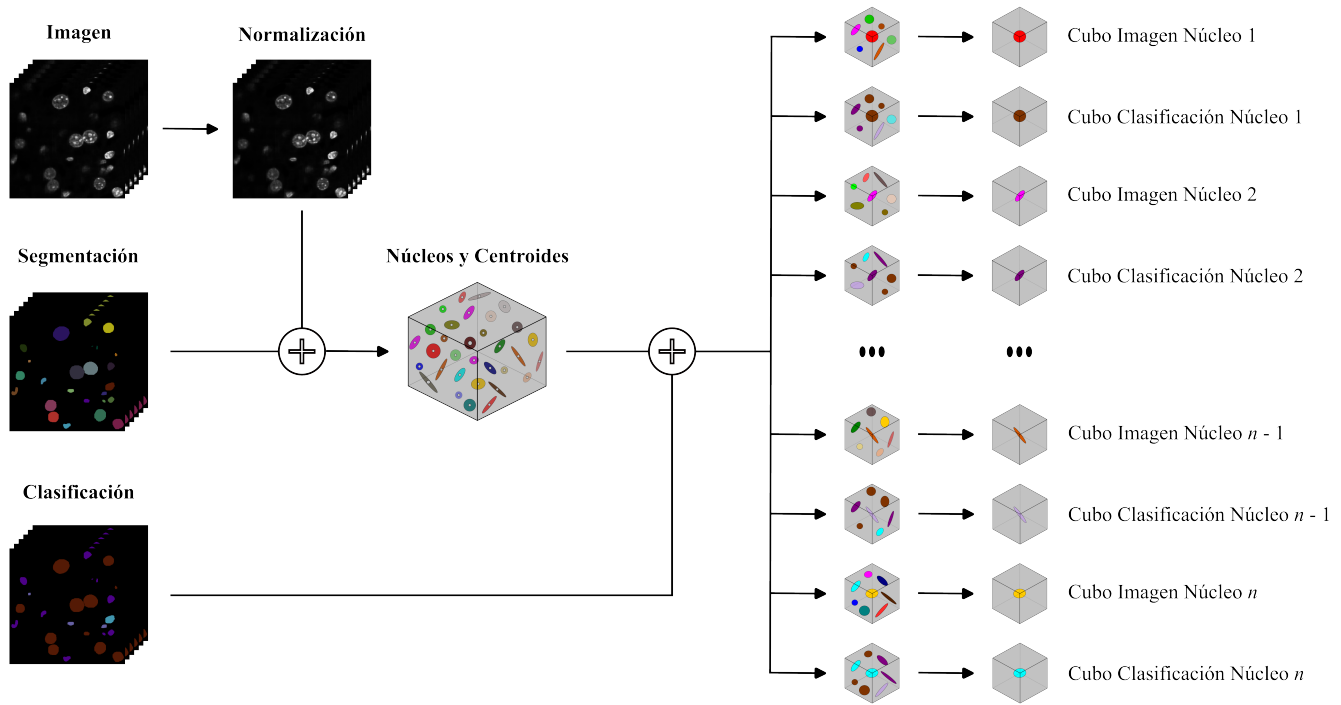


Fig. 4.1: Diagrama del preprocesamiento utilizado para entrenar las CNN.

4.2.3 Implementación de los modelos de clasificación

En este trabajo se implementaron tres CNN sobresalientes en la clasificación de imágenes, adaptadas para trabajar en un contexto tridimensional y sin preentrenamiento. Los modelos seleccionados fueron ResNet50 [16], InceptionV3 [17] e Inception-ResNet-v2 [18]. Todos los modelos se implementaron utilizando Python 3.9 [19] y TensorFlow 2.13.1 [20]. De estos, se terminó eligiendo uno solo para definir el modelo final basándonos en sus resultados.

Estas CNN se utilizaron como "backbone"(columna), y solo se modificó la capa final de clasificación. En particular, se eliminó la última capa densa de estos modelos, ya que estaban diseñados para clasificar 1000 clases [16], [17], [18]. Para adaptarlos a nuestras necesidades específicas, se reemplazó esta capa por un nuevo bloque de clasificación.

El nuevo bloque de clasificación consta de cinco capas secuenciales. En primer lugar, se emplea una capa de GlobalAveragePooling3D para reducir la dimensionalidad de las características extraídas de la red y convertirlas en un vector de tamaño fijo, facilitando así el paso a las últimas capas densas de clasificación de forma directa. Luego, se aplica una capa de Flatten para reducir la dimensionalidad nuevamente, convirtiendo el tensor de características a una dimensión, con el mismo objetivo que la capa

anterior. A continuación, se utiliza una capa densa con 2048 neuronas, con función de activación Leaky ReLU, y una capa de dropout con factor 0.5, para evitar el sobreajuste. Este enfoque previene la reducción abrupta de las características de la red y, junto con el dropout, ayuda a mitigar el overfitting. Finalmente, se incluye una capa densa con n neuronas, donde n es el número de clases a clasificar. Esta última capa tiene diferentes versiones, ya que se definieron modelos binarios y multiclase para distintos prototipos. Adicionalmente, para evitar el sobreajuste, se les aplicó a ambas capas densas una regularización L2 con valor de $1e^{-4}$. Un diagrama simplificado de esto se aprecia en la Fig. 4.2.

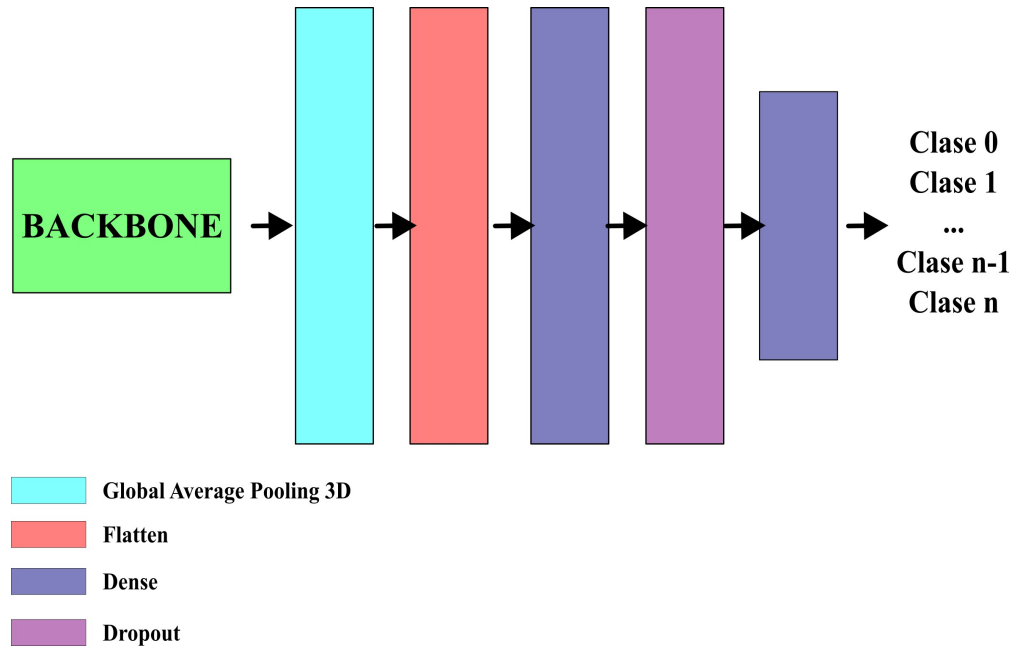


Fig. 4.2: Diagrama básico de los modelos de clasificación.

4.2.3.1 Aumentación de datos

Para entrenar los modelos se utilizaron 5 operaciones para aumentar la cantidad de imágenes de entrenamiento. Estas fueron: voltear la imagen en 3D con respecto a cualquiera de sus tres ejes de forma aleatoria; transponer la imagen en 3D con respecto a cualquiera de sus tres ejes de forma aleatoria; cambiar el brillo de la imagen utilizando la transformación gamma de ley de potencia con un rango de 0.8 a 1.2 para los parámetros $gain$ y γ , aplicados de forma independiente por canal; aplicar un desenfoque gaussiano aleatorio a la imagen, con un rango de 0 a 1 para σ ; y añadir ruido de Poisson aleatorio a la imagen, con un rango de 0 a 1. Estas aumentaciones se aplicaron a cada lote de imágenes durante el entrenamiento de forma aleatoria, es decir, de 100 imágenes se les aplican N aumentaciones, donde N puede ir de 0 a 5, siendo 5 la aplicación de todas las aumentaciones y 0 ninguna. La probabilidad de que ocurra cada aumentación es independiente entre sí, ya que se realiza de acuerdo a una probabilidad

aleatoria para cada imagen dentro del lote. Este método de aumentación es el utilizado en [37].

En la Fig. 4.3 se aprecia un ejemplo de las aumentaciones aplicadas a una imagen de un núcleo de hepatocito.

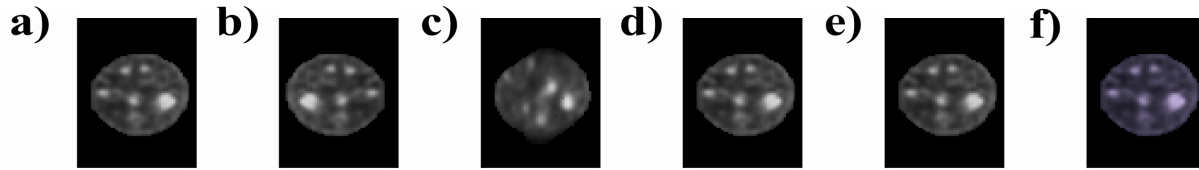


Fig. 4.3: Imagen de un núcleo de hepatocito junto con el resultado de las operaciones de aumentación de datos.

a) Imagen Original, b) Volteada, c) Transpuesta, d) Desenfocada, e) Con ruido Poisson, f) Modificación del brillo

4.2.3.2 Primer prototipo de los modelos

Se definió $n = 5$ en la última capa densa para clasificar en las cinco clases mencionadas anteriormente y se utilizó la función de activación softmax en la última capa densa.

Para los modelos ResNet50, InceptionV3 e Inception-ResNet-v2, se utilizó la función de pérdida de entropía cruzada categórica para los modelos de clasificación multiclase, y la función de pérdida de entropía cruzada binaria para los modelos de clasificación binaria. Se empleó RMSprop con una tasa de aprendizaje de 0.0045 y ϵ de 1.0 para mantener la estabilidad del entrenamiento. La tasa de aprendizaje se redujo un 10 % cada 2 épocas para mejorar la convergencia. Además, los modelos fueron entrenados durante 100 épocas con un tamaño de lote de 100.

4.2.3.3 Nuevo conjunto de entrenamiento

Dado que no fue posible manejar el desbalance de clases, se procedió a balancear el conjunto de entrenamiento. Se clasificaron el resto de las muestras 1 y 2, enfocándose solo en los núcleos de células estrelladas, células de Kupffer y células endoteliales. La segmentación de estas células no se realizó manualmente; se utilizó una red disponible en el laboratorio. Además, se incluyeron otras muestras previamente clasificadas y segmentadas, obtenidas de la misma forma y del mismo tipo que la muestra 1, aunque la segmentación fue realizada con software MT sin corrección manual. Debido al tiempo requerido para preparar un GT del tejido y a la baja proporción de las clases minoritarias en comparación con las mayoritarias, se decidió sacrificar algo de precisión en la segmentación para aumentar la cantidad de datos de entrenamiento y lograr un balance de clases.

Al definir estos nuevos núcleos, se encontró que las regiones periportales, especialmente los ductos biliares que contenían la mayoría de los núcleos de la clase 5, estaban mal segmentados. Por esta razón, se optó por eliminar esta clase y concentrarse únicamente en las otras cuatro clases.

A partir de este nuevo conjunto de entrenamiento, se crearon subconjuntos balanceados para diferentes enfoques. Se mantuvieron los núcleos iniciales y se agregaron los núcleos de las nuevas muestras de forma aleatoria para lograr el balance necesario. Este nuevo conjunto y sus subconjuntos balanceados se pueden observar en la Tabla 4.4.

Tabla 4.4: Distribución de clases para el nuevo conjunto de entrenamiento y sus subconjuntos balanceados.

Conjunto	Clase 1: Hepatocito	Clase 2: Estrellada	Clase 3: Kupffer	Clase 4: Endotelial	Total
Completo	2925	2187	1035	1041	7188
4 Clases	1035	1035	1035	1035	4140
Binario Hepatocito	2925	975	975	975	5850
Binario Estrellada	729	2187	729	729	4374
Binario Kupffer	345	345	1035	345	2070
Binario Endotelial	347	347	347	1041	2082

4.2.3.4 Segundo prototipo de los modelos

Se entrenaron y probaron los tres modelos (ResNet50, InceptionV3 e Inception-ResNet-v2) con el nuevo conjunto de datos balanceado de 4 clases, ajustando diversas combinaciones de hiperparámetros para seleccionar el mejor modelo para cada uno. Se definió $n=4$ en la última capa densa para clasificar en las cuatro clases mencionadas anteriormente, y se utilizó la función de activación softmax en esta capa.

Para los modelos de clasificación multiclase, se utilizó la función de pérdida de entropía cruzada categórica. En contraste, para los modelos de clasificación binaria, se empleó la función de pérdida de entropía cruzada binaria. Se utilizó RMSprop con una tasa de aprendizaje de 0.0045 y ϵ de 1.0 para mantener la estabilidad del entrenamiento. La tasa de aprendizaje se redujo un 10 % cada 2 épocas para mejorar la convergencia. Además, los modelos fueron entrenados durante 100 épocas con un tamaño de lote de 100.

4.2.3.5 Submodelos de clasificación binaria

Debido al enfoque en el estudio del tejido hepático y los resultados de los modelos, se optó por definir un modelo binario para los núcleos de hepatocitos. Además, se crearon modelos binarios para cada una de las tres clases restantes con el fin de analizar su comportamiento en este contexto y evaluar si mejorarían los resultados de la clasificación de las cuatro clases.

El modelo binario de núcleos de hepatocitos se centra en diferenciar si un núcleo es o no de hepatocito, considerando una predicción positiva cuando lo es. Esta misma lógica se aplica a los modelos binarios para núcleos de células estrelladas, de Kupffer y endoteliales.

Para los modelos binarios, se aumentó el dropout de 0.5 al 0.6 en el bloque de clasificación final y se modificó la función de activación de la última capa densa de softmax a sigmoide. El resto de la configuración se mantuvo igual para cada uno de los modelos binarios de cada red.

Para todos los modelos, se utilizó la función de pérdida de entropía cruzada binaria. Se empleó RMS-prop con una tasa de aprendizaje de 0.0045 y ϵ de 1.0 para mantener la estabilidad del entrenamiento. La tasa de aprendizaje se redujo en un 10 % cada 2 épocas para mejorar la convergencia. Además, los modelos fueron entrenados durante 100 épocas con un tamaño de lote de 100, a excepción del modelo binario de núcleos de hepatocitos, que se entrenó durante 50 épocas.

4.2.3.6 Prototipo final

Para determinar el prototipo final, se realizó una comparación exhaustiva entre las tres redes neuronales entrenadas. Primero, se evaluaron los modelos en dos escenarios: clasificación de cuatro clases y clasificación binaria para cada clase específica. Se seleccionó el mejor modelo para cada caso, basándose en el rendimiento observado durante la evaluación.

Una vez identificados los modelos óptimos para cada tipo de clasificación, se implementó una estrategia combinada para mejorar la precisión de las predicciones. En particular, se integraron los modelos binarios junto con el modelo de cuatro clases. El objetivo de esta integración es obtener una predicción más sólida y fiable para las cuatro clases. Al combinar los modelos, se busca aprovechar la capacidad de los modelos binarios para especializarse en cada clase individualmente, mientras que el modelo de cuatro clases proporciona una visión global. Esta metodología permitió mejorar la precisión global del sistema y reducir la incidencia de errores en las predicciones.

Con este prototipo final, se procedió a evaluar su rendimiento en comparación con un software de alto rendimiento como MT. Esto permitió obtener una visión más amplia de cómo funciona el modelo final.

Finalmente, se realizó una evaluación del prototipo en un caso de estudio particular: la regeneración hepática. Se utilizó el modelo binario de núcleos de hepatocitos para predecir en una imagen de un eje CV-PV previamente clasificada, pero solo de núcleos de hepatocitos. Esta imagen se obtuvo de la misma forma y es del mismo tipo que la muestra 1. En este caso, se observaron núcleos de hepatocitos deformados debido a las diferentes etapas del ciclo celular, lo que permitió evaluar el rendimiento del prototipo en un caso de estudio específico.

4.3 Discusión

Los modelos implementados, junto con el nuevo conjunto de entrenamiento, permiten analizar si estas redes se adaptan al contexto de clasificación. También es posible comparar tres niveles de modelos. Además, se evalúa el impacto de la cantidad de datos y el balanceo de las clases. Con clases balanceadas, se minimizan las posibles convergencias locales y los resultados son más sólidos en cuanto a la capacidad de clasificación de los modelos.

Cada modelo implementado y el orden en que se aplican, junto con el nuevo conjunto de entrenamiento, permiten comprender la complejidad y la capacidad de adaptación de las redes al contexto de clasificación de núcleos hepáticos usando solo la imagen de DAPI. También es posible comparar tres niveles de complejidad de modelos. Paralelamente, se evalúa el impacto del balanceo de clases en los diferentes modelos. Con clases balanceadas, se reducen las posibles convergencias locales y los resultados son más sólidos en cuanto a la capacidad de clasificación de los modelos.

Por otro lado, los modelos binarios permiten definir un modelo especializado en núcleos de hepatocitos, clave para el estudio del tejido hepático, y proporcionan una comprensión más específica de cómo cada modelo lidia con la clasificación de cada tipo de núcleo. Con esta información, es posible definir el mejor modelo para la clasificación de núcleos hepáticos en 3D e interpretar los resultados para mejorar la clasificación de estos núcleos usando solo la imagen de DAPI.

Con el prototipo final, se puede evaluar el rendimiento del modelo en comparación con un software de alto rendimiento como MT. También se evalúa el rendimiento del prototipo en un caso de estudio específico para verificar su robustez, incluso en una muestra anómala. Esto permite una evaluación más robusta del modelo y su impacto en el laboratorio.

Capítulo 5. Predicciones y Resultados

5.1 Introducción

En este capítulo se presentan los resultados obtenidos para los diferentes prototipos y modelos definidos en el capítulo anterior. Esto incluye las gráficas de pérdida y F1-Score para los modelos multiclase, así como las gráficas de pérdida y exactitud para los modelos binarios. Además, se incluyen las matrices de confusión junto con tablas que muestran las métricas de rendimiento para los diferentes casos presentados. El capítulo finaliza con la definición del prototipo final y del algoritmo final, y se evalúa el rendimiento del prototipo con el software MT y su comportamiento en una muestra de tejido de regeneración hepática.

5.2 Primer prototipo

Las curvas de pérdida y F1-Score durante el entrenamiento de los tres modelos con el conjunto de datos desbalanceado se muestran en las Fig. A.1, A.2 y A.3. A partir de la curva de pérdida de validación, se observa que, aproximadamente a las 50 épocas, los tres modelos comienzan a converger en un valor de pérdida cercano a 1. En la curva de pérdida de entrenamiento, se nota que, pocas épocas después, los modelos empiezan a dejar de aprender del conjunto de entrenamiento, pero convergen a valores cercanos a 0. Por otro lado, las curvas de F1-Score muestran que los valores de F1-Score para el entrenamiento convergen a valores cercanos a 1. Sin embargo, para la curva de validación ResNet50 converge aproximadamente a un F1-Score de 0.6, mientras que InceptionV3 e Inception-ResNet-v2 alcanzan valores alrededor de 0.7.

Para este primer prototipo se evaluó el modelo en 783 núcleos de 5 tipos: 316 de hepatocitos, 68 de células estrelladas, 42 de células de Kupffer, 329 de células endoteliales y 28 sin clasificar, con el fin de analizar el rendimiento de los tres modelos. En la Tabla 5.5 se presentan las métricas de calidad obtenidas en el conjunto de prueba. InceptionV3 obtiene los mejores resultados en Accuracy, F1-Score y Recall, con valores de 0.7775, 0.7371 y 0.7775, respectivamente. Por otro lado, ResNet50 obtiene el mejor resultado en Precision con un valor de 0.7834. Finalmente, Inception-ResNet-v2 presenta resultados relativamente peores, con todas sus métricas rondando el 0.72. Sin embargo, su F1-Score (0.7207) es superior al de ResNet50 (0.6794), aunque su Precision es menor, con un valor de 0.7569.

Las matrices de confusión para los tres modelos obtenidas para el primer prototipo con el conjunto de prueba se aprecian en las Fig. A.4, A.5 y A.6. En estas Fig., se puede apreciar claramente que tanto los núcleos de hepatocitos como los núcleos de células endoteliales muestran una mayoría de núcleos correctamente clasificados, especialmente los núcleos de hepatocitos, aunque también con muchos falsos positivos. En contraste, los núcleos de células estrelladas y células de Kupffer presentan 0 predicciones correctas en ResNet50. InceptionV3 e Inception-ResNet-v2 logran solo 3 y 6 predicciones correctas para núcleos de células estrelladas, respectivamente, y solo Inception-ResNet-v2 reconoce núcleos de células de Kupffer, aunque solo uno. Para los núcleos sin clasificar, los resultados son similares en las tres redes.

5.2.1 Discusión

Los resultados obtenidos muestran que, aunque las redes convergen de forma correcta, no lo hacen de manera ideal, ya que la pérdida no tiende a 0 y los valores de F1-Score no son altos. Esto indica un problema claro en la capacidad de generalización de las redes al clasificar núcleos con los cuales no fueron entrenadas, especialmente para núcleos de células estrelladas, células de Kupffer y núcleos sin clasificar. Esta baja capacidad de generalización puede deberse al gran desbalance entre la cantidad de estos tipos de núcleos y los núcleos de hepatocitos y células endoteliales. El desbalance puede llevar a que las redes aprendan a reconocer principalmente los núcleos que están en mayor cantidad, mientras que los núcleos con menor representatividad no se diferencian adecuadamente de los mayoritarios. Esto explicaría por qué, aunque se tienen todas las métricas de rendimiento por encima de 0.7 en InceptionV3 e Inception-ResNet-v2, la clasificación de los núcleos minoritarios no es buena. Por ello, un paso lógico a seguir es entrenar estas redes con un conjunto de entrenamiento balanceado entre los diferentes tipos de núcleos. Este enfoque podría mejorar la capacidad de las redes para reconocer y clasificar correctamente todos los tipos de núcleos, aumentando así la precisión y la generalización del modelo, lo cual se aborda en el segundo prototipo.

5.3 Segundo prototipo

Las curvas de pérdida y F1-Score durante el entrenamiento de los tres modelos con el nuevo conjunto de datos balanceado (4 Clases) se aprecian en las Fig. B.1, B.2 y B.3. En las curvas de pérdida de entrenamiento del segundo prototipo, se observa que los modelos comienzan a dejar de aprender del conjunto de entrenamiento en una época similar a la del primer prototipo y logrando convergencia similar. En la curva de pérdida de validación, aunque las redes aún convergen, la transición no es tan suave

como en el primer prototipo. Por otro lado, el F1-Score para el entrenamiento mantiene su convergencia cercana a 1. Sin embargo, para la validación, InceptionV3 e Inception-ResNet-v2 muestran una mejora, alcanzando un F1-Score de aproximadamente 0.8, mientras que ResNet50 se mantiene alrededor de 0.6.

Para este segundo prototipo se evaluó el modelo en 755 núcleos de 4 tipos: 316 de hepatocitos, 68 de células estrelladas, 42 de células de Kupffer y 329 de células endoteliales, con el fin de analizar el rendimiento de los tres modelos. En la Tabla 5.5 se presentan las métricas de calidad obtenidas en el conjunto de prueba. Se observa que, principalmente, Inception-ResNet-v2 obtiene los mejores resultados en Accuracy, F1-Score, Recall y Precision con 0.7881, 0.7588, 0.7881 y 0.7391, respectivamente. Comparado con los resultados del primer prototipo, este modelo muestra un aumento en todas las métricas de rendimiento, excepto en Precision, que disminuye en 0.0178. Por otro lado, tanto InceptionV3 como ResNet50 muestran una disminución en todas sus métricas de rendimiento en comparación con el primer prototipo. InceptionV3 presenta una disminución moderada, con una reducción máxima de 0.0676 y mínima de 0.0482, mientras que ResNet50 experimenta una disminución significativa, con una reducción máxima de 0.1708 y mínima de 0.1171.

Las matrices de confusión para los tres modelos obtenidas para el segundo prototipo con el conjunto de prueba se aprecian en las Fig. B.4, B.5 y B.6. De estas tenemos que los tres modelos mantuvieron la correcta predicción de la mayoría de los núcleos de hepatocitos. ResNet50 e InceptionV3 tuvieron una disminución considerable en la clasificación correcta de núcleos de células endoteliales, aumentando los falsos positivos en las demás clases. Por otra parte, Inception-ResNet-v2 mejoró la clasificación de estos núcleos. Para los núcleos de células estrelladas, ResNet50 pasó de 0 a 9, InceptionV3 de 6 a 2, e Inception-ResNet-v2 se mantuvo igual. En los núcleos de células de Kupffer, ResNet50 aumentó de 0 a 10, InceptionV3 de 0 a 7, e Inception-ResNet-v2 de 1 a 5.

Tabla 5.5: Métricas de calidad para los modelos de clasificación multiclase de núcleos hepáticos.

Modelo	Métrica			
	Accuracy	F1-Score	Recall	Precision
ResNet50	0.7338	0.6794	0.7338	0.7834
ResNet50 - Balanceado	0.5629	0.5623	0.5629	0.6452
InceptionV3	0.7775	0.7371	0.7775	0.7302
InceptionV3 - Balanceado	0.7099	0.6889	0.7099	0.6848
Inception-ResNet-v2	0.7298	0.7207	0.7298	0.7569
Inception-ResNet-v2 - Balanceado	0.7881	0.7588	0.7881	0.7391

5.3.1 Discusión

Aunque ResNet50 e InceptionV3 mejoraron en la clasificación de núcleos de células estrelladas y células de Kupffer, la pérdida de efectividad en la clasificación de núcleos de células endoteliales sugiere una menor capacidad de diferenciación entre estos tres tipos de núcleos. En el caso desbalanceado, la elevada cantidad de datos correctos en núcleos de células endoteliales podría estar sesgada. Esto se hace más evidente al notar una disminución del rendimiento de hasta un 0.17 en las métricas de una de estas dos redes al entrenarlas con el conjunto de datos balanceado. Por otro lado, Inception-ResNet-v2 se benefició del balanceo de clases. No solo logró reconocer núcleos de células de Kupffer que antes no identificaba, sino que también mantuvo la cantidad de núcleos de hepatocitos y células estrelladas correctamente clasificados y aumentó significativamente los núcleos de células endoteliales correctamente clasificados. A diferencia de las otras dos redes, Inception-ResNet-v2 mostró un efecto positivo del balanceo de clases en la tarea de clasificación, con todas sus métricas de rendimiento aumentando, excepto la precisión. A pesar de esto, los resultados muestran que, aunque Inception-ResNet-v2 mejoró sus métricas de rendimiento, sigue fallando en la clasificación de núcleos de células estrelladas y células de Kupffer. La diferencia en los resultados podría deberse a que Inception-ResNet-v2 es una red mucho más compleja que InceptionV3 y ResNet50, por lo que el aumento de datos utilizados para entrenar, junto con el balanceo de clases, resalta esta característica del modelo. Además, como se mencionó en la sección del nuevo conjunto de entrenamiento, el sacrificio en la calidad de la segmentación podría ser el motivo por el cual la red no logra clasificar mejor los núcleos minoritarios y sigue teniendo falsos positivos. Por todo esto, es evidente que, aunque Inception-ResNet-v2 clasifica correctamente los núcleos hepáticos, el aumento de la cantidad de datos para entrenar sería beneficioso para mejorar el rendimiento en la clasificación de estos núcleos. Además, si estos están correctamente segmentados sin sacrificar calidad en la segmentación, se podrían mejorar aún más los resultados de clasificación y permitiría discernir si es un problema de limitaciones de la red o no.

5.4 Modelos Binarios

5.4.1 Modelo Binario de Núcleos de Hepatocitos

Las curvas de pérdida y exactitud binaria para el entrenamiento con el conjunto balanceado de núcleos de hepatocitos (Binario Hepatocitos) de los modelos ResNet50, InceptionV3 e Inception-ResNet-v2 se presentan en las Fig. C.1, C.2 y C.3, respectivamente. En estas se observa que, en la curva de pérdida de entrenamiento, las tres redes alcanzan una convergencia en valores bajos cercanos a 0. Sin embar-

go, ResNet50 e InceptionV3 dejan de aprender después de aproximadamente 20 épocas, mientras que Inception-ResNet-v2 continúa aprendiendo hasta cerca de 40 épocas. En la curva de validación, InceptionV3 muestra una convergencia abrupta, mientras que Inception-ResNet-v2 y ResNet50 convergen de manera más suave. InceptionV3 e Inception-ResNet-v2 convergen en valores cercanos a 0, mientras que ResNet50 converge en un valor de pérdida intermedio. En la curva de exactitud binaria, los tres modelos logran una convergencia en 1 para el entrenamiento. Para la curva de validación, se observa que InceptionV3 e Inception-ResNet-v2 alcanzan una exactitud binaria superior a 0.9, siendo Inception-ResNet-v2 la que presenta la mayor exactitud. Por su parte, ResNet50 también muestra una alta exactitud binaria, pero con un valor ligeramente inferior a 0.9.

Para este modelo binario se evaluó su rendimiento con 755 núcleos, distribuidos en 316 clases verdaderas (1) y 439 clases negativas (0). En la Tabla 5.6 se presentan las métricas de rendimiento para los tres modelos en el conjunto de prueba. Las tres redes muestran buenos resultados, siendo el peor rendimiento para ResNet50, con métricas rondando 0.88. InceptionV3 e Inception-ResNet-v2 presentan resultados similares entre sí, pero mejores que ResNet50, con una diferencia de solo 0.1 en sus métricas de rendimiento. Sin embargo, Inception-ResNet-v2 destaca con un Accuracy, F1-Score, Recall y Precision de 0.9563, 0.9564, 0.9563 y 0.9571, respectivamente.

Las matrices de confusión para los tres modelos binarios con el conjunto de prueba se presentan en las Fig. C.4, C.5 y C.6. En estas matrices, se observa que Inception-ResNet-v2 tiene la menor cantidad de FN, con un total de 10, en comparación con 25 de InceptionV3 y 30 de ResNet50. Además, Inception-ResNet-v2 cuenta con la mayor cantidad de TP, con 306, frente a 291 de InceptionV3 y 286 de ResNet50. Por otro lado, InceptionV3 presenta la menor cantidad de FP, con 13, en comparación con 23 de Inception-ResNet-v2 y 54 de ResNet50. También muestra la mayor cantidad de TN, con 426, frente a 416 de Inception-ResNet-v2 y 385 de ResNet50.

5.4.2 Modelo Binario de Núcleos de Células Estrelladas

Las curvas de pérdida y exactitud binaria para el entrenamiento con el conjunto balanceado de núcleos de células estrelladas (Binario Estrelladas) de los modelos ResNet50, InceptionV3 e Inception-ResNet-v2 se presentan en las Fig. C.7, C.8 y C.9, respectivamente. En estas Fig., se observa que, en la curva de pérdida de entrenamiento, las tres redes alcanzan una convergencia en valores bajos cercanos a 0. Sin embargo, ResNet50 e Inception-ResNet-v2 dejan de aprender después de aproximadamente 80 épocas, mientras que InceptionV3 lo hace a las 40 épocas. En la curva de validación, InceptionV3 muestra convergencia, pero con posible sobre ajuste después de la época 20, mientras que Inception-ResNet-v2 y

ResNet50 convergen de manera más suave. InceptionV3 y ResNet50 convergen en valores cercanos a 0.7, mientras que Inception-ResNet-v2 converge en un valor de pérdida intermedio, aproximadamente 0.5. En la curva de exactitud binaria, los tres modelos logran una convergencia en 1 para el entrenamiento. Para la curva de validación, se observa que las tres redes alcanzan una exactitud binaria de aproximadamente 0.8.

Para este modelo binario se evaluó su rendimiento con 755 núcleos, de los cuales 68 pertenecen a la clase verdadera (1) y 687 a la clase negativa (0). En la Tabla 5.6 se pueden apreciar las métricas de rendimiento para los tres modelos en el conjunto de prueba. Los resultados son muy notorios, donde Inception-ResNet-v2 supera tanto a ResNet50 como a InceptionV3 en Accuracy, F1-Score y Recall con un 0.6358, 0.7135 y 0.6358, respectivamente, solo teniendo una precisión menor a ResNet50 con un 0.8880 frente a un 0.9101. Por otra parte, ResNet50 en el resto de métricas presenta resultados paupérrimos, menores a 0.4, mientras InceptionV3 muestra resultados menores a Inception-ResNet-v2 con una diferencia mínima de 0.0092 en Precisión y aproximadamente 0.1 en las demás métricas.

Las matrices de confusión para los tres modelos binarios evaluados con el conjunto de prueba se presentan en las Fig. C.10, C.11 y C.12. De estas matrices se observa un comportamiento similar entre InceptionV3 e Inception-ResNet-v2, ambos con 50 TP y 18 FN. En contraste, ResNet50 presenta 66 TP y solo 2 FN. Sin embargo, ResNet50 también muestra la mayor cantidad de FP con 510, comparado con 325 en InceptionV3 e Inception-ResNet-v2 con 257. Finalmente, Inception-ResNet-v2 destaca por tener la mayor cantidad de TN con 430, frente a 362 en InceptionV3 y 177 en ResNet50.

5.4.3 Modelo Binario de Núcleos de Células de Kupffer

Las curvas de pérdida y exactitud binaria para el entrenamiento con el conjunto balanceado de núcleos de células de Kupffer (Binario Kupffer) de los modelos ResNet50, InceptionV3 e Inception-ResNet-v2 se presentan en las Fig. C.13, C.14 y C.15, respectivamente. En estas Fig., se observa que, en la curva de pérdida de entrenamiento, las tres redes alcanzan una convergencia en valores bajos cercanos a 0. Sin embargo, ResNet50 e Inception-ResNet-v2 dejan de aprender después de aproximadamente 80 épocas, mientras que InceptionV3 lo hace a las 60 épocas. En la curva de validación, se muestra una convergencia suave, pero con un pico en la pérdida cerca de la época 20. Los tres modelos convergen a valores cercanos a 1 en la curva de exactitud binaria de entrenamiento. Para la curva de validación, ResNet50 e Inception-ResNet-v2 alcanzan una exactitud binaria de aproximadamente 0.6, mientras que InceptionV3 alcanza una exactitud binaria cercana a 0.8.

Para este modelo binario se evaluó su rendimiento con 755 núcleos, de los cuales 42 pertenecen a la clase verdadera (1) y 713 a la clase negativa (0). En la Tabla 5.6 se pueden apreciar las métricas de rendimiento para los tres modelos en el conjunto de prueba. Se observa una situación similar al modelo binario de núcleos de células estrelladas, donde Inception-ResNet-v2 supera a ResNet50 e InceptionV3 en Accuracy, F1-Score y Recall, con 0.6821, 0.7682 y 0.6821, respectivamente. En esta ocasión, InceptionV3 supera en Precision a Inception-ResNet-v2 con 0.9268 versus 0.9222. ResNet50 presenta nuevamente resultados paupérrimos, con Accuracy y Recall por debajo de 0.5 y un F1-Score de 0.5355, destacando únicamente en Precision con 0.9094. Por otra parte, InceptionV3 muestra un rendimiento bajo en Accuracy y Recall con valores menores a 0.5 y un F1-Score de 0.6063.

Las matrices de confusión para los tres modelos binarios con el conjunto de prueba se presentan en las Fig. C.16, C.17 y C.18. Estas muestran que InceptionV3 tiene el mayor número de verdaderos positivos (TP) con 30 y el menor número de falsos negativos (FN) con 8, en comparación con 27 TP y 15 FN para Inception-ResNet-v2, y 12 TP y 30 FN para ResNet50, respectivamente. Por otro lado, Inception-ResNet-v2 presenta la mayor cantidad de verdaderos negativos (TN) y la menor cantidad de falsos positivos (FP), con 488 y 225, respectivamente, frente a 334 TN y 379 FP para InceptionV3, y 282 TN y 431 FP para ResNet50.

5.4.4 Modelo Binario de Núcleos de Células Endoteliales

Las curvas de pérdida y exactitud binaria para el entrenamiento con el conjunto balanceado de núcleos de células endoteliales (Binario Endoteliales) de los modelos ResNet50, InceptionV3 e Inception-ResNet-v2 se presentan en las Fig. C.19, C.20 y C.21, respectivamente. En estas Fig., se observa que, en la curva de pérdida de entrenamiento, las tres redes alcanzan una convergencia en valores bajos cercanos a 0. Sin embargo, ResNet50 e Inception-ResNet-v2 dejan de aprender después de aproximadamente 80 épocas, mientras que InceptionV3 lo hace a las 60 épocas. En la curva de validación, ResNet50 muestra una convergencia rápida y cercana a 1, InceptionV3 alcanza una convergencia cercana a 0.8, e Inception-ResNet-v2 presenta una convergencia más suave, cercana a 0.6. Para la curva de exactitud binaria, las tres redes logran una convergencia cercana a 1 en el entrenamiento. En la curva de validación, ResNet50 alcanza una exactitud cercana a 0.5, InceptionV3 se acerca a 0.7, e Inception-ResNet-v2 alcanza aproximadamente 0.8.

Para este modelo binario se evaluó su rendimiento con 755 núcleos, de los cuales 329 pertenecen a la clase verdadera (1) y 426 a la clase negativa (0). En la Tabla 5.6 se pueden apreciar las métricas de rendimiento para los tres modelos en el conjunto de prueba. De esta tabla, podemos observar que

Inception-ResNet-v2 obtiene las mejores métricas de rendimiento con un Accuracy de 0.7881, un F1-Score de 0.7869, un Recall de 0.7881 y una Precision de 0.8285. Por otra parte, ResNet50 obtiene los peores resultados con todas sus métricas rondando 0.6, a excepción de Precision con un 0.6847, mientras que InceptionV3 tiene sus métricas aproximadamente 0.4 por debajo de Inception-ResNet-v2.

Las matrices de confusión para los tres modelos binarios con el conjunto de prueba se pueden apreciar en las Fig. C.22, C.23 y C.24. De donde podemos apreciar como Inception-ResNet-v2 cuenta con la menor cantidad de FN y de FP, además de la mayor cantidad de TN y TP, esto con 19, 141, 285 y 310, respectivamente. Esto, versus 21, 175, 251 y 308 de Inceptionv3, y 42, 254, 172 y 287 de ResNet50, respectivamente.

Tabla 5.6: Métricas de calidad para los diferentes modelos de clasificación de binaria de núcleos hepáticos en el conjunto de prueba.

Tipo de Modelo Binario	Modelo	Métrica			
		Accuracy	F1-Score	Recall	Precision
Hepatocitos-Otros	ResNet50	0.8887	0.8892	0.8887	0.8915
Hepatocitos-Otros	InceptionV3	0.9497	0.9495	0.9497	0.9499
Hepatocitos-Otros	Inception-ResNet-v2	0.9563	0.9564	0.9563	0.9571
Estrelladas-Otros	ResNet50	0.3218	0.3904	0.3218	0.9101
Estrelladas-Otros	InceptionV3	0.5457	0.6377	0.5457	0.8788
Estrelladas-Otros	Inception-ResNet-v2	0.6358	0.7135	0.6358	0.8880
Kupffer-Otros	ResNet50	0.4132	0.5355	0.4132	0.9094
Kupffer-Otros	InceptionV3	0.4874	0.6063	0.4874	0.9268
Kupffer-Otros	Inception-ResNet-v2	0.6821	0.7682	0.6821	0.9222
Endoteliales-Otros	ResNet50	0.6079	0.5901	0.6079	0.6847
Endoteliales-Otros	InceptionV3	0.7404	0.7364	0.7404	0.7985
Endoteliales-Otros	Inception-ResNet-v2	0.7881	0.7869	0.7881	0.8285

5.4.5 Discusión

No fue sorpresa encontrar resultados similares a los obtenidos en el segundo prototipo al evaluar los modelos binarios para los cuatro tipos de núcleos de estas redes. Inception-ResNet-v2 supera ampliamente el rendimiento de InceptionV3 y ResNet50 para los cuatro tipos de núcleos. Aunque en algunos casos tuvo una menor cantidad de TP y/o FP/FN, la diferencia era mínima. Inception-ResNet-v2 no solo presentaba mejores métricas de rendimiento, sino que el rendimiento global observado en las ma-

trices de confusión y la relación TP/TN/FP/FN era mucho mejor. Este resultado era esperable, ya que, como se mencionó, Inception-ResNet-v2 es una red más compleja y, por lo tanto, es capaz de captar diferencias más sutiles entre las diferentes clases en un conjunto de datos un poco más grande, lo que según los resultados no lograron captar ResNet50 e InceptionV3. Si bien estos modelos también podrían haberse visto afectados por el sacrificio de calidad de segmentación, en caso de aumentar más los datos y mejorar la segmentación, las tres redes se verían beneficiadas. Sin embargo, claramente lo haría más Inception-ResNet-v2, ya que mostró una amplia mejoría con los ajustes aplicados. En conclusión, Inception-ResNet-v2 ha demostrado ser la red más eficaz para la clasificación de núcleos de células hepáticas, superando a InceptionV3 y ResNet50 en métricas clave como Accuracy, F1-Score y Recall. Su capacidad para manejar un conjunto de datos más grande y balanceado resalta su superioridad en comparación con las otras redes. Para mejorar aún más el rendimiento de la clasificación, sería ideal aumentar la cantidad de datos de entrenamiento y mejorar la calidad de la segmentación, lo que beneficiaría especialmente a Inception-ResNet-v2 debido a su complejidad y capacidad de capturar diferencias más sutiles.

5.5 Prototipo final

Basándose en los resultados obtenidos con los modelos ResNet50, InceptionV3 e Inception-ResNet-v2 para la clasificación de 4 clases y los modelos binarios, se observa claramente la predominancia de Inception-ResNet-v2 en ambos casos. Por lo tanto, para definir los modelos del prototipo final, se eligieron modelos con este backbone. Se utilizaron tanto los modelos binarios de Inception-ResNet-v2 para las 4 clases como el modelo de clasificación de cuatro clases con el mismo backbone para optimizar las predicciones. Adicionalmente, para una mayor comparación y flexibilidad, se desarrolló un modelo que combina las predicciones de los modelos binarios en el siguiente orden: primero el modelo de células de Kupffer, luego el de células estrelladas, seguido del de células endoteliales, y finalmente el de hepatocitos. En este enfoque, se integran las predicciones de cada modelo de manera secuencial.

La optimización se llevó a cabo utilizando condicionales, con base en las predicciones del modelo base, que corresponde al modelo de clasificación de 4 clases con backbone de Inception-ResNet-v2. Además, se utilizaron los modelos binarios con backbone de Inception-ResNet-v2 para los cuatro tipos de núcleos. Todos estos modelos ya fueron entrenados y testados en las secciones anteriores.

La lógica de optimización se dividió en dos etapas:

1. **Primera etapa:** Si el modelo base clasifica el núcleo como núcleo de hepatocito, célula estrellada

o célula de Kupffer y el modelo binario de núcleos de células endoteliales clasifica ese mismo núcleo como núcleo de célula endotelial, entonces se utilizan los modelos binarios de núcleos de hepatocitos, células estrelladas o células de Kupffer, dependiendo del tipo de núcleo predicho por el modelo base. Si la predicción de estos es negativa, se asigna a este núcleo como núcleo de célula endotelial. Esto se hace para eliminar los FN de núcleos de células estrelladas.

2. **Segunda etapa:** Después de la primera etapa, se revisan los núcleos clasificados como núcleos de hepatocitos utilizando los cuatro modelos binarios como condicionales. Si el modelo binario de núcleos de células estrelladas predice el núcleo de forma positiva y el resto de modelos de forma negativa, se asigna este núcleo como núcleo de célula estrellada. Luego se realiza lo mismo con el modelo binario de núcleos de células de Kupffer, donde si este tiene una predicción positiva y el resto de modelos binarios una predicción negativa, se asigna este núcleo como núcleo de célula de Kupffer. Esto se hace para eliminar los FN de estos dos tipos de núcleos.

Con esta lógica de optimización, se busca mejorar la clasificación de núcleos de células hepáticas, al mismo tiempo que se minimiza la cantidad de FP para los núcleos de hepatocitos.

En la Fig. D.1 se puede apreciar la matriz de confusión obtenida al realizar esta optimización, y en la Fig. D.2 se puede apreciar la matriz de confusión obtenida de esta combinación de modelos binarios. En estas se puede apreciar como para el modelo optimizado se eliminan la mayor parte de los FP de los núcleos de hepatocito y disminuyendo al mismo tiempo los FN de los núcleos de células endoteliales, pero a costa de perder TP de núcleos de hepatocitos. Por otra parte, la combinación de los modelos binarios, permite dar una idea de como se llega a estos resultados, ya que son bastante similares. En la Tabla 5.7 se puede observar la comparación entre el modelo base, el optimizado, y la combinación de los modelos binarios. En esta tabla podemos observar como tanto la combinación de los modelos binarios como el modelo optimizado obtienen mejores resultados a nivel de métricas de rendimiento que el modelo base, donde los modelos binarios tiene los mejores resultados para Accuracy con un 0.8132, Recall con un 0.8132 y Precision con un 0.7812, mientras el modelo optimizado tiene el mejor F1-Score con 0.7786, con una diferencia solo del 0.0001 con los modelos binarios combinados, pero una diferencia mínima igual en el resto de métricas con este modelo.

Tabla 5.7: Métricas de calidad para las diferentes clasificaciones finales en el conjunto de prueba.

Modelo	Métrica			
	Accuracy	F1-Score	Recall	Precision
Modelo 4 Clases	0.7881	0.7588	0.7881	0.7391
Modelo 4 Clases Optimizado	0.8066	0.7786	0.8066	0.7670
Modelos Binarios Combinados	0.8132	0.7785	0.8132	0.7812

El algoritmo final que incluye tanto el preprocesamiento como la predicción se aprecia en la 5.1

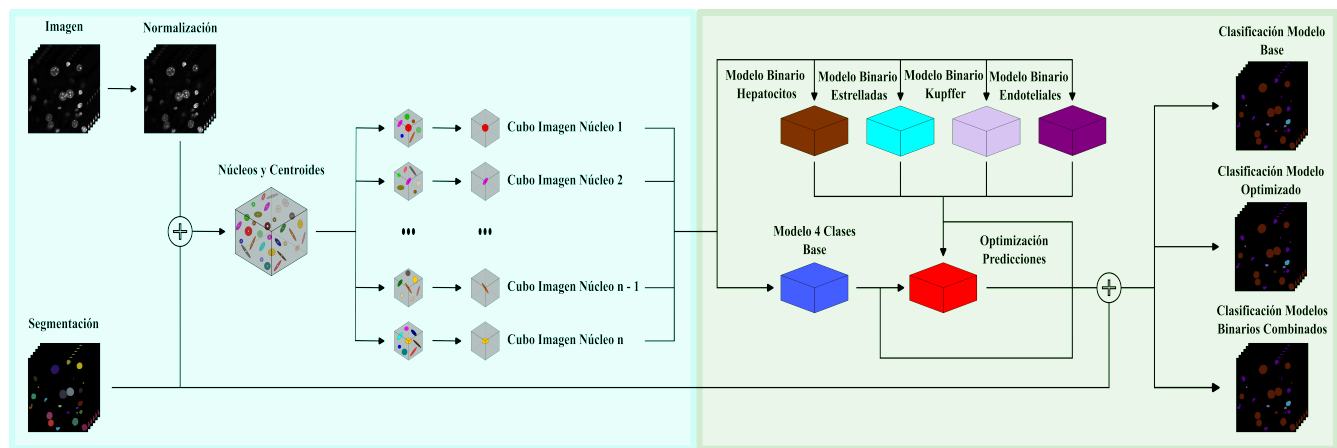


Fig. 5.1: Algoritmo final de la clasificación de núcleos hepáticos.

En cian se aprecia la sección de preprocesamiento de los núcleos con respecto a la imagen de DAPI y la máscara de segmentación de la misma, y en verde se aprecia la sección de clasificación con los diferentes modelos y las clasificaciones que da como resultado la sección.

5.5.1 Discusión

Al optimizar el modelo base utilizando los modelos binarios, se logró eliminar la mayoría de los falsos positivos en la clasificación de núcleos de hepatocitos, aunque se produjo una ligera pérdida en la cantidad de verdaderos positivos. Esto se debe a que, como se observó en secciones anteriores, aunque los modelos binarios muestran un rendimiento excelente para los núcleos de hepatocitos y un buen rendimiento para los núcleos de células endoteliales, el rendimiento para los otros tipos de núcleos, aunque aceptable, no es óptimo. Como resultado, se pierden algunas predicciones correctas en favor de mejorar la clasificación de otros tipos. A pesar de que la diferencia en las métricas entre el modelo base y el modelo optimizado no es grande, las matrices de confusión proporcionan una visión más amplia y detallada de la mejora, que es considerable. La combinación de los modelos binarios muestra una pequeña diferen-

cia en comparación con el modelo optimizado. Aunque parece que la combinación de modelos binarios ofrece mejores resultados, esto se debe en parte a la cantidad de núcleos presentes en las clases mayoritarias. Con base en estas observaciones, se decidió que el modelo final incorporará tres componentes: la predicción del modelo base, el modelo optimizado con los modelos binarios, y una combinación de las predicciones de los modelos binarios. Esta estrategia permitirá evaluar el impacto de cada enfoque y seleccionar la mejor predicción posible. La motivación para esta estrategia se abordará en el próximo capítulo. Además, los resultados individuales de cada modelo binario estarán disponibles para acceso si es necesario. Aunque, con las consideraciones ya mencionadas en secciones anteriores, se podría llegar a una única salida robusta, lo cual requeriría una evaluación adicional.

5.6 Comparación con MotionTracking

Utilizando una muestra obtenida de la misma forma que el resto de muestras de la zona central de un sujeto del mismo tipo que la muestra 1, pero no utilizada para el entrenamiento, validación o prueba de los modelos, se clasificó la misma con MT y con los modelos del prototipo final, ambos utilizando la misma segmentación. Esta muestra contaba con 171 núcleos de hepatocitos, 27 núcleos de células estrelladas, 18 núcleos de células de Kupffer y 55 núcleos de células endoteliales. En la Fig. D.3 se aprecia una matriz de confusión obtenida con el GT de la muestra y la predicción de MT. En la Fig. D.4 se aprecia la matriz de confusión obtenida con el modelo de 4 clases final (Inception-ResNet-v2), en la Fig. D.5 se aprecia la matriz de confusión obtenida con el modelo optimizado, y en la Fig. D.6 se puede apreciar la matriz de confusión obtenida de esta combinación de modelos binarios. En la Tabla 5.8 se puede apreciar una comparación de estas tres clasificaciones con métricas de rendimiento.

Los resultados muestran que las tres predicciones entregadas por el algoritmo final definido son mejores que las obtenidas con MT. El modelo base tiene un F1-Score de 0.7483, el modelo optimizado 0.7712 y la combinación de los modelos binarios 0.7541, comparados con 0.7157 de MT. Aunque la diferencia en el F1-Score es pequeña, la mejora es considerable. Para el algoritmo final, se aprecia un aumento en las predicciones correctas de núcleos de hepatocitos, pasando de un 74 % de predicciones correctas a un 92 %-96 %. Aunque MT tiene menos FP en esta clase, genera más FP en las otras clases, clasificando incorrectamente núcleos de células estrelladas, células de Kupffer y células endoteliales como hepatocitos. En los núcleos de células estrelladas, la diferencia es considerable, con el algoritmo clasificando correctamente seis veces más estos núcleos como mínimo. En los núcleos de células de Kupffer, la clasificación correcta es similar, pero MT genera falsos positivos de núcleos de hepatocitos, mientras que el algoritmo no. La clase donde no hubo mejora fue en los núcleos de células endoteliales, ya que MT

supera los resultados del algoritmo.

Tabla 5.8: Métricas de calidad para las diferentes clasificaciones en la muestra nueva.

Modelo	Métrica			
	Accuracy	F1-Score	Recall	Precision
Motion Tracking	0.6926	0.7157	0.6926	0.7786
Modelo 4 Clases	0.7592	0.7483	0.7592	0.7634
Modelo 4 Clases Optimizado	0.7592	0.7712	0.7592	0.7921
Modelos Binarios Combinados	0.7518	0.7541	0.7518	0.7745

5.6.1 Discusión

Los resultados muestran una mejora en comparación con el software MT. Sin embargo, MT sigue obteniendo más clasificaciones correctas para los núcleos de células endoteliales. Esto se debe a que, como se mencionó en capítulos anteriores, MT utiliza imágenes adicionales a la imagen de DAPI, como FLK1 y Faloidina. La Faloidina marca las paredes celulares, mientras que FLK1 está relacionado con los sinusoides. Estas imágenes proporcionan características basadas en la intensidad en diferentes distancias dentro del núcleo, lo que facilita la clasificación precisa de células endoteliales. A pesar de que el modelo propuesto en esta memoria no cuenta con imágenes adicionales y utiliza datos de menor calidad, es decir, con segmentaciones no ideales, ha logrado resultados comparables e incluso superiores a los del software MT. Esto sugiere que, con un aumento en la cantidad de datos y una mejor calidad de segmentación, el algoritmo podría superar al software MT en la clasificación de núcleos de células endoteliales, que es la única categoría en la que MT muestra superioridad. Por lo tanto, es crucial considerar las recomendaciones ya mencionadas en secciones anteriores para maximizar el rendimiento del modelo.

5.7 Muestra de regeneración hepática

Como se mencionó en el capítulo anterior, esta muestra fue obtenida de la misma manera que el resto de las imágenes utilizadas en esta memoria y corresponde al mismo tipo que la muestra 1. Se utilizó el modelo binario para la clasificación de núcleos de hepatocitos en esta muestra, utilizando la versión final del modelo con Inception-ResNet-v2 como backbone. Los resultados de esta prueba en la muestra de regeneración hepática se detallan en la Tabla 5.9. En esta, podemos apreciar que de 3957 núcleos de hepatocitos, ya sean normales o se encuentren en mitosis, se clasificaron de forma correcta 3581, lo que

equivaldría a un 90 % de los núcleos correctamente clasificados. De los núcleos de hepatocitos normales, se clasificaron de forma correcta 3469 de 3754, equivalente a un 92 % de los núcleos correctamente clasificados. Además, en los núcleos de hepatocitos en mitosis se aprecia la correcta clasificación de la mayoría de núcleos en mitosis, principalmente en citocinesis, metafase y profase, destacando la citocinesis. No obstante, en los núcleos en anafase se clasificaron correctamente una cantidad considerable de núcleos, pero menos que los clasificados de forma errónea.

Tabla 5.9: Rendimiento por predicciones positivas y negativas del modelo binario de núcleos de hepatocitos en la muestra de regeneración hepática.

Modelo	Tipo de Hepatocito											
	Normal		Profase		Metafase		Anatelofase		Citocinesis		Total	
	+	-	+	-	+	-	+	-	+	-	+	-
Binario Hepatocitos	3469	285	5	2	14	9	56	76	37	4	3581	376

5.7.1 Discusión

En esta última prueba se lograron resultados robustos, con la correcta clasificación del 90 % de los núcleos de hepatocitos y la mayoría de los núcleos de hepatocitos en mitosis. Aunque los resultados obtenidos en esta prueba podrían ser considerados contra intuitivos, permiten denotar la robustez del modelo. Esto se debe a que el modelo no solo se enfoca principalmente en la forma geométrica de los núcleos, sino también en la información proporcionada por los píxeles de intensidad del núcleo. Esto explicaría por qué, incluso cuando la forma de los núcleos de hepatocitos es diferente a la de un núcleo en estado normal, es capaz de clasificar correctamente este tipo de núcleos. Por otra parte, el rendimiento en los núcleos en anatelofase podría explicarse por las formas extremas que adoptan a lo largo de esta etapa de la mitosis, lo que a su vez también cambia la concentración de las intensidades dentro de estos núcleos, dificultando que el modelo clasifique correctamente algunos de ellos en esta etapa. Sin embargo, los resultados obtenidos en este tipo de núcleos son alentadores, ya que indican que, si se entrenara el modelo con más núcleos, se podría generar un modelo que clasifique de manera casi ideal tanto núcleos de hepatocitos normales como en mitosis. Esto otorgaría una gran robustez al modelo, permitiendo su uso tanto en muestras de tejido común como en muestras de tejido anómalo, como en la regeneración hepática.

5.8 Discusión

Los resultados obtenidos de los primeros y segundos prototipos, junto con los modelos binarios, indican que, debido al significativo desbalance de clases, las matrices de confusión del caso desbalanceado muestran una ausencia de falsos positivos para las clases minoritarias. En cambio, los modelos tienden a concentrarse en núcleos de hepatocitos y células endoteliales. A pesar de que los núcleos no clasificados tienen más datos de entrenamiento en comparación con los núcleos de células estrelladas y células de Kupffer, la baja cantidad de predicciones correctas para estos últimos sugiere un sesgo hacia los núcleos de hepatocitos y células endoteliales en el primer prototipo. Además, los pequeños sobreajustes observados en algunos de los modelos entrenados podrían ser mitigados fácilmente mediante un aumento en la cantidad de datos de entrenamiento.

Con todo lo discutido, es correcto asumir que las redes presentan un problema de generalización, especialmente para los núcleos de células estrelladas y células de Kupffer. Aunque, a pesar del balanceo de clases, los modelos logran más predicciones correctas para estos núcleos, la proporción de predicciones correctas respecto al total de núcleos de estas clases sigue siendo alta. Esto puede deberse a la similitud entre los núcleos de células estrelladas, células de Kupffer y células endoteliales. No obstante, estos resultados también confirman que, aunque el algoritmo tiene fallas, presenta una robustez considerable y obtiene resultados comparables e incluso superiores a los de un software de alto rendimiento como MT. Además, los resultados en la muestra de regeneración hepática refuerzan esta robustez, al clasificar correctamente el 90 % de los núcleos en un tejido con una morfología diferente. Esto sugiere que, si el algoritmo se entrena con más muestras y/o muestras de regeneración hepática, podría clasificar óptimamente este tipo de muestras al igual que las muestras normales, lo cual es importante para el estudio de estos tejidos.

Capítulo 6. Conclusiones

6.1 Discusión

La base de datos utilizada, aunque desbalanceada, reveló la incapacidad de los modelos para reconocer adecuadamente los núcleos de células estrelladas, células de Kupffer y núcleos no clasificados. Este problema se confirmó al balancear las clases, mostrando que, aunque algunos núcleos de células estrelladas y células de Kupffer se clasificaron correctamente, su número era muy reducido en comparación con el total. Los resultados de los modelos binarios también corroboraron esta deficiencia, ya que, aunque clasificaron una gran cantidad de estos núcleos, presentaron muchos falsos positivos.

A pesar de que el algoritmo diseñado mostró resultados comparables e incluso superiores en algunos casos respecto al software de alto rendimiento MT, se observó una deficiencia en la clasificación de ciertos tipos de núcleos. Este problema puede estar relacionado tanto con las limitaciones de las redes utilizadas como con la calidad de la base de datos. La exclusividad en el uso de la imagen de DAPI y la clasificación núcleo a núcleo se centra únicamente en la morfología del núcleo y la intensidad de píxeles, sin tener en cuenta la relación espacial entre los núcleos y su interacción con otras estructuras biológicas en el tejido. Esta limitación afecta la capacidad de las redes para distinguir adecuadamente entre los distintos tipos de núcleos.

El modelo optimizado del prototipo final, definido mediante la combinación de los mejores modelos, logró mejorar considerablemente la eficiencia. Aunque el modelo optimizado superó al modelo base en términos generales, se observó una ligera pérdida en la precisión de la predicción de los hepatocitos. Esta pérdida se atribuye a una deficiencia en la correcta clasificación de los núcleos de clases minoritarias, lo cual afecta la capacidad del modelo para identificar correctamente los hepatocitos mientras se busca reducir los falsos positivos en esta clase y otras.

Al comparar el algoritmo final con la clasificación de MT, se encontraron resultados similares e incluso mejores en ciertos tipos de núcleos, excepto en los núcleos endoteliales, donde la clasificación fue inferior, aunque aceptable. Esta inferioridad se debe a que MT utiliza canales adicionales como FLK1 y Faloidina para calcular la intensidad en diferentes distancias dentro del núcleo. La falta de estas imágenes en el modelo propuesto puede afectar la calidad de la clasificación.

Este enfoque se alinea con los objetivos del laboratorio de reducir el uso de inmunomarcadores me-

dian­te técnicas de DL. La introducción de imágenes generadas con técnicas de DL generativo [38] o la implementación de un preprocesamiento más específico podrían mejorar la calidad de la clasificación. Los resultados prometedores en la muestra de tejido hepático en regeneración demuestran la robustez del modelo, sugiriendo que tiene una alta capacidad para diferenciar entre núcleos hepáticos normales y en mitosis.

Con base en estos resultados, se puede concluir que el modelo posee una robustez considerable. Si se consideran las recomendaciones de mejorar la segmentación y aumentar el conjunto de datos, es probable que se obtengan resultados óptimos no solo en tejidos normales, sino también en tejidos anómalos como los de regeneración hepática. Además, el modelo tiene el potencial de clasificar correctamente no solo los hepatocitos en estos contextos, sino también los cuatro tipos de núcleos trabajados en esta memoria, con la posibilidad de expandir gradualmente a más tipos de núcleos a medida que se incremente el conjunto de datos.

6.2 Conclusiones

En esta memoria se abordó de manera exhaustiva la clasificación de imágenes de núcleos hepáticos en 3D mediante la implementación de técnicas avanzadas de DL. El objetivo fue mejorar la clasificación de los diferentes tipos de núcleos celulares presentes en el tejido hepático. Para ello, se adaptaron y evaluaron modelos de CNN de vanguardia en clasificación de imágenes, como ResNet50, InceptionV3 e Inception-ResNet-v2, ajustándolos al contexto específico de la clasificación de núcleos en 3D. De esto se obtuvieron cuatro grandes resultados:

1. Un algoritmo robusto capaz de clasificar diferentes tipos de núcleos en 3D presentes en el tejido hepático, con un rendimiento igual o superior al de un software de alto rendimiento como MT utilizando solo la imagen de DAPI.
2. Un modelo robusto capaz de clasificar de forma casi ideal los núcleos de hepatocitos en 3D, tanto normales como en mitosis, en una muestra de regeneración hepática utilizando solo la imagen de DAPI.
3. Los modelos generados destacan la efectividad del DL para esta tarea y sugieren que el rendimiento de los modelos puede mejorar aún más con el incremento de la cantidad de datos y una mejora en la segmentación de los núcleos.

4. Los modelos presentan una deficiencia en la clasificación de núcleos de células estrelladas y células de Kupffer.

Estos hallazgos no solo subrayan la eficacia del DL en la tarea de clasificación de núcleos hepáticos en imágenes en 3D, sino que también ofrecen un camino a seguir para superar las deficiencias de los modelos. Además, sugieren que aumentando la cantidad de datos y mejorando la segmentación, el rendimiento de estos modelos podría seguir mejorando, logrando resultados ideales. Esta perspectiva es muy alentadora, especialmente en el contexto del análisis de tejidos hepáticos anómalos. Por otro lado, este algoritmo permitirá avanzar en los objetivos del laboratorio al reducir la cantidad de inmunomarcadores necesarios para lograr una clasificación precisa. Sin estos inmunomarcadores, el software MT muestra resultados no óptimos. Por lo tanto, será posible redirigir estos inmunomarcadores hacia el estudio de otras estructuras dentro del tejido hepático, lo cual es crucial debido a la complejidad del tejido y las limitaciones en el número de tinciones que se pueden aplicar sin comprometer la calidad de las imágenes.

6.3 Trabajos Futuros

Dado los resultados obtenidos y las conclusiones derivadas, es claro cuál debería ser uno de los principales enfoques para futuras mejoras. Aunque los núcleos utilizados para balancear el conjunto de datos no estaban perfectamente segmentados, incluían la mayor parte de la información relevante, aunque no en todos los casos. Por lo tanto, sería beneficioso generar un conjunto de datos balanceado mucho mayor y bien segmentado para confirmar estas suposiciones.

Uno de los enfoques considerados para mejorar la clasificación del algoritmo implementado es la adición de grafos. Al realizar la clasificación con el algoritmo, se podría utilizar una Red Neuronal de Grafos (GNN) para obtener relaciones espaciales o de distribución de los diferentes núcleos respecto a la imagen completa. Esto permitiría refinar la clasificación de los núcleos [39] [40] [41].

Otro enfoque con gran potencial es el uso de redes que aprovechan la imagen completa, como es el caso de la U-Net. Por ejemplo, Hover-Net [42] no realiza la clasificación núcleo a núcleo como el algoritmo implementado, sino que utiliza el enfoque de encoder-decoder de U-Net [43] para analizar la imagen completa. Esto le permite considerar la relación entre los diferentes núcleos y estructuras en la imagen. Este enfoque es prometedor porque, al igual que MT, considera múltiples canales y la relación de los núcleos con el entorno, aunque debería adaptarse al contexto 3D.

Un tercer enfoque sería cambiar las redes utilizadas, combinando la capacidad de U-Net para obtener información global de la imagen con una GNN para obtener relaciones espaciales o de distribución de los núcleos en la imagen completa. Esta combinación podría generar un enfoque muy prometedor para mejorar la clasificación de núcleos [39].

Aunque el uso de grafos con una GNN es prometedor, tiene limitaciones, ya que son bastante complejos de manejar en comparación con las CNN, y el desafío se amplía al contexto 3D.

Por último, se considera que aprovechar la eficiencia y la capacidad de los modelos Transformers para reconocer patrones más complejos podría mejorar la clasificación de núcleos hepáticos. Esto podría hacerse complementando la capacidad de clasificación de una CNN [44] o utilizando un modelo Transformer para realizar la clasificación [45].

Además, es importante recordar que el enfoque principal al utilizar DAPI para la clasificación es minimizar el uso de marcadores en la inmunotinción. Esto no solo reduce costos y tiempo, sino que también permite el estudio de otras estructuras dentro del tejido hepático. Al utilizar un enfoque más avanzado como U-Net, GNN y Transformers en 3D, podríamos mantener este objetivo de reducción de marcadores mientras mejoramos la precisión y capacidad de clasificación de los modelos.

En resumen, aunque los datos actuales y los modelos utilizados han proporcionado información y resultados valiosos, es evidente que avanzar hacia redes más sofisticadas, como U-Net, GNN y Transformers, adaptadas a un entorno 3D, podría ofrecer resultados mucho mejores. Este enfoque no solo optimizaría la clasificación de los núcleos, sino que también permitiría una exploración más profunda y detallada de otras estructuras biológicas en el tejido hepático, alineándose con los objetivos de investigación del laboratorio.

Capítulo 7. Glosario

Glosario

BNC	Clasificador de Redes Bayesianas , en inglés, Bayesian Network Classifier
CNN	Red Neuronal Convolutacional , en inglés, Convolutional Neural Network
CV	Vena central , en inglés, Central Vein
2D	Dos dimensiones , en inglés, Two-dimesional
3D	Tres dimensiones , en inglés, Three-dimesional
DAPI	4', 6-diamidino-2-fenilindol , en inglés, 4', 6-diamidino-2-phenylindole
DL	Aprendizaje profundo , en inglés, Deep Learning
FN	Falsos negativos , en inglés, False Negative
FP	Falsos positivos , en inglés, False Positive
GaAsp	Fosfuro de arseniuro de galio , en inglés, Gallium Arsenide Phosphide
GNN	Red Neuronal de Grafos , en inglés, Graph Neural Network
GT	Verdad de referencia , en inglés, Ground Truth
LDA	Análisis Discriminante Lineal , en inglés, Linear Discriminant Analysis
Leaky ReLU	Unidad Lineal Rectificada con Fuga , en inglés, Leaky Rectified Linear Unit
ML	Aprendizaje automatico , en inglés, Machine Learning
MT	MotionTracking
PV	Vena porta , en inglés, Portal Vein
ReLU	Unidad Lineal Rectificada , en inglés, Rectified Linear Unit
RMSprop	Propagación de la media cuadrática , en inglés, Root Mean Square Propagation
TN	Verdaderos negativos , en inglés, True Negative
TP	Verdaderos positivos , en inglés, True Positive

Capítulo 8. Referencias

- [1] 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, 2015. [Online]. Available: <https://doi.org/10.7554/eLife.11214>
- [2] XXIX, “The anatomy and physiology of the liver,” *Philosophical Transactions of the Royal Society of London*, vol. 123, pp. 711–770, Dec. 1833.
- [3] 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. 8, pp. 379–456, Nov. 1949.
- [4] R. S. McCuskey, “The hepatic microvascular system in health and its response to toxicants,” *The Anatomical Record: Advances in Integrative Anatomy and Evolutionary Biology*, vol. 291, no. 6, pp. 661–671, 2008.
- [5] W. Pawlina and M. H. Ross, *Histology: A Text and Atlas: With Correlated Cell and Molecular Biology*, 8th ed. Wolters Kluwer, 2018.
- [6] 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, p. 2823, 2024. [Online]. Available: <https://doi.org/10.1038/s41598-024-53309-4>
- [7] F. Segovia-Miranda, A. Scholich, S. Syga *et al.*, “Quantification of nematic cell polarity in three-dimensional tissues,” *PLOS Computational Biology*, vol. 17, no. 8, p. e1008412, 2021.
- [8] Y. LeCun, Y. Bengio, and G. Hinton, “Deep learning,” *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.
- [9] M. T. Ke, S. Fujimoto, and T. Imai, “Seedb: A simple and morphology-preserving optical clearing agent for neuronal circuit reconstruction,” *Nature neuroscience*, vol. 16, pp. 1154–1161, 2013. [Online]. Available: <https://doi.org/10.1038/nn.3447>
- [10] H. Morales-Navarrete, H. Nonaka, A. Scholich, F. Segovia-Miranda, W. de Back, K. Meyer, R. L. Bogorad, V. Kotliansky, L. Brusch, Y. Kalaidzidis, F. Jülicher, B. M. Friedrich, and M. Zerial,

- “Liquid-crystal organization of liver tissue,” *eLife*, vol. 8, p. e44860, 2019. [Online]. Available: <https://doi.org/10.7554/eLife.44860>
- [11] 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,” *Nature methods*, vol. 9, no. 7, pp. 676–682, 2012. [Online]. Available: <https://doi.org/10.1038/nmeth.2019>
- [12] F. Luisier, C. Vonesch, T. Blu, and M. Unser, “Fast interscale wavelet denoising of poisson-corrupted images,” *Signal Processing*, vol. 90, pp. 415–427, 2010. [Online]. Available: <https://doi.org/10.1016/j.sigpro.2009.07.009>
- [13] 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, p. 14836, 2017. [Online]. Available: <https://doi.org/10.1038/ncomms14836>
- [14] N. Contributors, “Napari: a multi-dimensional image viewer for python,” 2019.
- [15] D. Bauer, “Napari nd-annotator,” [Online]. Available: <https://github.com/bauerdavid/napari-nD-annotator>. [Accessed:10/03/2024].
- [16] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” 2015. [Online]. Available: <https://arxiv.org/abs/1512.03385>
- [17] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, “Rethinking the inception architecture for computer vision,” 2015. [Online]. Available: <https://arxiv.org/abs/1512.00567>
- [18] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. Alemi, “Inception-v4, inception-resnet and the impact of residual connections on learning,” 2016. [Online]. Available: <https://arxiv.org/abs/1602.07261>
- [19] G. Van Rossum and F. L. Drake, *Python 3 Reference Manual*. Scotts Valley, CA: CreateSpace, 2009.
- [20] 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/>

- [21] P. Ongsulee, “Artificial intelligence, machine learning and deep learning,” 2017.
- [22] L. Saba, M. Biswas, V. Kuppili, E. C. Godia, H. S. Suri, D. R. Edla, T. Omerzu, J. R. Laird, N. N. Khanna, S. Mavrogeni, A. Protogerou, P. P. Sfikakis, V. Viswanathan, G. D. Kitas, A. Nicolaides, A. Gupta, and J. S. Suri, “The present and future of deep learning in radiology,” *European Journal of Radiology*, vol. 114, pp. 14–24, 2019.
- [23] N. Aloysius and M. Geetha, “A review on deep convolutional neural networks,” 2017.
- [24] R. Yamashita, M. Nishio, R. Do, and K. Togashi, “Convolutional neural networks: an overview and application in radiology,” *Insights into Imaging*, vol. 9, no. 4, pp. 611–629, 2018.
- [25] L. Alzubaidi, J. Zhang, A. J. Humaidi *et al.*, “Review of deep learning: concepts, cnn architectures, challenges, applications, future directions,” *Journal of Big Data*, vol. 8, p. 53, 2021.
- [26] A. K. Dubey and V. Jain, “Comparative study of convolution neural network’s relu and leaky-relu activation functions,” in *Applications of Computing, Automation and Wireless Systems in Electrical Engineering*, ser. Lecture Notes in Electrical Engineering, S. Mishra, Y. Sood, and A. Tomar, Eds. Singapore: Springer, 2019, vol. 553.
- [27] G. Litjens, T. Kooi, B. E. Bejnordi, A. A. A. Setio, F. Ciompi, M. Ghafoorian, J. A. W. M. van der Laak, B. van Ginneken, and C. I. Sánchez, “A survey on deep learning in medical image analysis,” *Medical Image Analysis*, vol. 42, pp. 60–88, 2017.
- [28] S. Ruder, “An overview of gradient descent optimization algorithms,” 2017. [Online]. Available: <https://arxiv.org/abs/1609.04747>
- [29] R. Moradi, R. Berangi, and B. Minaei, “A survey of regularization strategies for deep models,” *Artificial Intelligence Review*, vol. 53, no. 6, pp. 3947–3986, 2020.
- [30] A. A. Taha and A. Hanbury, “Metrics for evaluating 3d medical image segmentation: analysis, selection, and tool,” *BMC Medical Imaging*, vol. 15, p. 29, 2015.
- [31] H. Zhang, C. Liu, Z. Zhang, Y. Xing, X. Liu, R. Dong, Y. He, L. Xia, and F. Liu, “Recurrence plot-based approach for cardiac arrhythmia classification using inception-resnet-v2,” *Frontiers in Physiology*, vol. 12, no. 5, 2021.

- [32] H. Kim, A. Cosa-Linan, N. Santhanam *et al.*, “Transfer learning for medical image classification: a literature review,” *BMC Medical Imaging*, vol. 22, p. 69, 2022.
- [33] D. Čapek, M. Safroshkin, H. Morales-Navarrete *et al.*, “Embryonet: using deep learning to link embryonic phenotypes to signaling pathways,” *Nature Methods*, vol. 20, pp. 815–823, 2023.
- [34] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, “Imagenet: A large-scale hierarchical image database,” in *2009 IEEE Conference on Computer Vision and Pattern Recognition*, 2009, pp. 248–255.
- [35] M. Jan, A. Spangaro, M. Lenartowicz, and M. Mattiazzi Usaj, “From pixels to insights: Machine learning and deep learning for bioimage analysis,” *BioEssays*, vol. 46, p. e2300114, 2024.
- [36] R. Solovyev, A. A. Kalinin, and T. Gabruseva, “3d convolutional neural networks for stalled brain capillary detection,” *Computers in Biology and Medicine*, vol. 141, no. Elsevier, p. 105089, 2022.
- [37] 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,” *Scientific Reports*, vol. 13, no. 1, p. 9533, 2023.
- [38] N. Bettancourt, C. Pérez-Gallardo, V. Candia, P. Guevara, Y. Kalaidzidis, M. Zerial *et al.*, “Virtual tissue microstructure reconstruction across species using generative deep learning,” *PLoS ONE*, vol. 19, no. 7, p. e030673, 2024. [Online]. Available: <https://doi.org/10.1371/journal.pone.030673>
- [39] J. Pérez-Cano, I. S. Valero, D. Anglada-Rotger, O. Pina, P. Salembier, and F. Marques, “Combining graph neural networks and computer vision methods for cell nuclei classification in lung tissue,” *Heliyon*, vol. 10, no. 7, p. e28463, 2024. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S2405844024044943>
- [40] S. Brussee, G. Buzzanca, A. M. R. Schrader, and J. Kers, “Graph neural networks in histopathology: Emerging trends and future directions,” *arXiv preprint arXiv:2406.12808*, 2024. [Online]. Available: <https://arxiv.org/abs/2406.12808>
- [41] J. Sims, H. I. Grabsch, and D. Magee, “Using hierarchically connected nodes and multiple gnn message passing steps to increase the contextual information in cell-graph classification,” in *Imaging Systems for GI Endoscopy, and Graphs in Biomedical Image Analysis*, L. Manfredi, S.-A. Ahmadi, M. Bronstein, A. Kazi, D. Lomanto, A. Mathew, L. Magerand, K. Mullakaeva, B. Papiez, R. H. Taylor, and E. Trucco, Eds. Cham: Springer Nature Switzerland, 2022, pp. 99–107.

- [42] G. Litjens, T. Kooi, B. E. Bejnordi, A. A. A. Setio, F. Ciompi, M. Ghafoorian, J. A. W. M. van der Laak, B. van Ginneken, and C. I. Sánchez, “A survey on deep learning in medical image analysis,” *Medical Image Analysis*, vol. 42, pp. 60–88, 2017.
- [43] O. Ronneberger, P. Fischer, and T. Brox, “U-net: Convolutional networks for biomedical image segmentation,” *arXiv preprint arXiv:1505.04597*, 2015. [Online]. Available: <https://arxiv.org/abs/1505.04597>
- [44] Y. Chen, Y. Jia, X. Zhang, J. Bai, X. Li, M. Ma, Z. Sun, and Z. Pei, “Tshvnet: Simultaneous nuclear instance segmentation and classification in histopathological images based on multiattention mechanisms,” *Biomedical Research International*, vol. 2022, p. 7921922, Nov 2022.
- [45] J. Huang, H. Li, X. Wan, and G. Li, “Affine-consistent transformer for multi-class cell nuclei detection,” *arXiv preprint arXiv:2310.14154*, 2023. [Online]. Available: <https://arxiv.org/abs/2310.14154>

Anexo A - Resultados Primer Prototipo

A continuación se presentan las curvas de pérdida y F1-Score para los resultados obtenidos con el primer prototipo para ResNet50, InceptionV3 e Inception-ResNet-v2 en las Fig. A.1 A.2 y A.3, respectivamente.

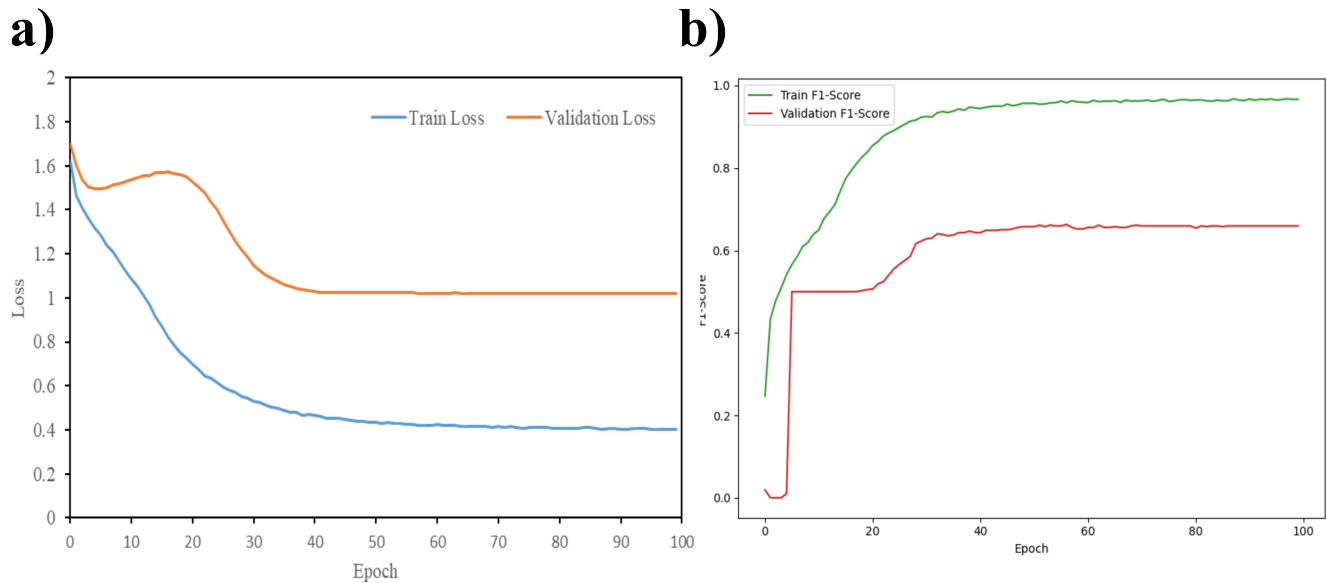


Fig. A.1: Curvas de pérdida y F1-Score del modelo de 5 clases con ResNet50.

a) Curva de pérdida del entrenamiento de ResNet50, b) Curva F1-Score del entrenamiento de ResNet50.

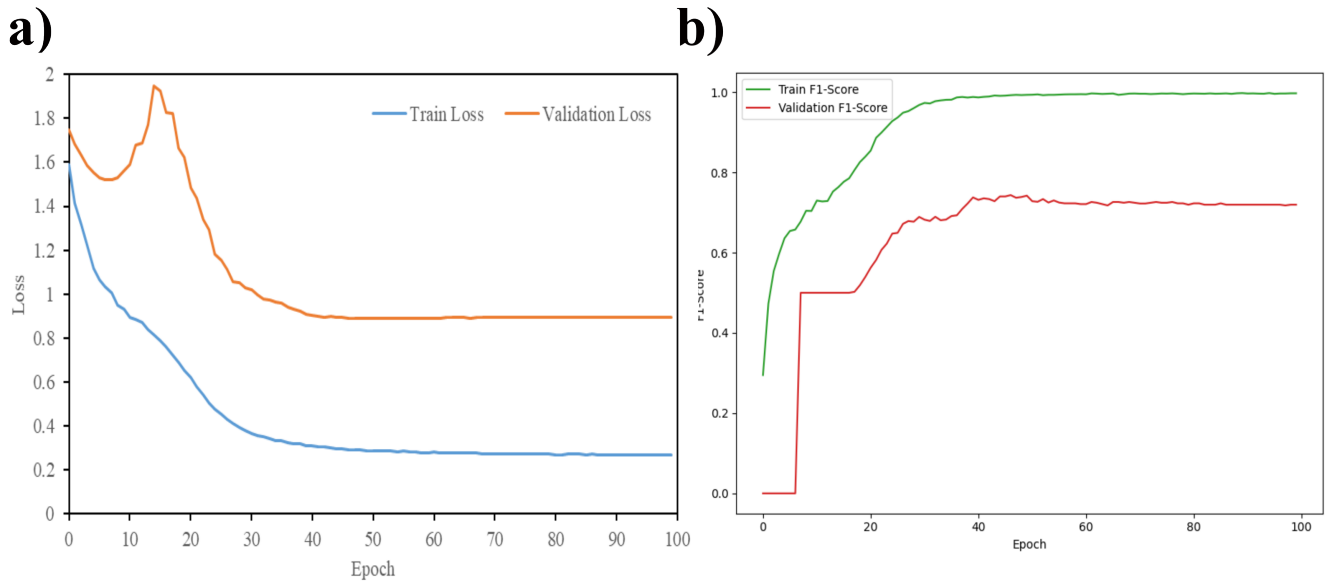


Fig. A.2: Curvas de pérdida y F1-Score del modelo de 5 clases con InceptionV3.

a) Curva de pérdida del entrenamiento de InceptionV3, b) Curva F1-Score del entrenamiento de InceptionV3.

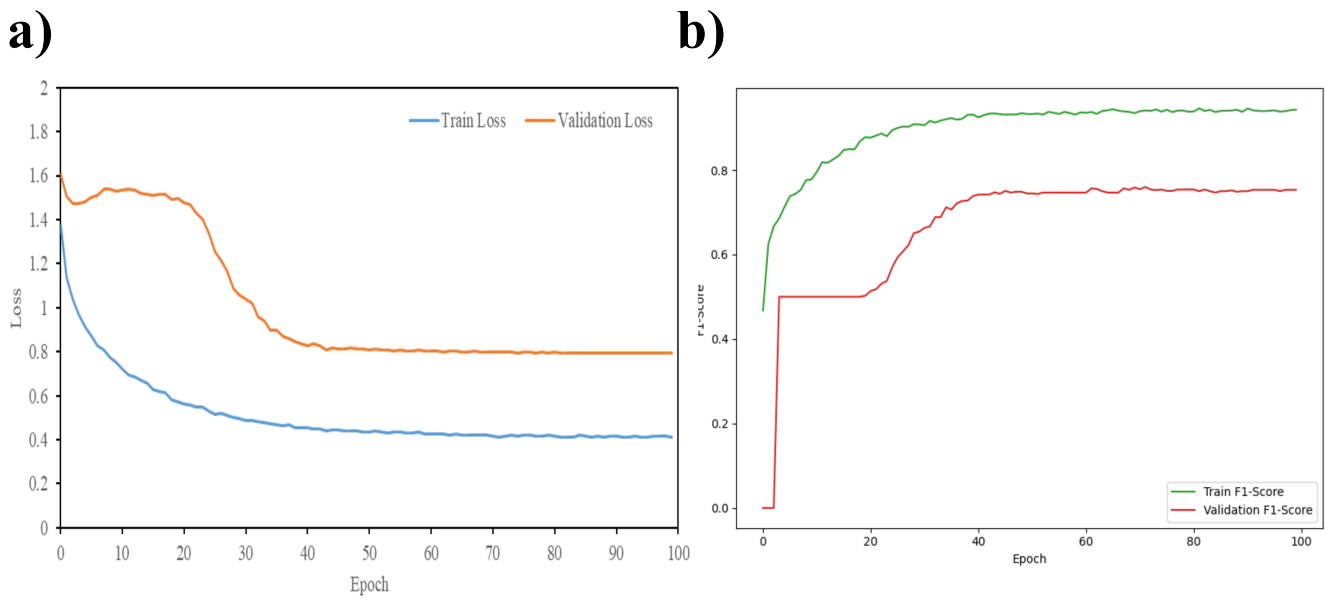


Fig. A.3: Curvas de pérdida y F1-Score del modelo de 5 clases con Inception-ResNet-v2.

a) Curva de pérdida del entrenamiento de Inception-ResNet-v2, b) Curva F1-Score del entrenamiento de Inception-ResNet-v2.

A continuación se aprecian las matrices de confusión para los tres modelos obtenidas para el primer prototipo con el conjunto de prueba se aprecian en las Fig. A.4, A.5 y A.6.

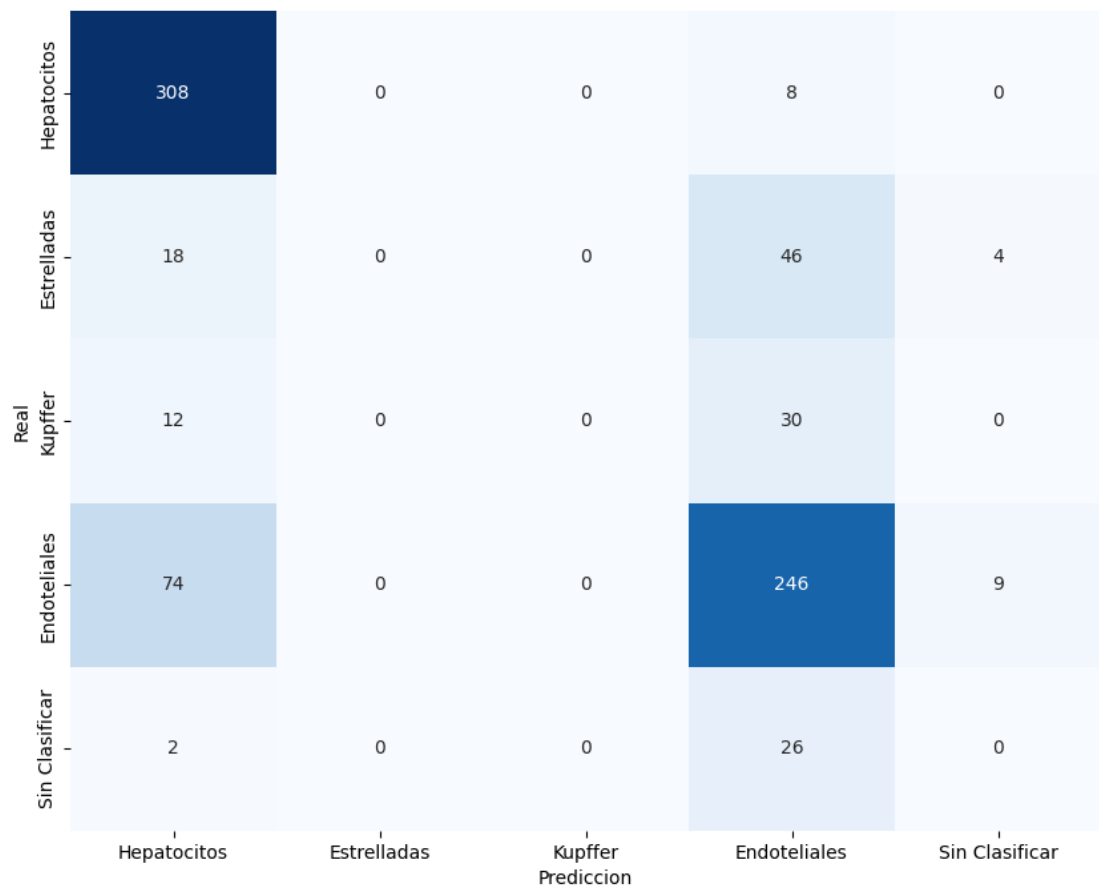


Fig. A.4: Matriz de confusión en el conjunto de prueba con el modelo de 5 clases con ResNet50.

Real	Hepatocitos	312	0	0	4	0
	Estrelladas	15	3	0	43	7
	Kupffer	8	1	0	31	2
	Endoteliales	34	3	1	272	19
	Sin Clasificar	0	0	0	25	3
		Prediccion				
		Hepatocitos	Estrelladas	Kupffer	Endoteliales	Sin Clasificar

Fig. A.5: Matriz de confusión en el conjunto de prueba con el modelo de 5 clases con InceptionV3.

Real	Hepatocitos	313	0	0	3	0
	Estrelladas	15	6	0	36	11
	Kupffer	9	0	1	27	5
	Endoteliales	31	4	5	232	57
	Sin Clasificar	1	0	0	25	2
		Prediccion				
		Hepatocitos	Estrelladas	Kupffer	Endoteliales	Sin Clasificar

Fig. A.6: Matriz de confusión en el conjunto de prueba con el modelo de 5 clases con Inception-ResNet-v2.

Anexo B - Resultados Segundo Prototipo

A continuación se presentan las curvas de pérdida y F1-Score para los resultados obtenidos con el segundo prototipo para ResNet50, InceptionV3 e Inception-ResNet-v2 en las Fig. B.1 B.2 y B.3, respectivamente.

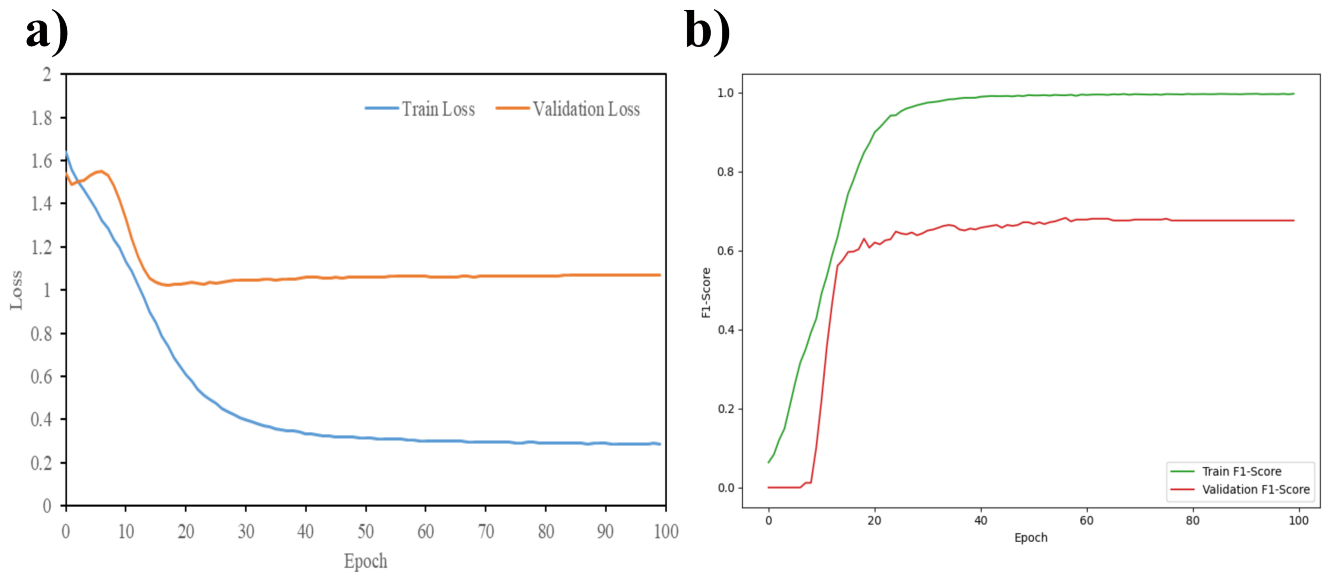


Fig. B.1: Curvas de pérdida y F1-Score del modelo de 4 clases con ResNet50.

a) Curva de pérdida del entrenamiento de ResNet50, b) Curva F1-Score del entrenamiento de ResNet50.

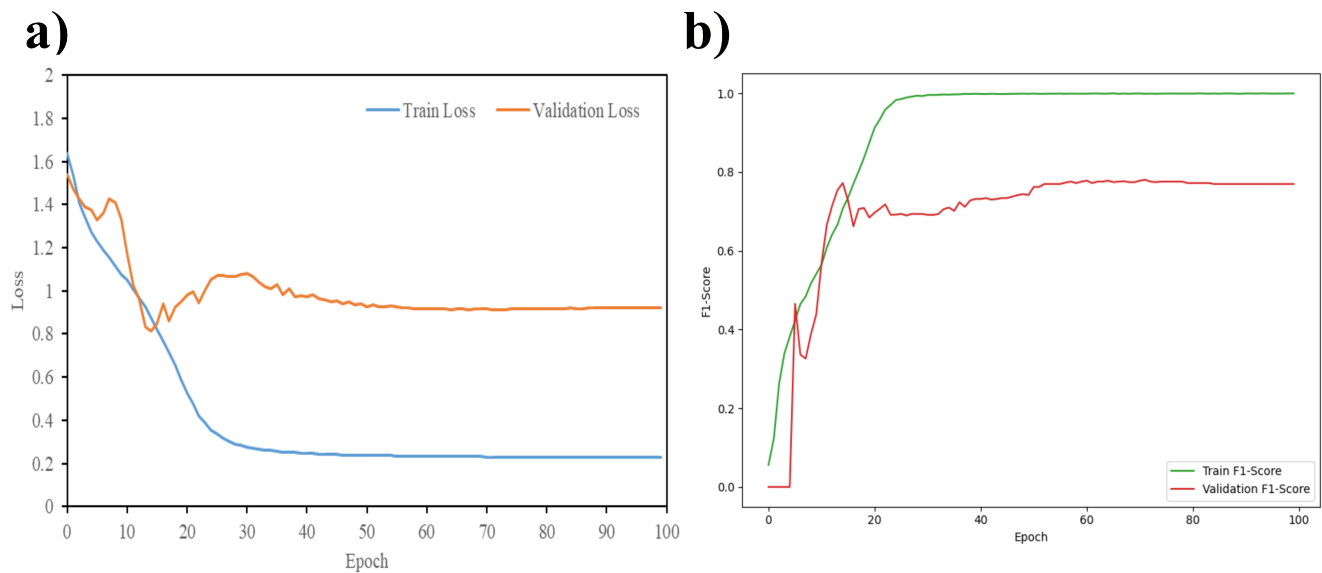


Fig. B.2: Curvas de pérdida y F1-Score del modelo de 4 clases con InceptionV3.

a) Curva de pérdida del entrenamiento de InceptionV3, b) Curva F1-Score del entrenamiento de InceptionV3.

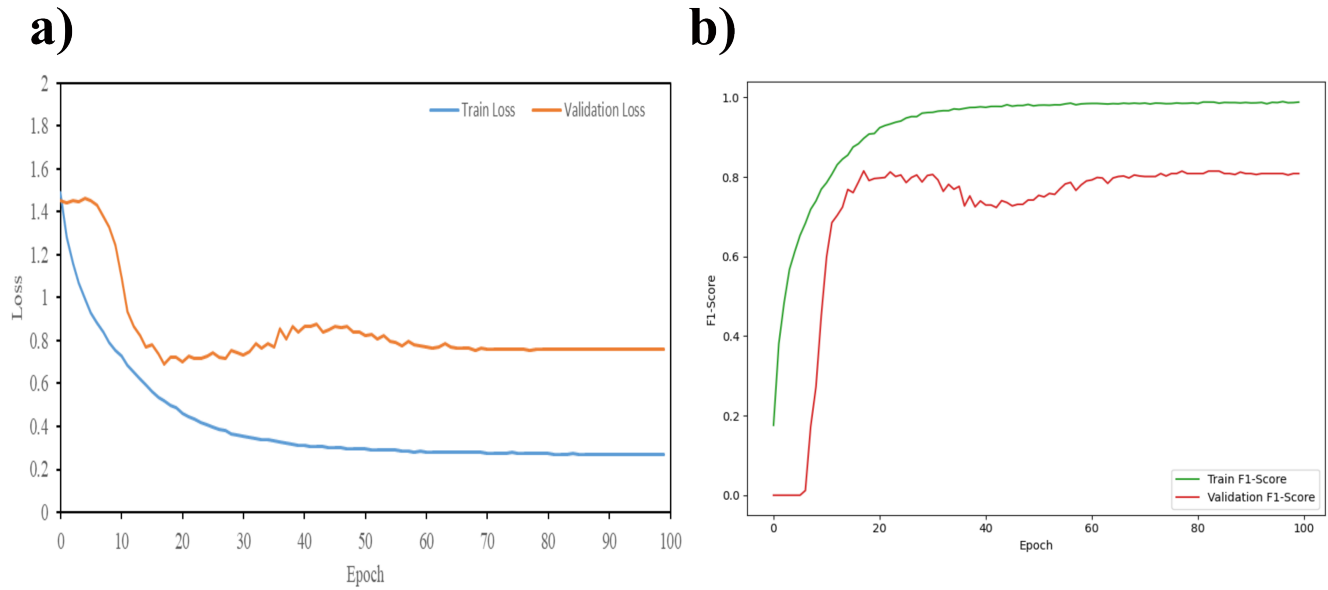


Fig. B.3: Curvas de pérdida y F1-Score del modelo de 4 clases con Inception-ResNet-v2.

a) Curva de pérdida del entrenamiento de Inception-ResNet-v2, b) Curva F1-Score del entrenamiento de Inception-ResNet-v2.

A continuación se aprecian las matrices de confusión para los tres modelos obtenidas para el segundo prototipo con el conjunto de prueba se aprecian en las Fig. B.4, B.5 y B.6.

Real	Hepatocitos	Estrelladas	Kupffer	Endoteliales	
	298	0	17	1	
	17	9	19	23	
	11	5	10	16	
Endoteliales	77	59	86	107	
		Predicción			
		Hepatocitos	Estrelladas	Kupffer	Endoteliales

Fig. B.4: Matriz de confusión en el conjunto de prueba con el modelo de 4 clases con ResNet50.

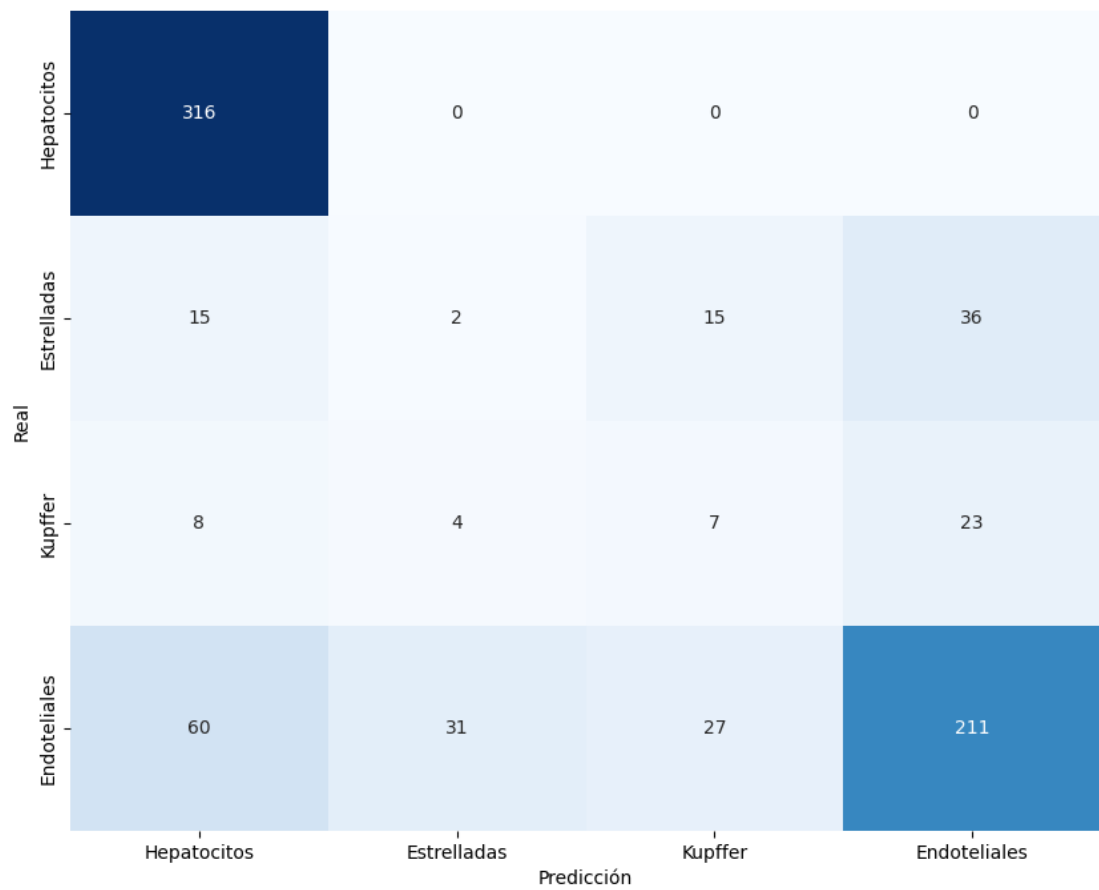


Fig. B.5: Matriz de confusión en el conjunto de prueba con el modelo de 4 clases con InceptionV3.

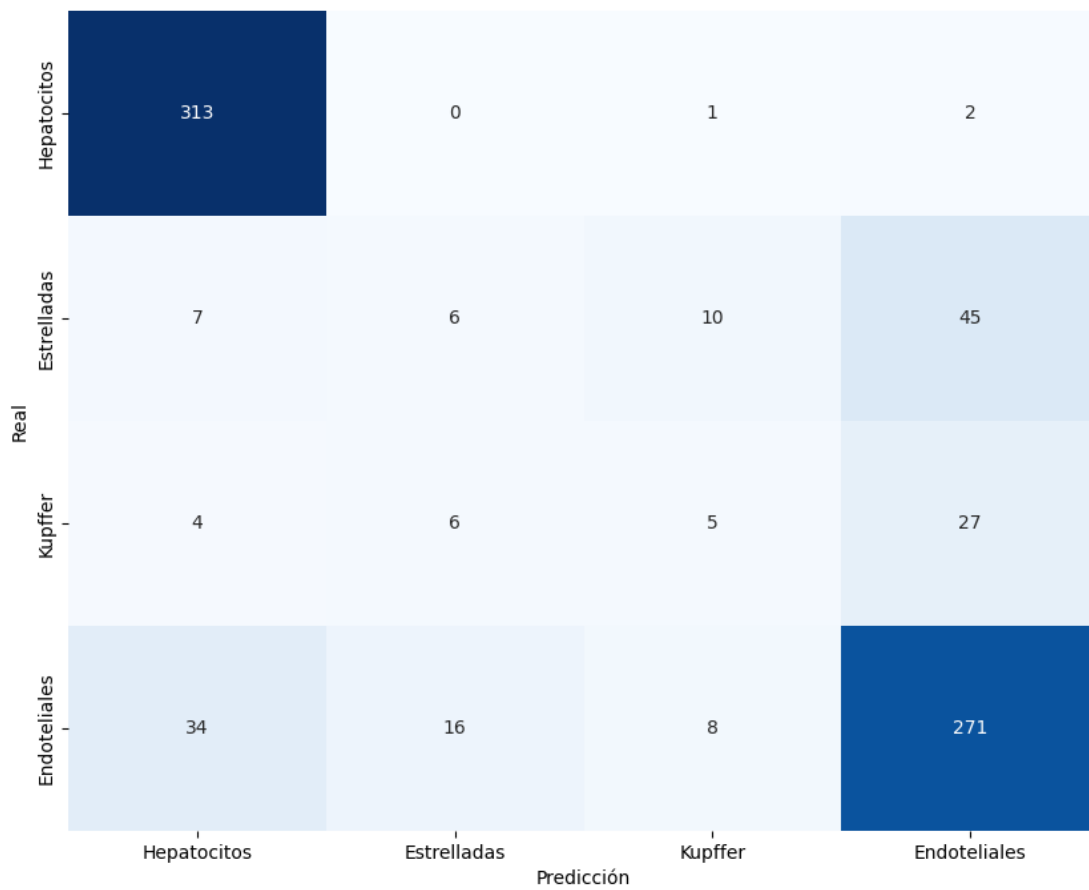


Fig. B.6: Matriz de confusión en el conjunto de prueba con el modelo de 4 clases con Inception-ResNet-v2.

Anexo C - Resultados Modelos Binarios

C.1 Modelo Binario de Núcleos de Hepatocitos

A continuación se presentan las curvas de pérdida y exactitud binaria para los modelos binarios de núcleos de hepatocitos para ResNet50, InceptionV3 e Inception-ResNet-v2 en las Fig. C.1, C.2 y C.3, respectivamente.

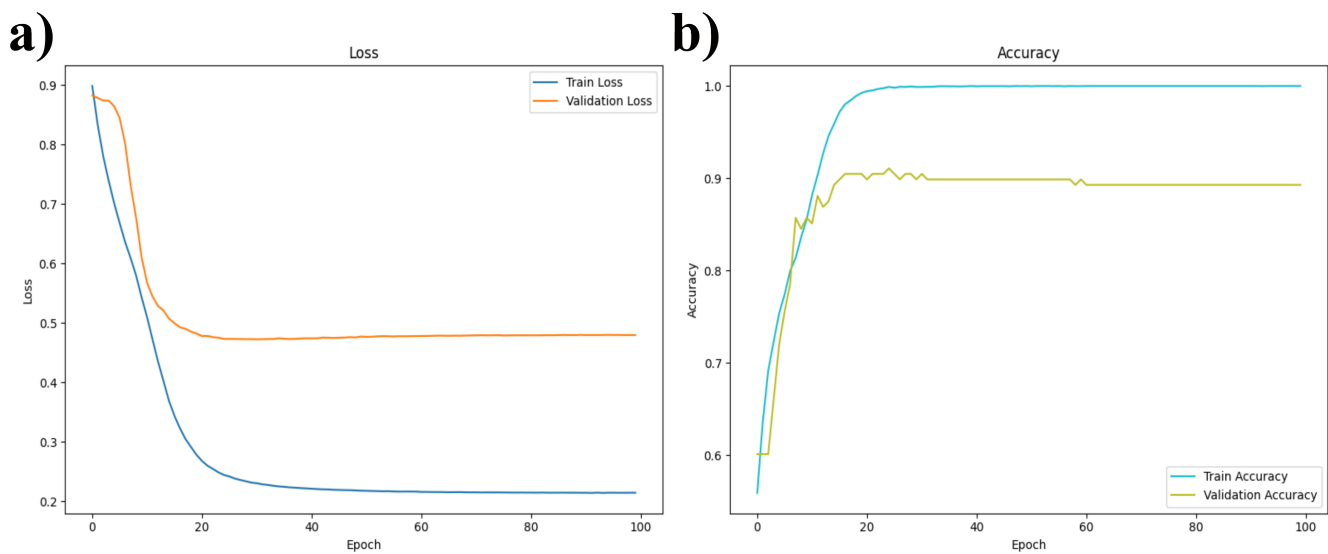


Fig. C.1: Curvas de pérdida y exactitud binaria del modelo binario de núcleos de hepatocitos con ResNet50.

a) Curva de pérdida del entrenamiento de ResNet50, b) Curva de exactitud binaria del entrenamiento de ResNet50.

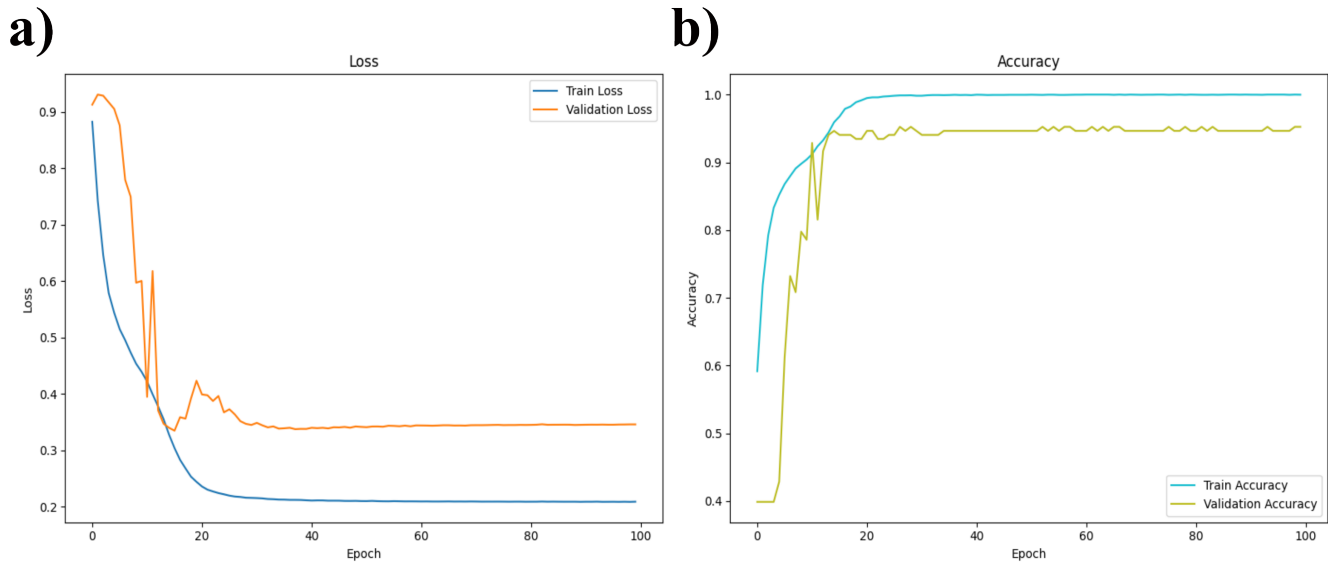


Fig. C.2: Curvas de pérdida y exactitud binaria del modelo binario de núcleos de hepatocitos con InceptionV3.

a) Curva de pérdida del entrenamiento de InceptionV3, b) Curva de exactitud binaria del entrenamiento de InceptionV3.

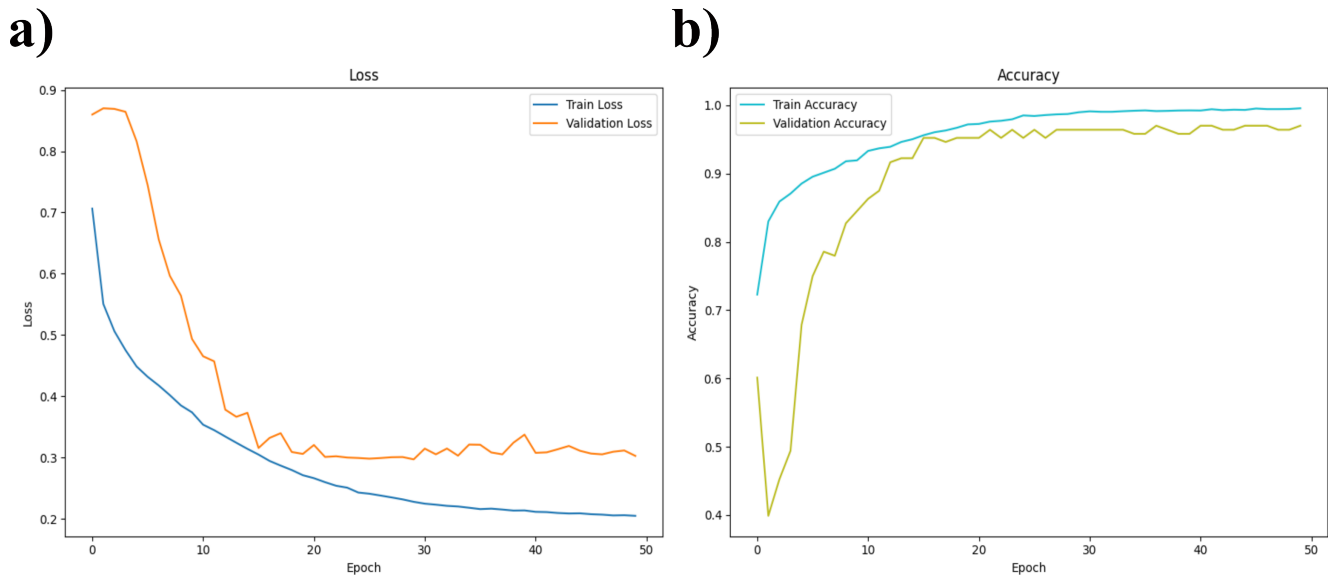


Fig. C.3: Curvas de pérdida y exactitud binaria del modelo binario de núcleos de hepatocitos con Inception-ResNet-v2.

a) Curva de pérdida del entrenamiento de Inception-ResNet-v2, b) Curva de exactitud binaria del entrenamiento de Inception-ResNet-v2.

A continuación se aprecian las matrices de confusión para los tres modelos binarios con el conjunto de prueba se pueden apreciar en las Fig. C.4, C.5 y C.6.

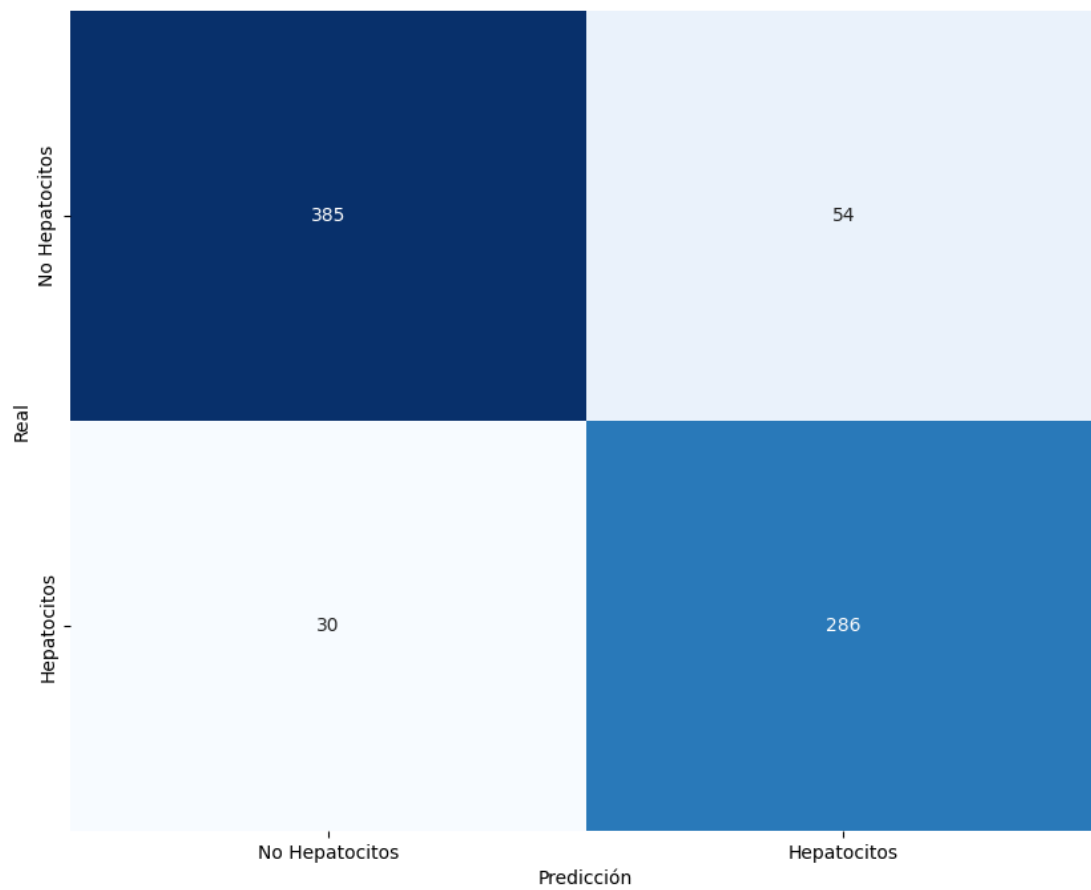


Fig. C.4: Matriz de confusión en el conjunto de prueba con el modelo binario de núcleos de hepatocitos con ResNet50.

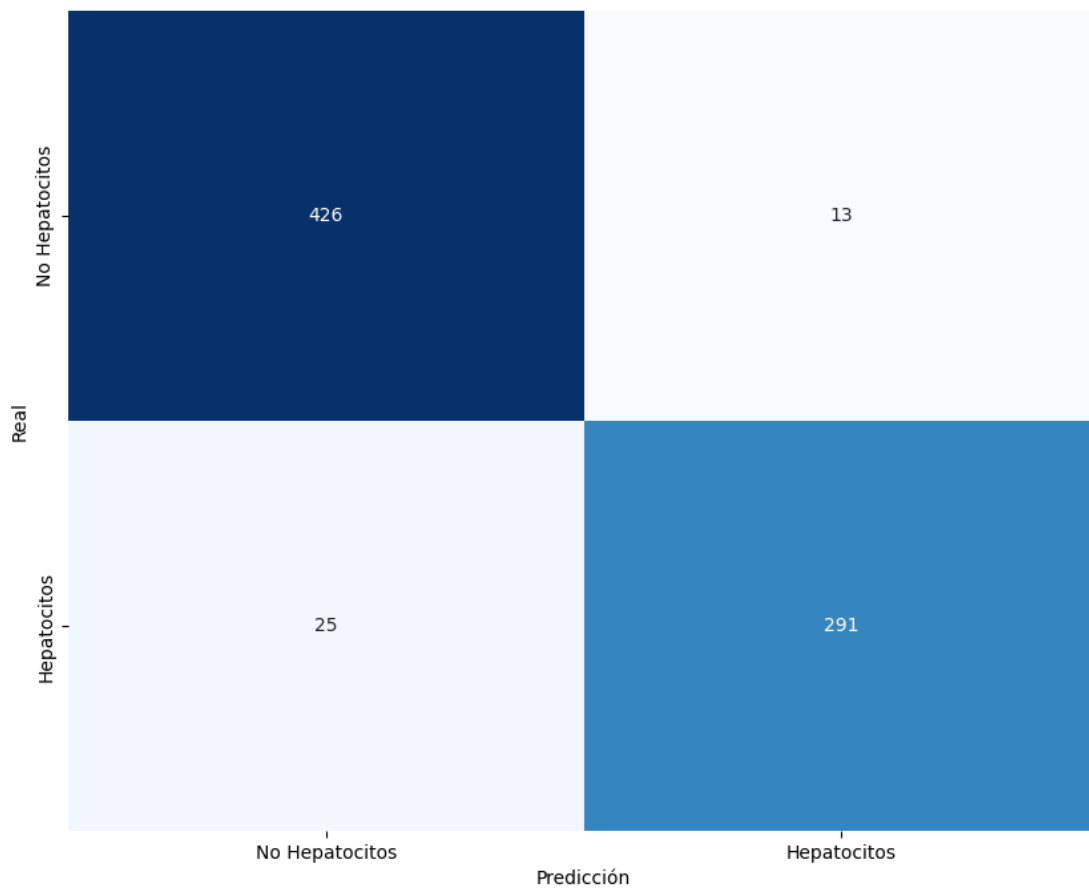


Fig. C.5: Matriz de confusión en el conjunto de prueba con el modelo binario de núcleos de hepatocitos con InceptionV3.

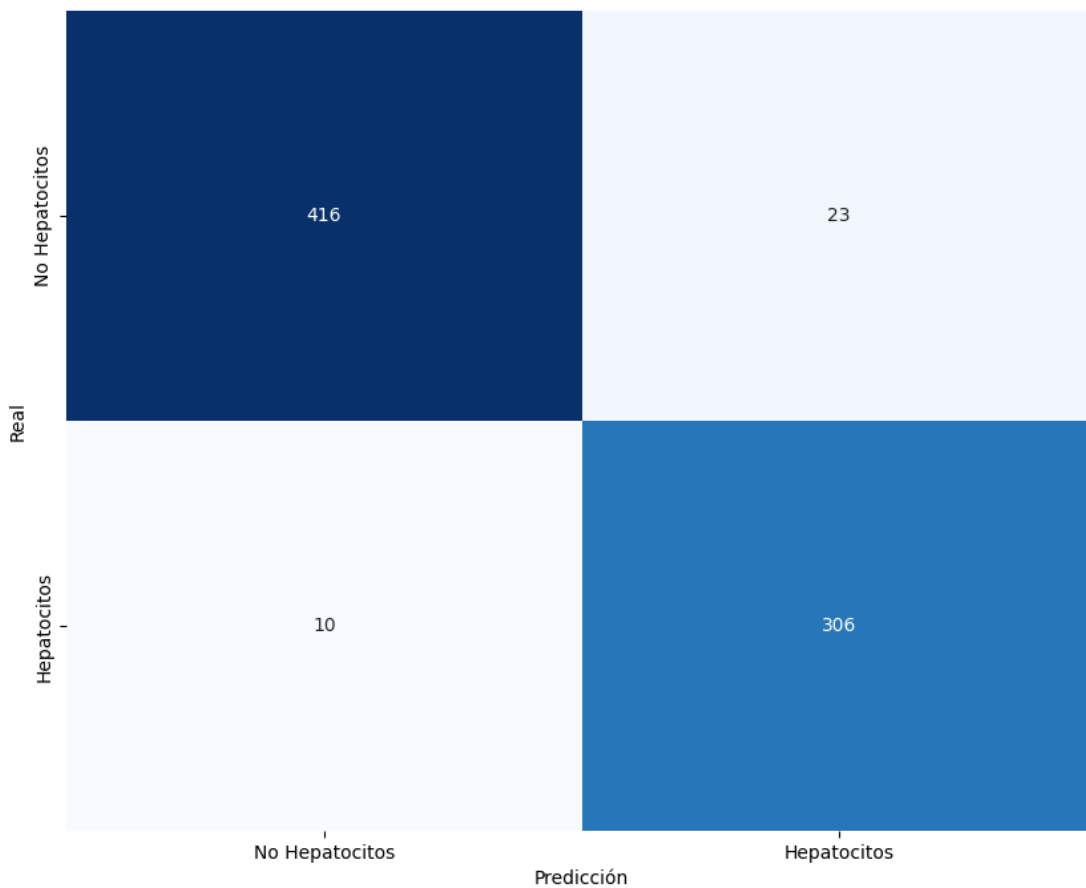


Fig. C.6: Matriz de confusión en el conjunto de prueba con el modelo binario de núcleos de hepatocitos con Inception-ResNet-v2.

C.2 Modelo Binario de Núcleos de Células Estrelladas

A continuación se presentan las curvas de pérdida y exactitud binaria para los modelos binarios de núcleos de células estrelladas para ResNet50, InceptionV3 e Inception-ResNet-v2 en las Fig. C.7, C.8 y C.9, respectivamente.

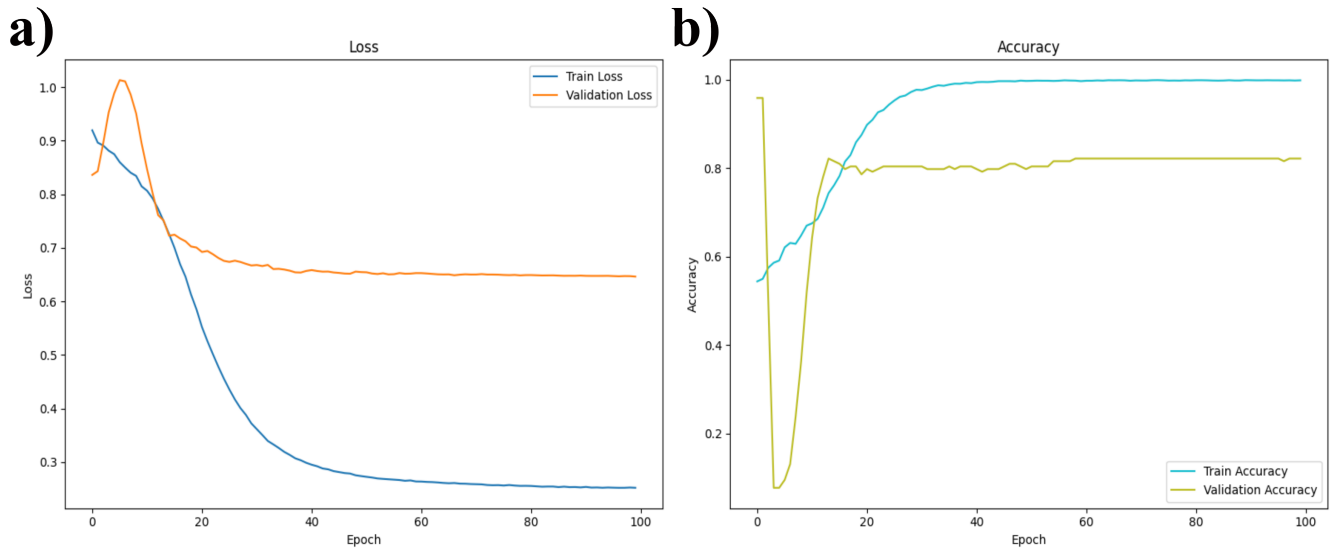


Fig. C.7: Curvas de pérdida y exactitud binaria del modelo binario de núcleos de células estrelladas con ResNet50.

a) Curva de pérdida del entrenamiento de ResNet50, b) Curva de exactitud binaria del entrenamiento de ResNet50.

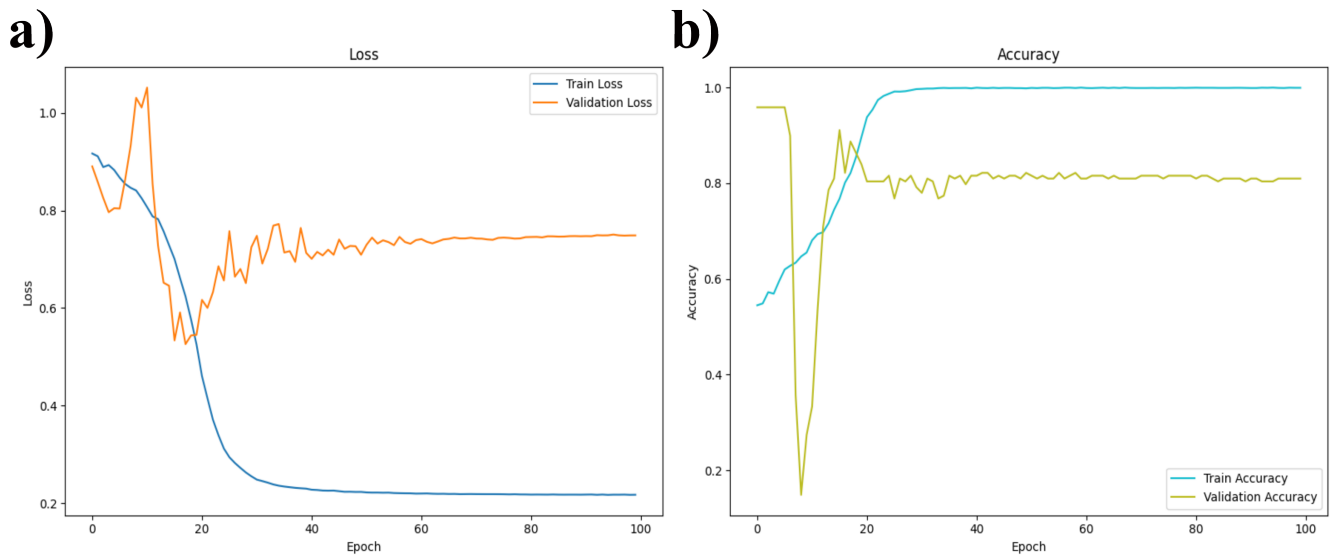
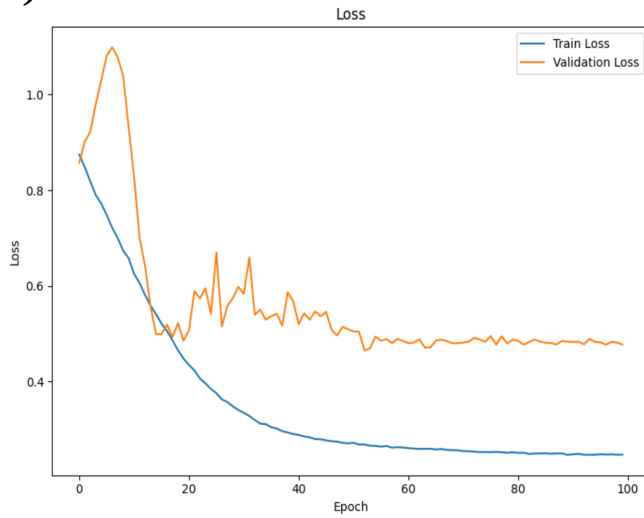


Fig. C.8: Curvas de pérdida y exactitud binaria del modelo binario de núcleos de células estrelladas con InceptionV3.

a) Curva de pérdida del entrenamiento de InceptionV3, b) Curva de exactitud binaria del entrenamiento de InceptionV3.

a)



b)

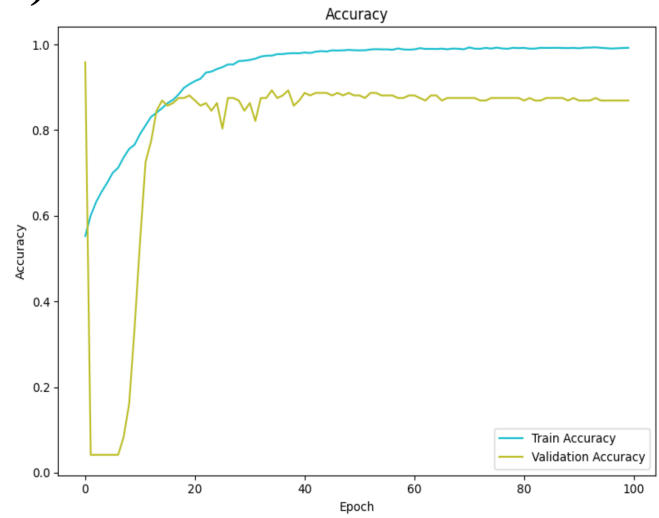


Fig. C.9: Curvas de pérdida y exactitud binaria del modelo binario de núcleos de células estrelladas con Inception-ResNet-v2.

a) Curva de pérdida del entrenamiento de Inception-ResNet-v2, b) Curva de exactitud binaria del entrenamiento de Inception-ResNet-v2.

A continuación se aprecian las matrices de confusión para los tres modelos binarios con el conjunto de prueba se pueden apreciar en las Fig. C.10, C.11 y C.12.

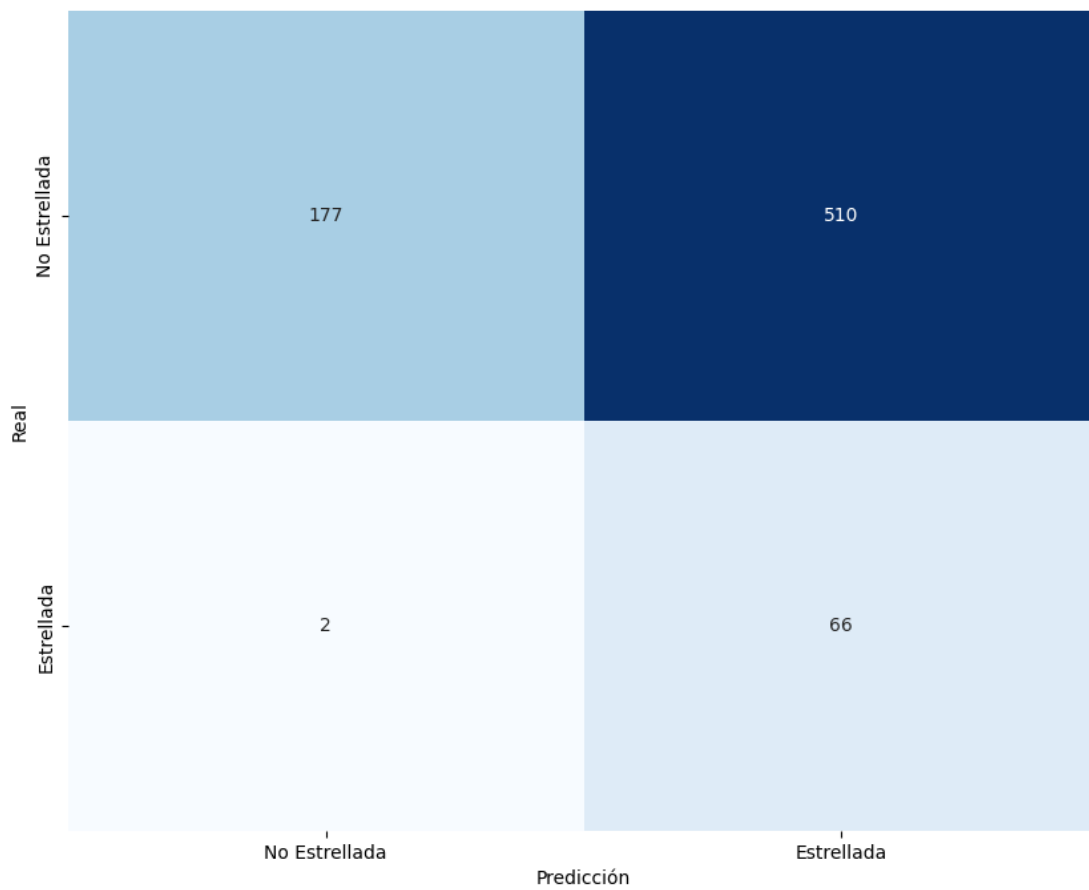


Fig. C.10: Matriz de confusión en el conjunto de prueba con el modelo binario de núcleos de células estrelladas con ResNet50.

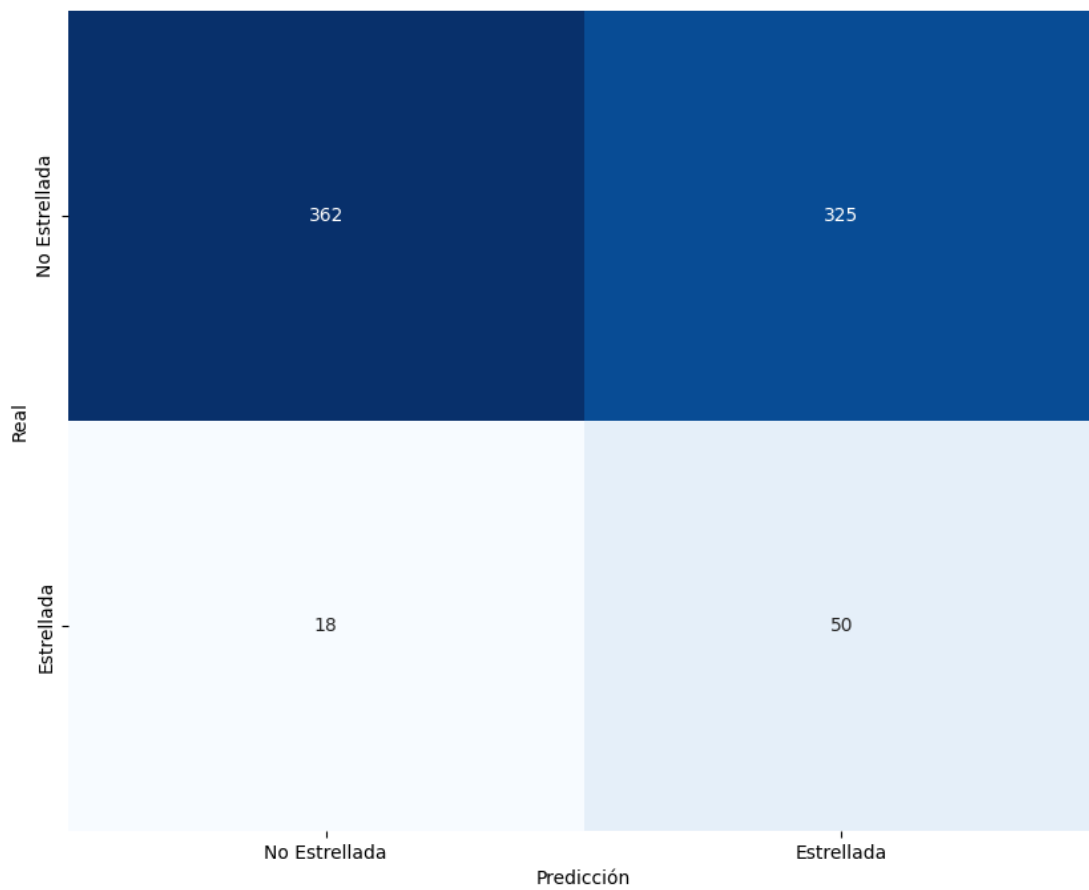


Fig. C.11: Matriz de confusión en el conjunto de prueba con el modelo binario de núcleos de células estrelladas con InceptionV3.

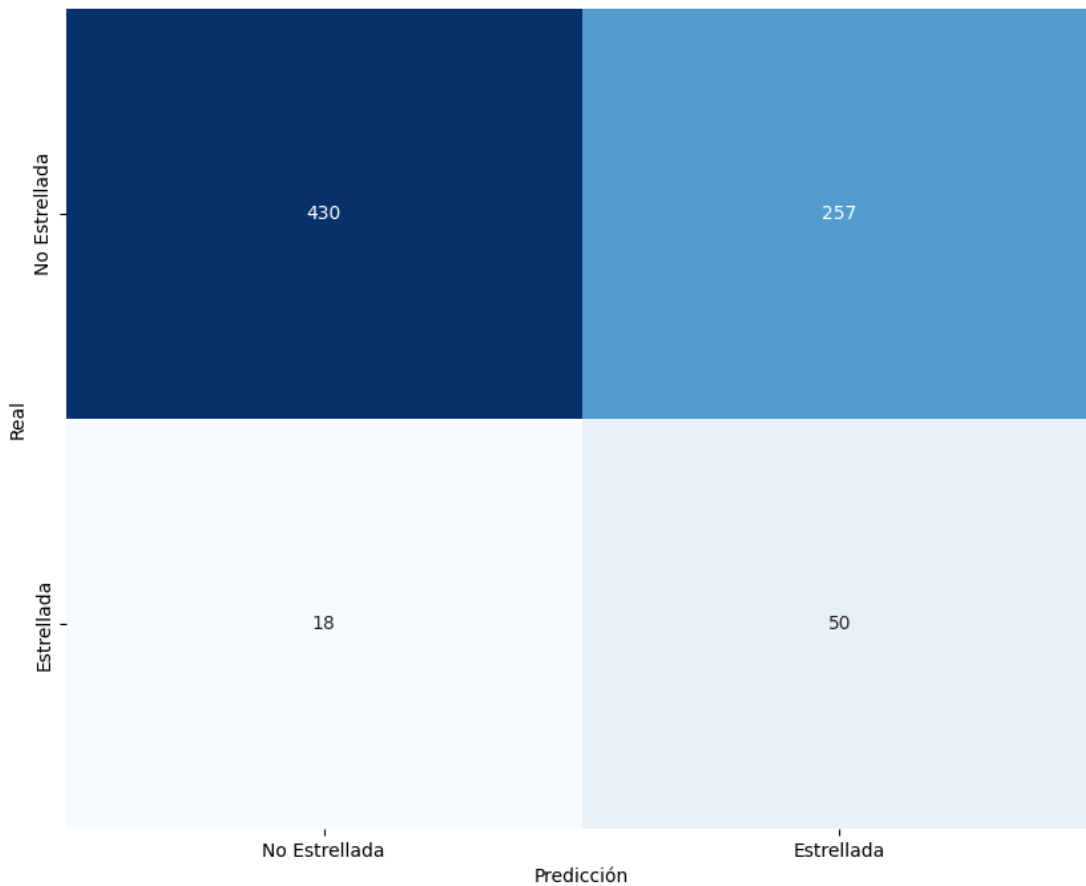
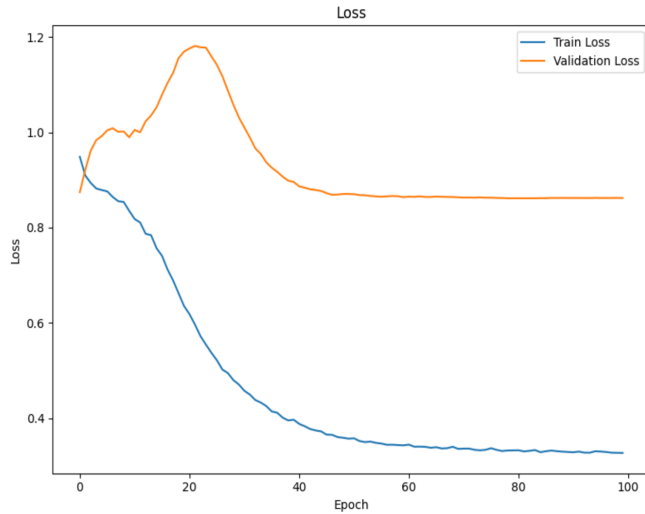


Fig. C.12: Matriz de confusión en el conjunto de prueba con el modelo binario de núcleos de células estrelladas con Inception-ResNet-v2.

C.3 Modelo Binario de Núcleos de Células de Kupffer

A continuación se presentan las curvas de pérdida y exactitud binaria para los modelos binarios de núcleos de células de Kupffer para ResNet50, InceptionV3 e Inception-ResNet-v2 en las Fig. C.13, C.14 y C.15, respectivamente.

a)



b)

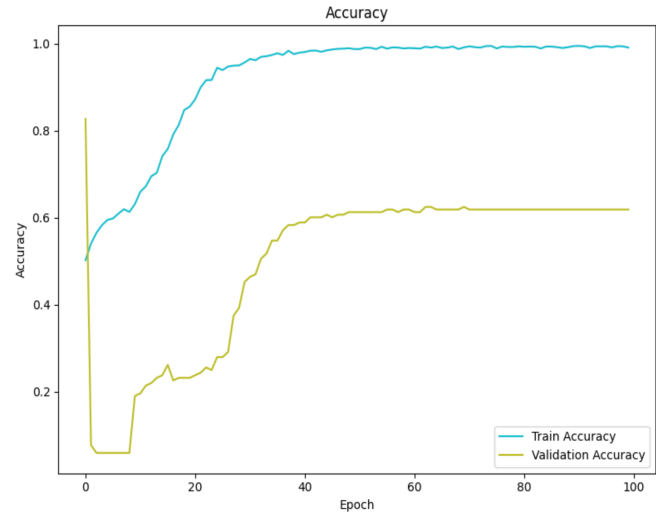
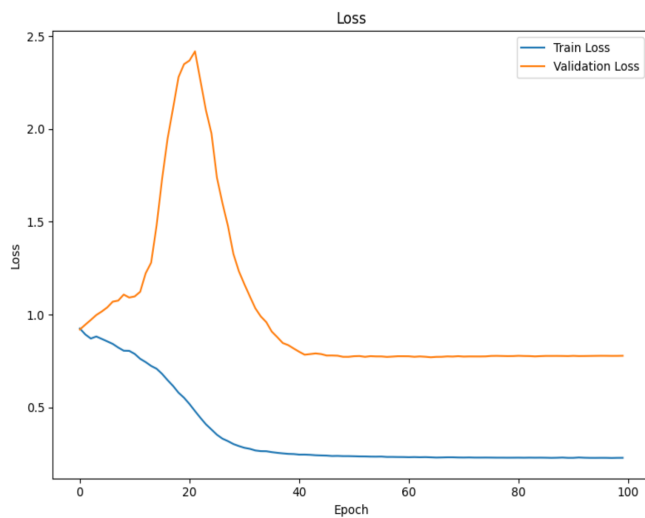


Fig. C.13: Curvas de pérdida y exactitud binaria del modelo binario de núcleos de células de Kupffer con ResNet50.
 a) Curva de pérdida del entrenamiento de ResNet50, b) Curva de exactitud binaria del entrenamiento de ResNet50.

a)



b)

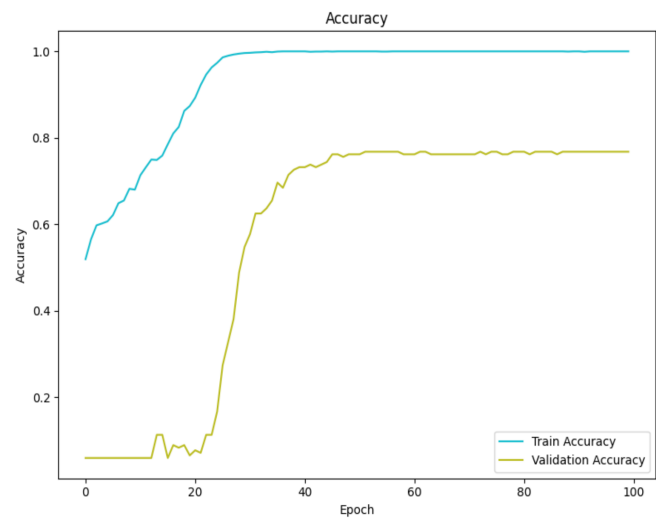
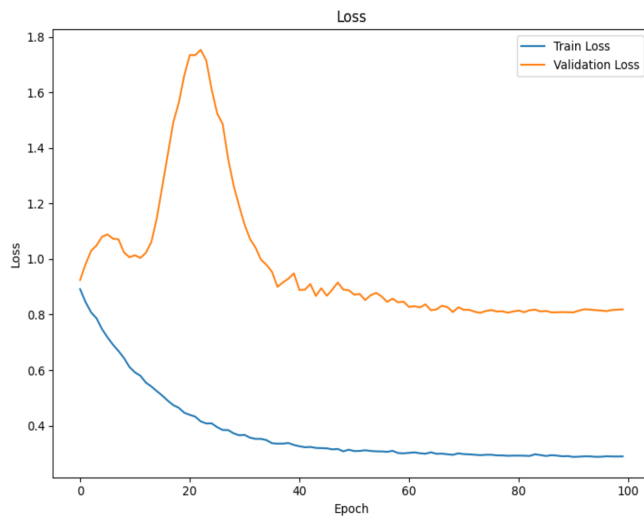


Fig. C.14: Curvas de pérdida y exactitud binaria del modelo binario de núcleos de células de Kupffer con InceptionV3.
 a) Curva de pérdida del entrenamiento de InceptionV3, b) Curva de exactitud binaria del entrenamiento de InceptionV3.

a)



b)

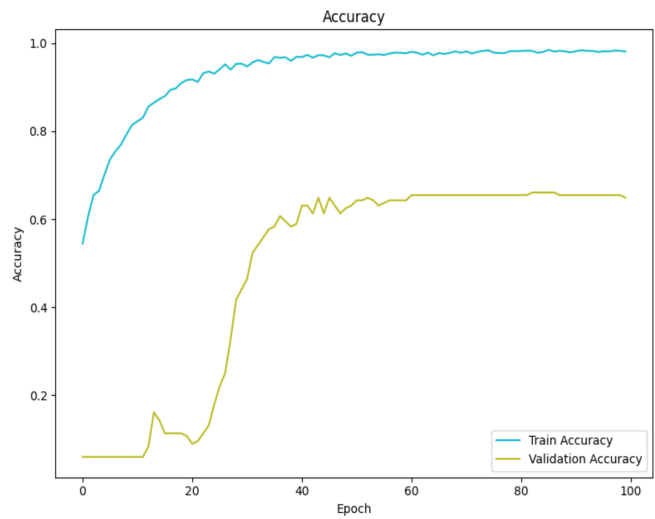


Fig. C.15: Curvas de pérdida y exactitud binaria del modelo binario de núcleos de células de Kupffer con Inception-ResNet-v2.

a) Curva de pérdida del entrenamiento de Inception-ResNet-v2, b) Curva de exactitud binaria del entrenamiento de Inception-ResNet-v2.

A continuación se aprecian las matrices de confusión para los tres modelos binarios con el conjunto de prueba se pueden apreciar en las Fig. C.16, C.17 y C.18.

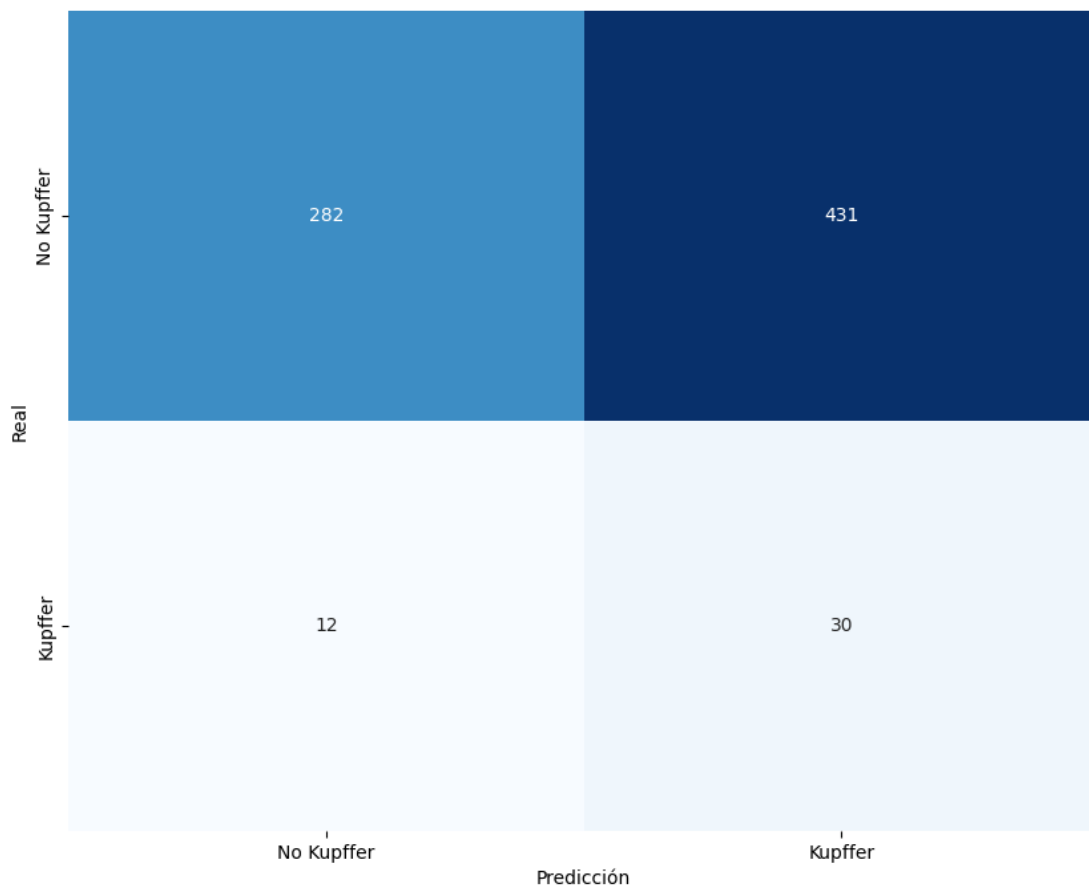


Fig. C.16: Matriz de confusión en el conjunto de prueba con el modelo binario de núcleos de células de Kupffer con ResNet50.

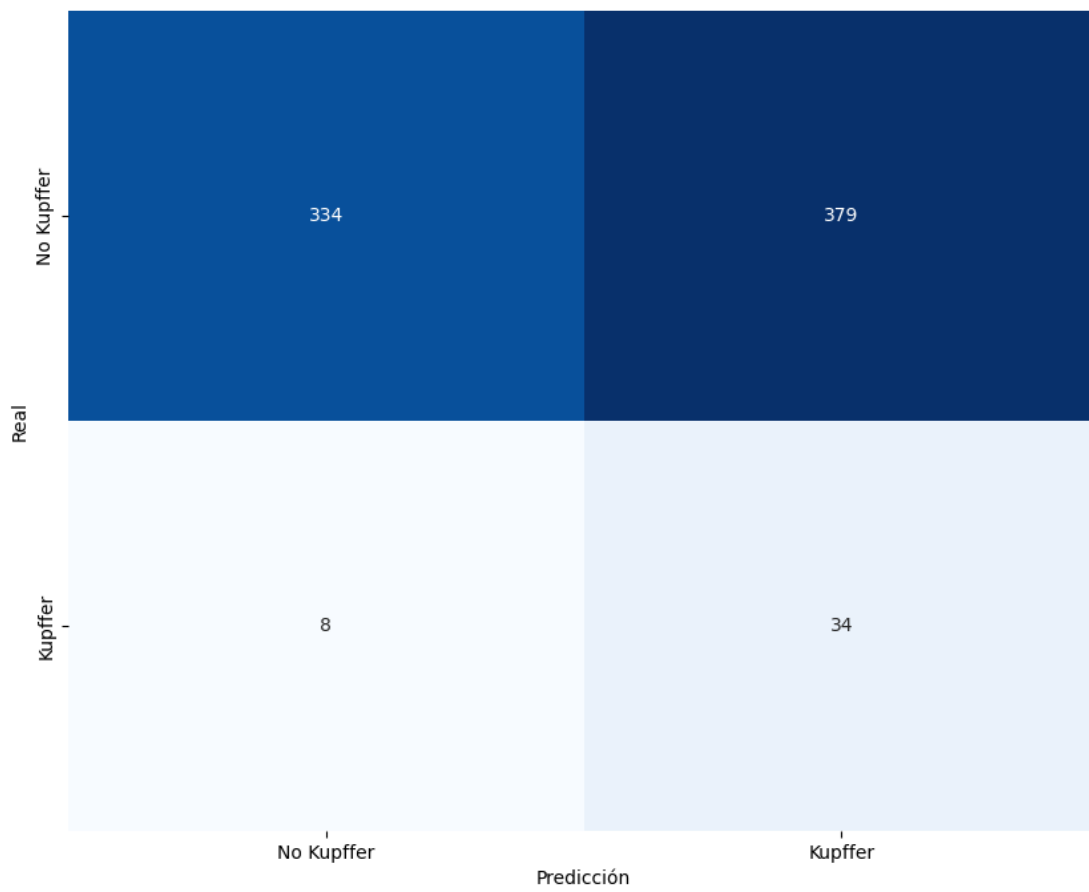


Fig. C.17: Matriz de confusión en el conjunto de prueba con el modelo binario de núcleos de células de Kupffer con InceptionV3.

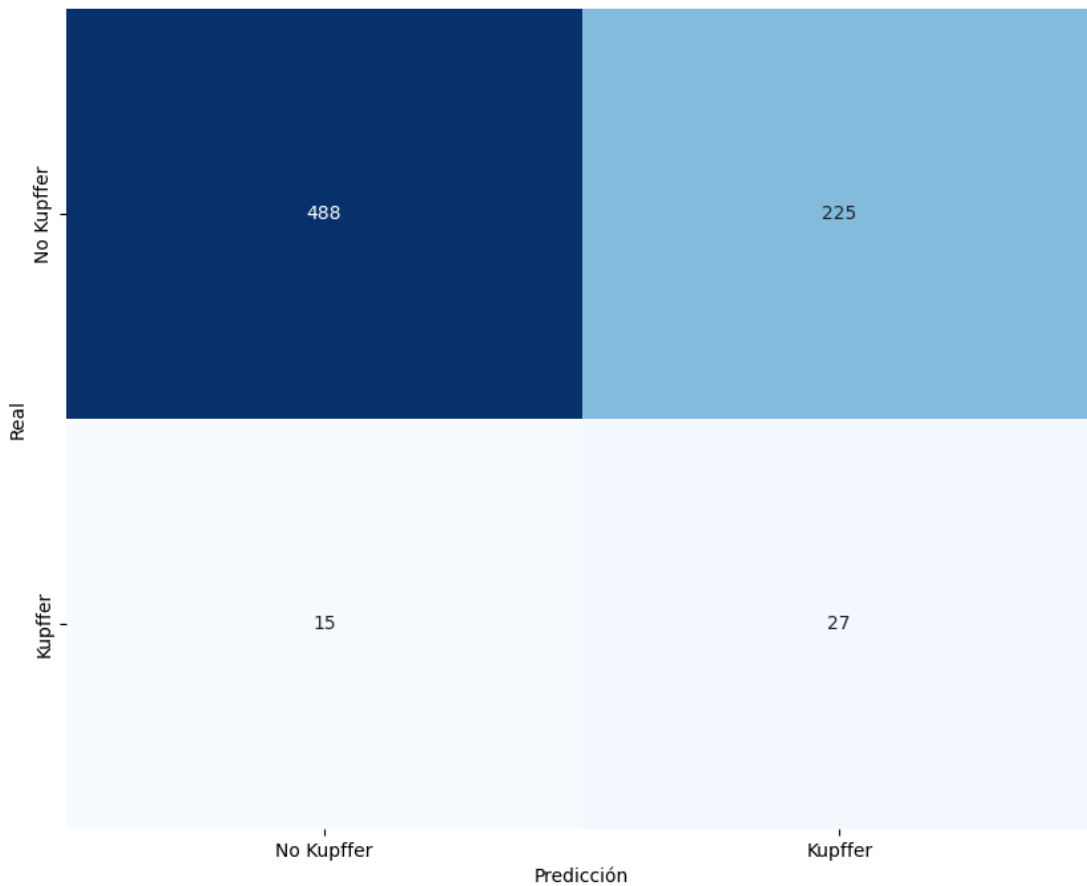


Fig. C.18: Matriz de confusión en el conjunto de prueba con el modelo binario de núcleos de células de Kupffer con Inception-ResNet-v2.

C.4 Modelo Binario de Núcleos de Células Endoteliales

A continuación se presentan las curvas de pérdida y exactitud binaria para los modelos binarios de núcleos de células endoteliales para ResNet50, InceptionV3 e Inception-ResNet-v2 en las Fig. C.19, C.20 y C.21, respectivamente.

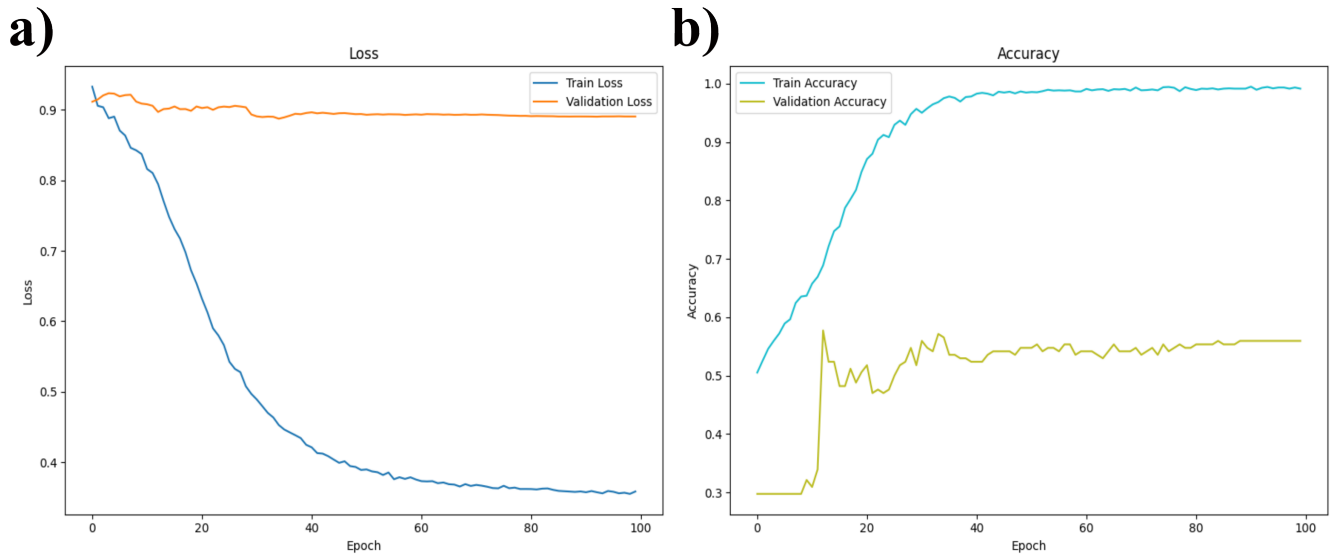


Fig. C.19: Curvas de pérdida y exactitud binaria del modelo binario de endoteliales con ResNet50.

a) Curva de pérdida del entrenamiento de núcleos de células ResNet50, b) Curva de exactitud binaria del entrenamiento de ResNet50.

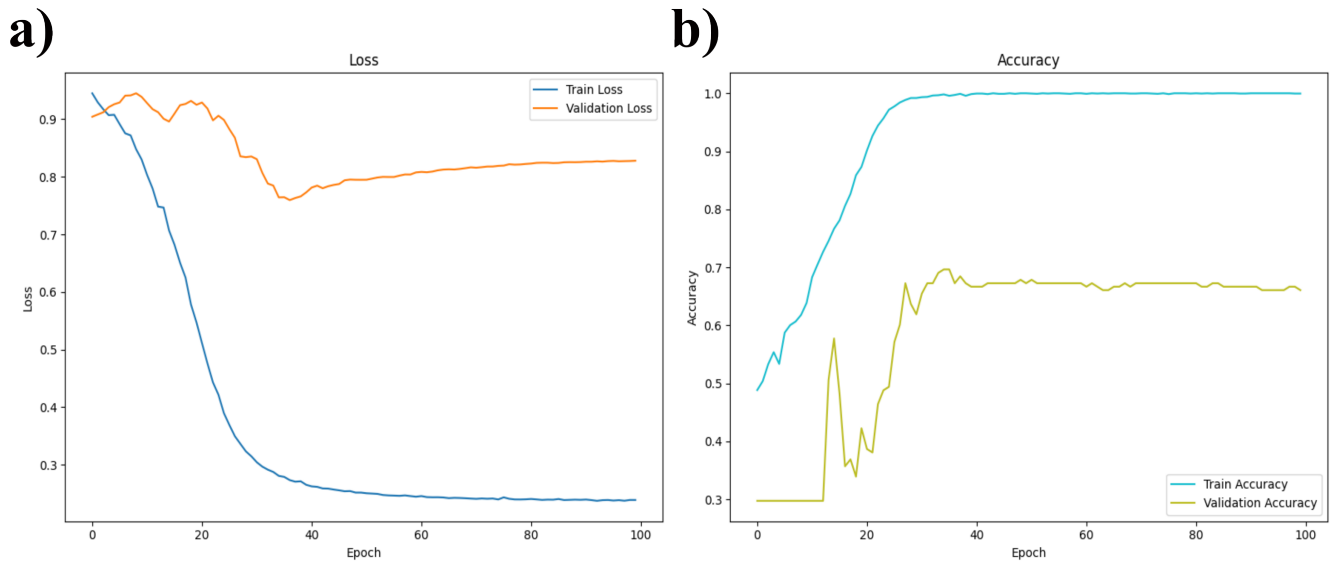
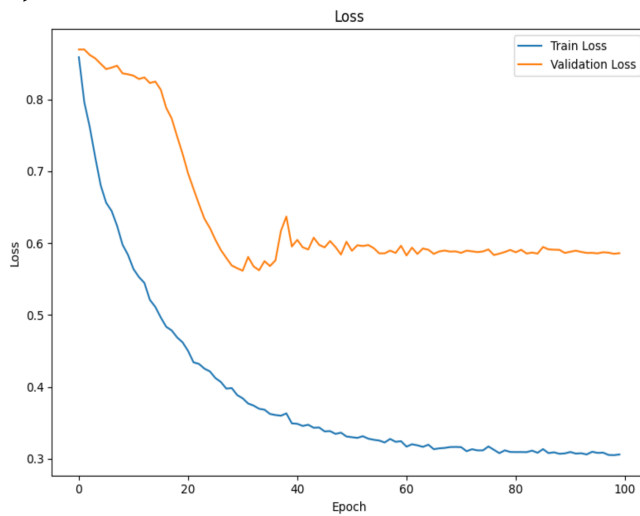


Fig. C.20: Curvas de pérdida y exactitud binaria el modelo binario de núcleos de células endoteliales con InceptionV3.

a) Curva de pérdida del entrenamiento de InceptionV3, b) Curva de exactitud binaria del entrenamiento de InceptionV3.

a)



b)

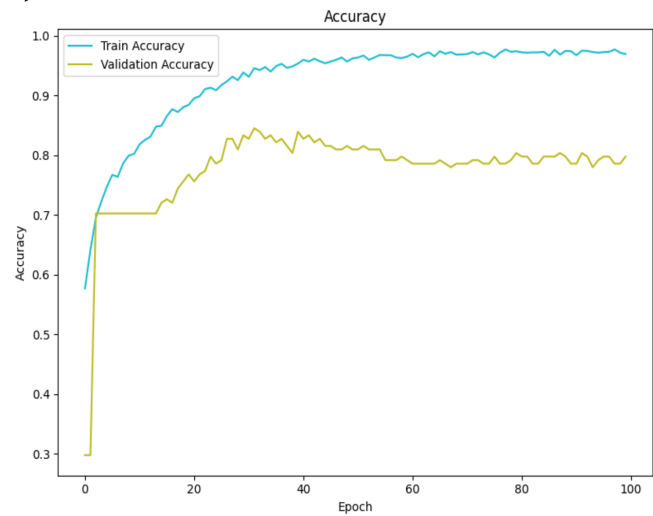


Fig. C.21: Curvas de pérdida y exactitud binaria el modelo binario de núcleos de células endoteliales con Inception-ResNet-v2.

a) Curva de pérdida del entrenamiento de Inception-ResNet-v2, b) Curva de exactitud binaria del entrenamiento de Inception-ResNet-v2.

A continuación se aprecian las matrices de confusión para los tres modelos binarios con el conjunto de prueba se pueden apreciar en las Fig. C.22, C.23 y C.24.

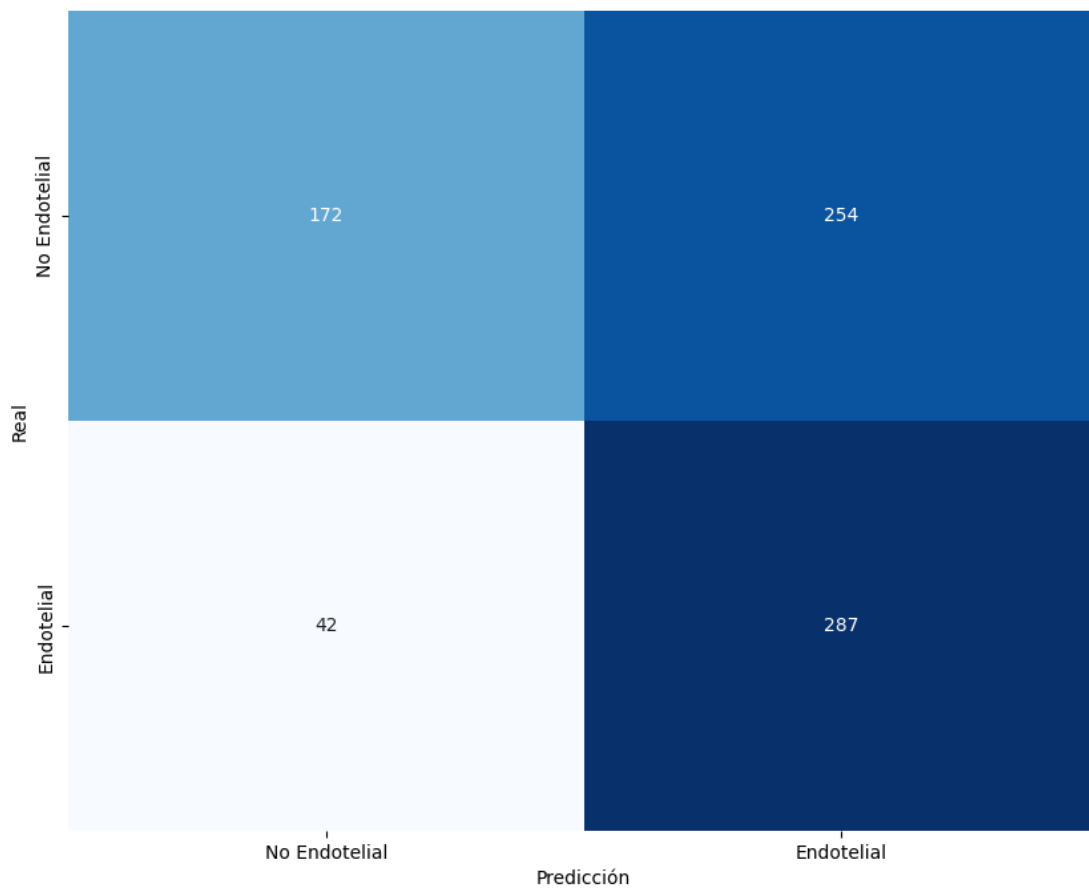


Fig. C.22: Matriz de confusión en el conjunto de prueba con el modelo binario de núcleos de células endoteliales con ResNet50.

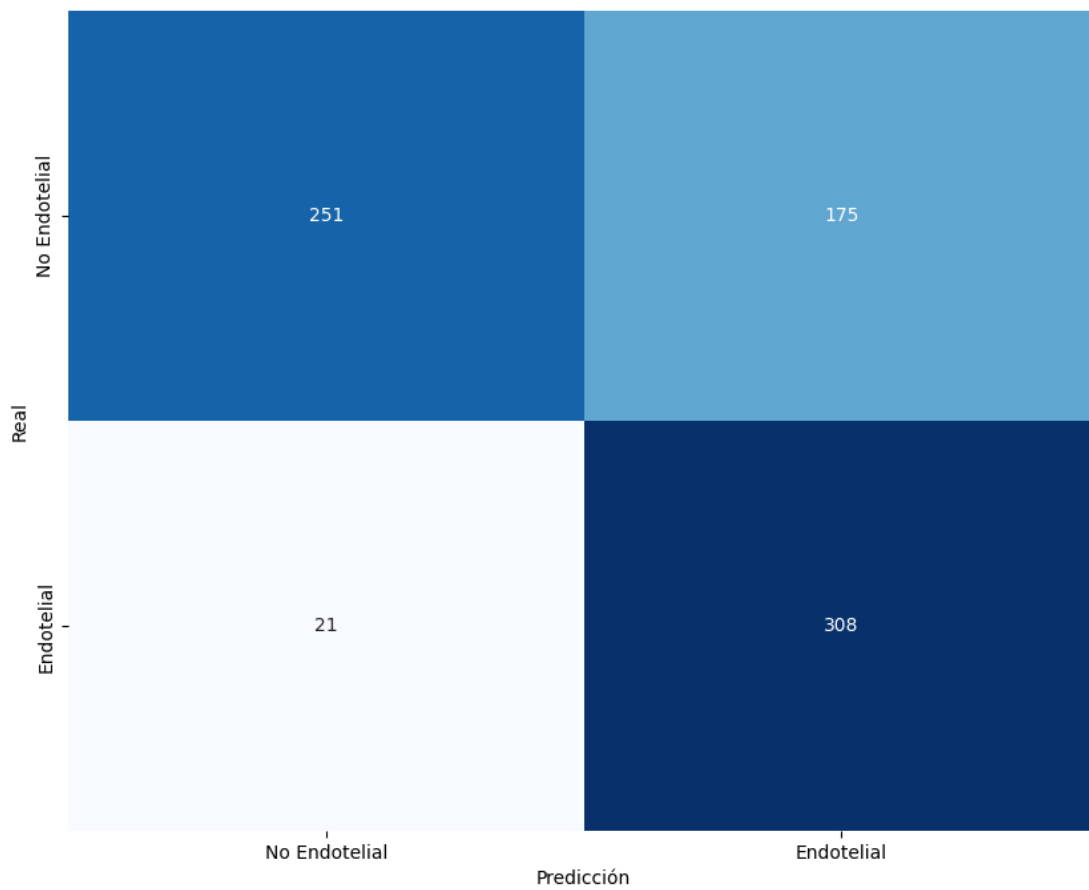


Fig. C.23: Matriz de confusión en el conjunto de prueba con el modelo binario de núcleos de células endoteliales con InceptionV3.

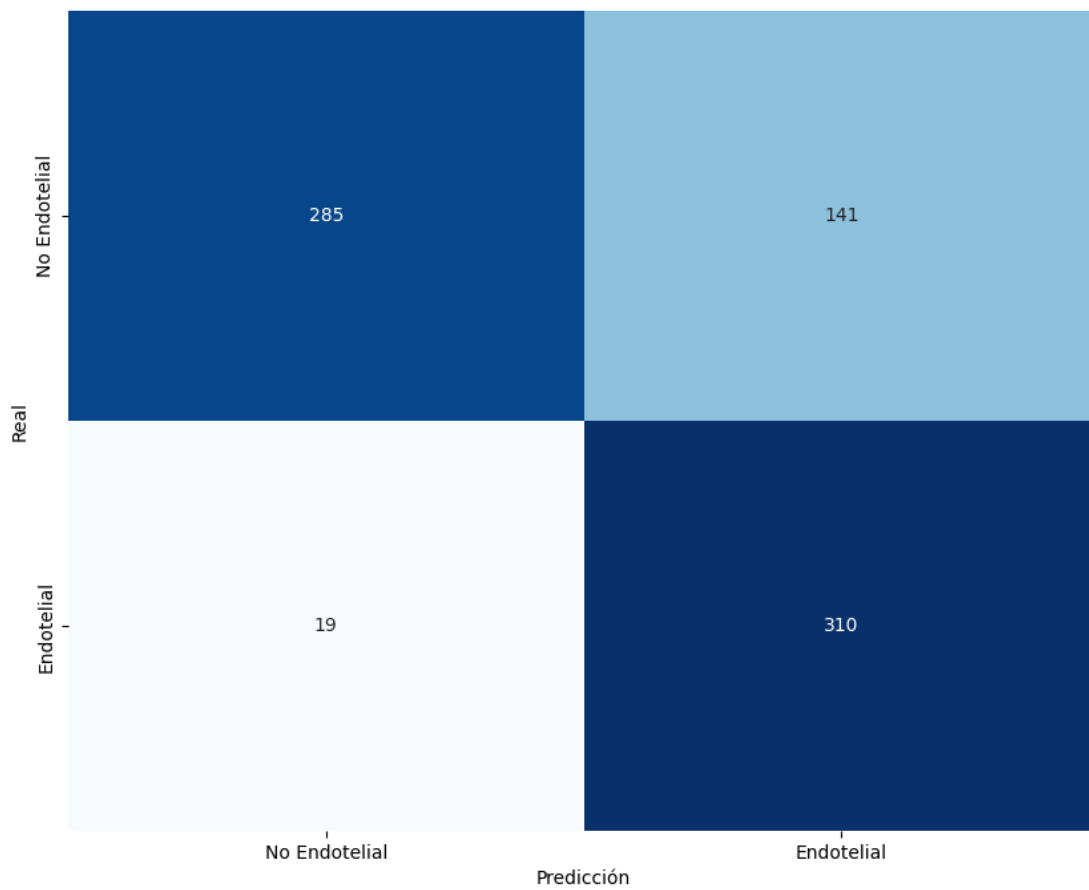


Fig. C.24: Matriz de confusión en el conjunto de prueba con el modelo binario de núcleos de células endoteliales con Inception-ResNet-v2.

Anexo D - Resultados Prototipo Final y Comparación MT

D.1 Prototipo Final

A continuación en la Fig. D.1 se aprecia la matriz de confusión obtenida para el conjunto de prueba con el prototipo final, y en la Fig. D.2 se aprecia la matriz de confusión obtenida para este mismo conjunto de prueba para la combinación de los modelos binarios de este prototipo final.

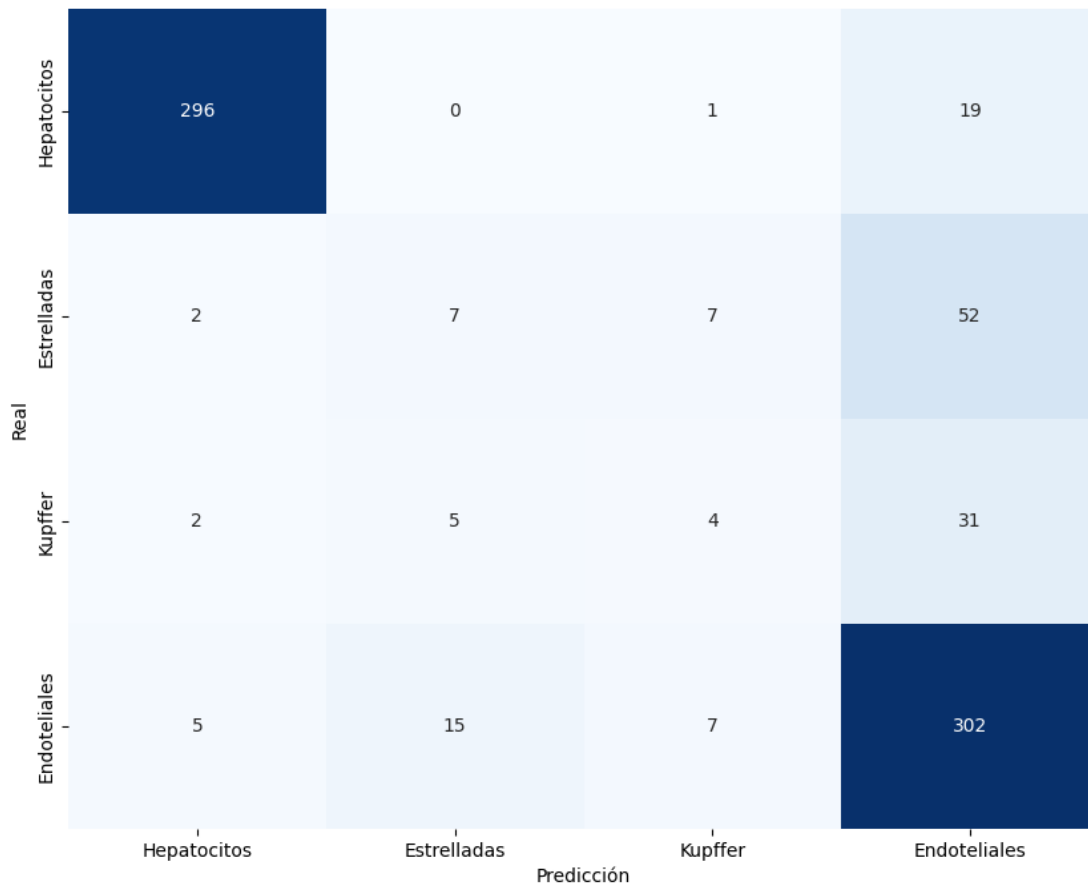


Fig. D.1: Matriz de confusión de la predicción optimizada del modelo final de 4 clases en el conjunto de prueba.

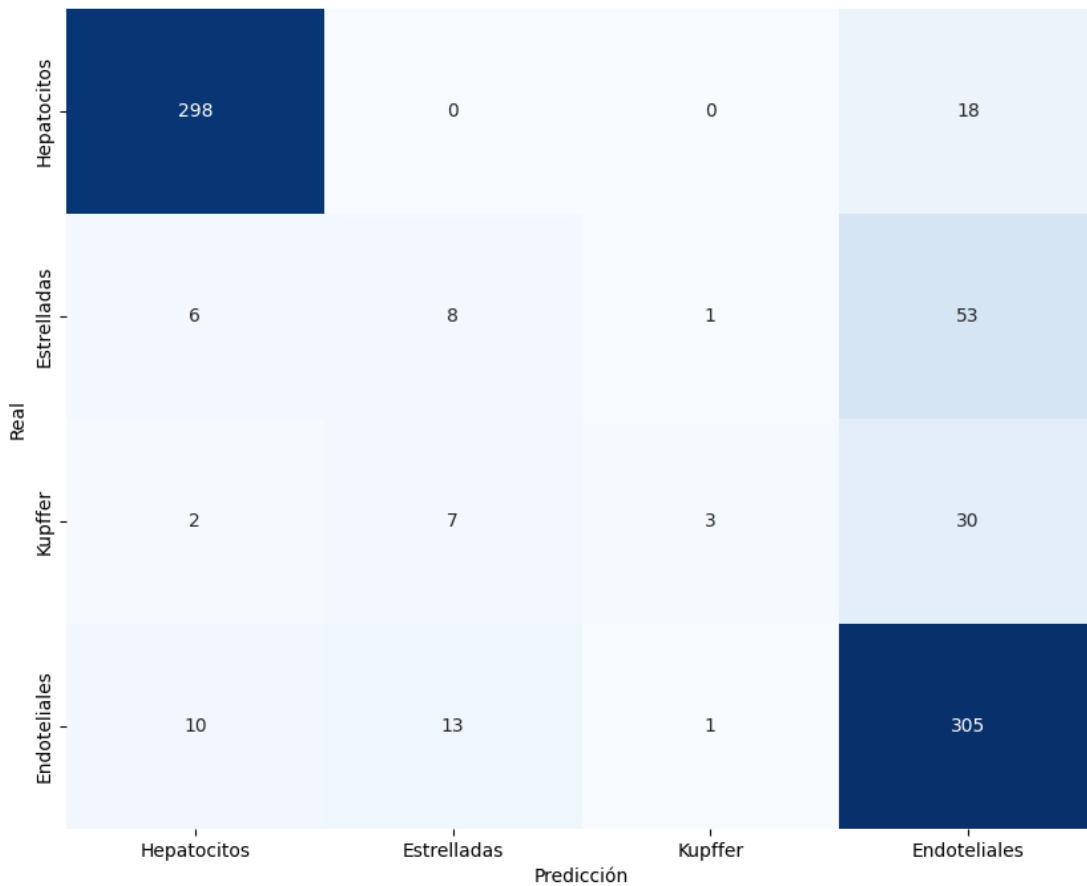


Fig. D.2: Matriz de confusión de la predicción de los modelos binarios finales combinados en el conjunto de prueba.

D.2 Comparación MT

A continuación en la Fig. D.3 se aprecia una matriz de confusión obtenida con el GT de la muestra nueva y la predicción de MT. En la Fig. D.4 se aprecia la matriz de confusión obtenida con el modelo de 4 clases final (Inception-ResNet-v2), en la Fig. D.5 se aprecia la matriz de confusión obtenida con el modelo optimizado, y en la Fig. D.6 la matriz de confusión obtenida con la combinación de los modelos binarios.

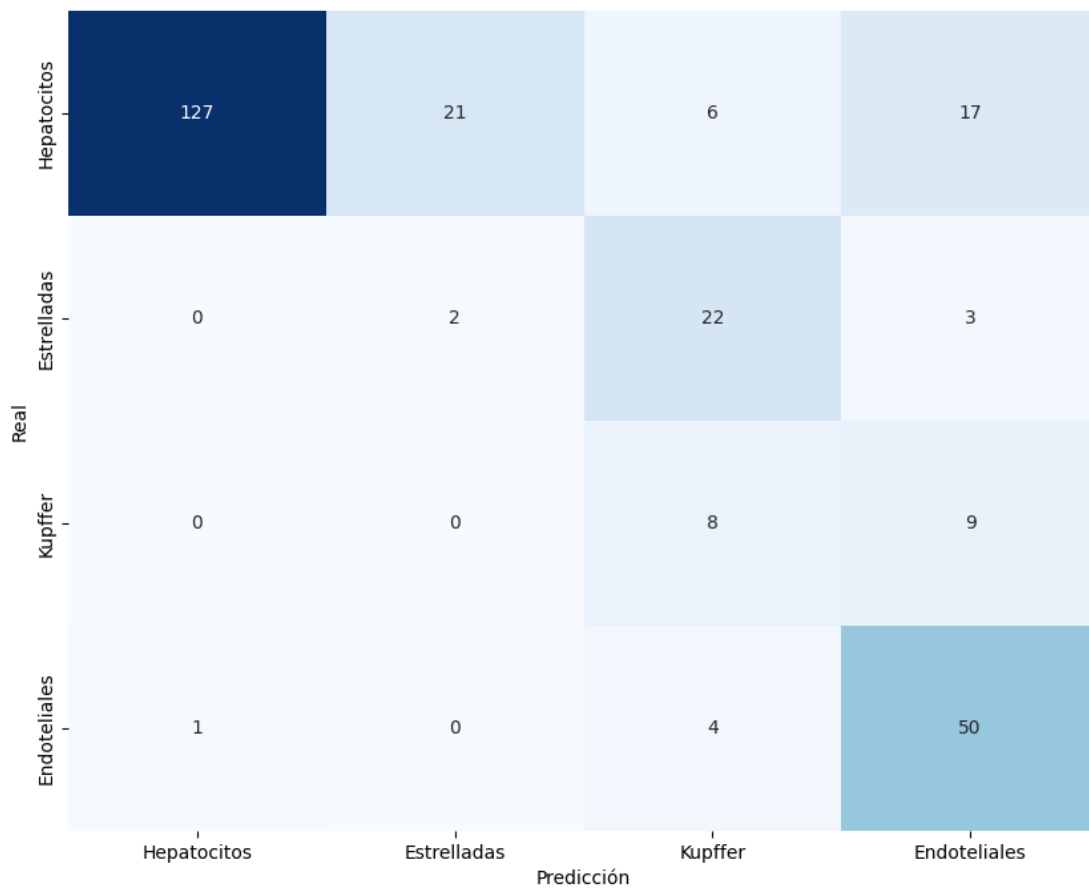


Fig. D.3: Matriz de confusión de la clasificación de MotionTracking en la muestra nueva.



Fig. D.4: Matriz de confusión de la predicción modelo final de 4 clases en la muestra nueva.

Real	Hepatocitos	159	3	0	9
	Estrelladas	2	13	3	9
	Kupffer	0	3	7	7
	Endoteliales	5	17	6	27
		Hepatocitos	Estrelladas	Kupffer	Endoteliales
		Predicción			

Fig. D.5: Matriz de confusión de la predicción optimizada del modelo final de 4 clases en la muestra nueva.

Real	Hepatocitos	159	2	0	10
	Estrelladas	2	19	1	5
	Kupffer	1	8	2	6
	Endoteliales	1	26	5	23
		Hepatocitos	Estrelladas	Kupffer	Endoteliales
Predicción					

Fig. D.6: Matriz de confusión de la predicción de los modelos binarios finales combinados en la muestra nueva.

UNIVERSIDAD DE CONCEPCION – FACULTAD DE INGENIERIA

RESUMEN DE MEMORIA DE TITULO

Departamento	: Departamento de Ingeniería Eléctrica
Carrera	: Ingeniería Civil Biomédica
Nombre del memorista	: Jean Paul Taylor Muñoz
Título de la memoria	: Clasificación de núcleos celulares en 3D basada en Deep Learning para la reconstrucción digital de tejido hepático
Fecha de la presentación oral	: 29 de Agosto 2024
Profesor(es) Guía	: Dra. Pamela Guevara A., Dr. Fabián Segovia M., Dr. Hernán Morales N.
Profesor(es) Revisor(es)	: Dra. Rosa Figueroa I.
Concepto	:
Calificación	:

Resumen

El hígado, esencial en la digestión y el sistema circulatorio, alberga diversos tipos celulares con variaciones nucleares y morfológicas. Para tratar enfermedades hepáticas, es crucial diferenciar estos tipos celulares, lo que requiere herramientas avanzadas de análisis. En este contexto, el Deep Learning (DL) y las Redes Neuronales Convolucionales (CNN) ofrecen una solución prometedora. Estos enfoques pueden superar las limitaciones de los modelos 3D actuales, que dependen de múltiples tipos de imágenes y presentan desafíos en la clasificación precisa debido a las limitaciones de los algoritmos existentes. Este trabajo se centra en la clasificación de núcleos hepáticos en 3D mediante DL. Se creó un Ground Truth (GT) 3D a partir de imágenes de microscopía confocal y se entrenaron CNNs adaptadas a 3D (ResNet50, InceptionV3, Inception-ResNet-v2). Se evaluaron diversos modelos de clasificación, seleccionando el más eficaz mediante métricas de rendimiento. El resultado fue un algoritmo robusto que mejoró significativamente la clasificación 3D de núcleos hepáticos, superando la eficacia del software MotionTracking, destacando un modelo específico para hepatocitos con una clasificación correcta del 95 %. Este avance subraya la importancia de mejorar la reconstrucción en 3D del tejido hepático, optimizando recursos y reduciendo la dependencia de marcadores específicos para estudios detallados del hígado.