



**UNIVERSIDAD
DE GRANADA**

TRABAJO FIN DE GRADO
INGENIERÍA INFORMÁTICA

Análisis Avanzado de Imágenes Médicas con Deep Learning para Cáncer de Páncreas

Autor
Álvaro Ruiz López

Director
Francisco Herrera Triguero



ESCUELA TÉCNICA SUPERIOR DE INGENIERÍAS INFORMÁTICA Y DE
TELECOMUNICACIÓN

Granada, 16 de junio de 2025

Análisis Avanzado de Imágenes Médicas con Deep Learning para Cáncer de Páncreas

Álvaro Ruiz López

Palabras clave: Cáncer de Páncreas, Deep Learning, Inteligencia Artificial, Segmentación de Imágenes Médicas, Asistencia Quirúrgica

Resumen

Este Trabajo Fin de Grado aborda el reto del análisis de imágenes médicas para el cáncer de páncreas, una enfermedad de alta mortalidad y diagnóstico tardío. En colaboración con el Hospital Universitario Clínico San Cecilio, el proyecto busca mejorar la precisión y eficiencia en la segmentación de estructuras pancreáticas en imágenes de tomografía computarizada (TC), superando las limitaciones del proceso manual en tiempo y variabilidad.

El objetivo general es desarrollar métodos avanzados de Deep Learning para segmentar con precisión páncreas, tumores y vasos adyacentes. Se ha empleado una metodología de investigación empírica en Python y PyTorch, con una fase inicial manual y una fase optimizada con el *framework* MONAI para transformaciones y eficiencia.

Se han explorado arquitecturas de Redes Neuronales Convolucionales (CNN) 2D y 3D, acompañadas por un robusto *pipeline* de preprocesamiento que incluye homogeneización a la misma orientación y espaciado, aplicación de ventana y normalización de intensidades (unidades Hounsfield), y aumento de datos dinámico con muestreo por subvolúmenes priorizando el tumor. Se han empleado distintas funciones de pérdida como *Dice Loss*, *Cross-Entropy* o *Focal Loss*, junto con optimizadores como *Adam* y técnicas de regularización como *Dropout* y *Early Stopping*. Los experimentos se han validado con *Hold-Out* estratificado por tipo de tumor.

Finalmente, los resultados obtenidos en el conjunto de *test* muestran coeficientes Dice (DSC) de 0,7932 para el páncreas, 0,8133 para las arterias y 0,7955 para las venas. La segmentación de tumores pancreáticos sigue siendo un desafío, alcanzando un DSC medio de 0,4145 y con un mejor caso de 0,8512.

Advanced Medical Image Analysis with Deep Learning for Pancreatic Cancer

Álvaro Ruiz López

Keywords: Pancreatic Cancer, Deep Learning, Artificial Intelligence, Medical Image Segmentation, Surgical Assistance

Abstract

This Bachelor's Thesis addresses the challenge of medical image analysis for pancreatic cancer, a disease characterized by high mortality and late diagnosis. In collaboration with the Hospital Universitario Clínico San Cecilio, the project aims to enhance the accuracy and efficiency of segmenting pancreatic structures in computed tomography (CT) images, overcoming the limitations of manual processes in terms of time consumption and variability.

The main objective is to develop advanced Deep Learning methods to accurately segment the pancreas, tumors, and adjacent vessels. An empirical research methodology has been applied using Python and PyTorch, beginning with a manual phase and transitioning to an optimized pipeline utilizing the MONAI framework for data transformations and computational efficiency.

The study explores both 2D and 3D Convolutional Neural Network (CNN) architectures, supported by a robust preprocessing pipeline that includes orientation and spacing standardization, windowing and intensity normalization (Hounsfield Units), and dynamic data augmentation via subvolume sampling with a focus on tumor regions. Various loss functions such as Dice Loss, Cross-Entropy, and Focal Loss have been used, alongside optimizers like Adam and regularization techniques including Dropout and Early Stopping. Experiments were validated using stratified Hold-Out based on tumor type.

The final results on the test set show Dice Similarity Coefficients (DSC) of 0.7932 for the pancreas, 0.8133 for arteries, and 0.7955 for veins. Pancreatic tumor segmentation remains a significant challenge, achieving a mean DSC of 0.4145, with the best case reaching 0.8512.

Yo, **Álvaro Ruiz López**, alumno de la titulación Grado en Ingeniería Informática de la **Escuela Técnica Superior de Ingenierías Informática y de Telecomunicación de la Universidad de Granada**, con DNI 78154316V, autorizo la ubicación de la siguiente copia de mi Trabajo Fin de Grado en la biblioteca del centro para que pueda ser consultada por las personas que lo deseen.

Fdo: Álvaro Ruiz López

Granada a 16 de junio de 2025.

D. **Francisco Herrera Triguero**, Profesor del Departamento de Ciencias de la Computación e Inteligencia Artificial de la Universidad de Granada.

Informa:

Que el presente trabajo, titulado *Análisis Avanzado de Imágenes Médicas con Deep Learning para Cáncer de Páncreas*, ha sido realizado bajo su supervisión por **Álvaro Ruiz López**, y autoriza la defensa de dicho trabajo ante el tribunal que corresponda.

Y para que conste, expide y firma el presente informe en Granada a 16 de junio de 2025.

El director:

Francisco Herrera Triguero

Agradecimientos

A mi tutor e investigador de referencia, Francisco Herrera Triguero, por confiar en mí y darme la oportunidad de desarrollar un Trabajo Fin de Grado de estas características.

A Juan Luis Suárez Díaz, por su apoyo a lo largo de todo el desarrollo de este trabajo, su constante disposición y sus valiosos consejos.

Al equipo clínico del Hospital Universitario Clínico San Cecilio de Granada, el Dr. José Luis Martín Rodríguez y José Ramón A. N., por haber posibilitado este proyecto y haberme aportado su conocimiento y ayuda.

Al excelente profesorado de la Escuela Técnica Superior de Ingenierías Informática y de Telecomunicación de la Universidad de Granada que me ha impartido clase durante estos años y que ha contribuido a mi desarrollo académico y personal.

A mis compañeros de clase y amigos Daniel Lozano, Jose Pineda y Carlos Muñoz, por todos esos momentos vividos como grupo de trabajo.

A mis amigos, en especial a Isaac, María y Ainhoa, por haberme apoyado desde el primer momento y haberme sacado una sonrisa cuando lo necesitaba.

A mi familia, por haber sido un pilar fundamental, estando siempre conmigo y motivándome en los peores momentos de esta etapa.

A mis padres y mi hermano, por su amor incondicional, su paciencia y comprensión, y por enseñarme el valor del esfuerzo desde pequeño.

A mis abuelos, presentes y ausentes, porque siempre he sentido su cariño y por haberme acercado, desde muy temprano, a la realidad del cáncer. Su vivencia despertó mi motivación para enfocarme en este camino.

Índice general

1. Introducción	1
1.1. Contexto	1
1.2. Motivación	3
1.3. Objetivos	4
1.4. Estructura de la memoria	5
2. Estado del Arte	7
2.1. Metodología de Revisión	7
2.2. Estado del Arte en Segmentación del Páncreas	8
2.2.1. Arquitecturas de Redes Neuronales	8
2.2.2. Enfoques de Segmentación de Dos Etapas	9
2.2.3. Conjuntos de Datos	9
2.2.4. Resumen	10
2.3. Estado del Arte en Segmentación de Tumores de Páncreas	11
2.3.1. Métodos y Resultados Específicos	11
2.3.2. Resumen	12
2.4. Estado del Arte en Segmentación de Vasos Adyacentes	13
2.4.1. Métodos y Resultados	13
2.4.2. Resumen	13
2.5. Desafíos y Direcciones Futuras	14
2.6. Conclusión del Estado del Arte	15
3. Descripción y planteamiento del problema	17
3.1. Anatomía del páncreas	17
3.2. Tumores pancreáticos	19
3.2.1. Resecabilidad	19
3.2.2. Tipos de tumores	20
3.3. Segmentación semántica	21
3.4. Relevancia y aplicaciones reales	23
3.5. Fundamentos de tomografía computarizada	25
3.6. Descripción del conjunto de datos y aspectos legales	28
3.6.1. Legalidad	30
3.7. Metodología	31

3.8. Planificación y diario de trabajo	33
3.9. Presupuesto	35
3.10. Descripción de las propuestas elaboradas	36
4. Análisis de los datos y preprocesamiento	39
4.1. Descripción detallada y análisis de datos	39
4.1.1. Estadísticas del dataset	40
4.1.2. Análisis exploratorio	42
4.1.3. Visualización	47
4.2. Técnicas de preprocesamiento de datos	56
4.2.1. Preprocesamiento manual	56
4.2.2. Preprocesamiento con MONAI	58
5. Deep Learning	63
5.1. Fundamentos teóricos	63
5.2. Técnicas 2D	71
5.2.1. U-Net 2D	71
5.2.2. DeepLabV3	72
5.3. Técnicas 3D	73
5.3.1. 3D U-Net	73
5.3.2. Attention U-Net	74
5.3.3. DynUNet	75
5.3.4. V-Net	76
5.3.5. SegResNet	77
5.3.6. UNETR	78
5.3.7. Swin UNETR	80
6. Desarrollo del marco experimental de trabajo	83
6.1. Implementaciones	83
6.2. Reproducibilidad	86
6.3. Automatizaciones	87
7. Análisis experimental	89
7.1. Métricas y métodos de validación	89
7.2. Descripción de los experimentos	92
7.3. Resultados	98
7.4. Análisis de resultados	107
8. Conclusiones y trabajos futuros	117
8.1. Conclusiones	117
8.2. Trabajos futuros	118
Bibliografía	128
A. Hoja de resultados de Google Sheets	129

Índice de figuras

3.1.	Anatomía del páncreas. <i>Fuente:</i> [1]	18
3.2.	Irrigación arterial y drenaje venoso del páncreas (arterias en <i>rojo</i> y venas en <i>azul</i>). <i>Fuente:</i> [2]	18
3.3.	Resecabilidad de tumores (masa <i>marrón</i>). <i>Fuente:</i> [3]	20
3.4.	Comparativa de tareas de visión por computador. <i>Fuente:</i> [4]	22
3.5.	Imagen de entrada y segmentación. <i>Fuente:</i> [5]	22
3.6.	Codificación one-hot de la segmentación. <i>Fuente:</i> [5]	23
3.7.	Modelado 3D a partir de segmentaciones. Páncreas en <i>verde</i> , tumor en <i>morado</i> , arterias en <i>rojo</i> y venas en <i>azul</i> . No corresponden al mismo paciente. <i>Fuentes:</i> a) elaboración propia con 3D Slicer [6], b) Proyecto DIGIPanCa (privada)	24
3.8.	Planos anatómicos de imagen de TC. <i>Fuente:</i> [7]	26
3.9.	Sistema de coordenadas anatómicas. <i>Fuente:</i> adaptado de [8]	26
3.10.	Diferencia en la visualización según HU. <i>Elaboración propia</i> .	27
3.11.	Diferencia entre fases. La flecha roja señala la arteria. <i>Elaboración propia</i>	28
3.12.	Diagrama de Gantt del proyecto, correspondiente al periodo Febrero–Junio de 2025	34
3.13.	Ejemplo de tarea en Notion	34
3.14.	Estimación de horas dedicadas al proyecto según registro en Notion	35
4.1.	Visualizaciones de la distribución del espaciado de véxeles. <i>Elaboración propia</i>	42
4.2.	Distribución y dependencia del número de cortes por volumen. <i>Elaboración propia</i>	43
4.3.	Proporción relativa de cada clase segmentada. <i>Elaboración propia</i>	44
4.4.	Distribución por paciente de la proporción de cada clase. <i>Elaboración propia</i>	44
4.5.	Proporción de clases sobre volumen total de TC. “Clases” representa la unión de páncreas, tumor, arterias y venas. <i>Elaboración propia</i>	45

4.6.	Distribución de intensidades por clase, incluyendo el fondo. <i>Elaboración propia</i>	46
4.7.	Distribución de intensidades por clase, excluyendo el fondo. <i>Elaboración propia</i>	47
4.8.	Visualización de TC corte a corte. <i>Elaboración propia</i>	48
4.9.	Visualización de TC y segmentación corte a corte. <i>Elaboración propia</i>	49
4.10.	Visualización multiplanar de corte centrífugo en TC. <i>Elaboración propia</i>	50
4.11.	Visualización multiplanar de corte centrífugo en TC con contornos de máscaras. <i>Elaboración propia</i>	50
4.12.	Intersección de planos en el centro de TC. La figura humana indica la orientación. <i>Elaboración propia</i> usando Slicer [6]	51
4.13.	Intersección de planos junto con segmentación en 3D. La figura humana indica la orientación. <i>Elaboración propia</i> usando Slicer [6]	51
4.14.	Zoom en región de interés. <i>Elaboración propia</i>	52
4.15.	Identificación de estructuras en zona ampliada. <i>Elaboración propia</i>	52
4.16.	Reconstrucción 3D de segmentaciones de paciente con tumor neuroendocrino (TNE). La figura humana indica la orientación. <i>Elaboración propia</i> usando Slicer [6]	53
4.17.	Visualización 3D de tumor resecable y resecable borderline. La figura humana indica la orientación. <i>Elaboración propia</i> usando Slicer [6]	54
4.18.	Visualización 3D sin páncreas de tumor resecable y resecable borderline. La figura humana indica la orientación. <i>Elaboración propia</i> usando Slicer [6]	54
4.19.	Visualización 3D de tumor resecable en distintas perspectivas. La figura humana indica la orientación. <i>Elaboración propia</i> usando Slicer [6]	55
4.20.	Visualización 3D de tumor resecable borderline en distintas perspectivas. La figura humana indica la orientación. <i>Elaboración propia</i> usando Slicer [6]	55
4.21.	Efecto de dilatación binaria con 5 iteraciones. <i>Elaboración propia</i>	59
5.1.	Operación de convolución. El resultado es la suma de la multiplicación elemento a elemento del filtro con la imagen. <i>Fuente: [9]</i>	64
5.2.	Función de activación ReLU con definición matemática. <i>Elaboración propia</i>	64
5.3.	Comparación entre max pooling y average pooling. <i>Fuente: [10]</i>	65

5.4.	Red neuronal convolucional con capa fully connected para clasificar imágenes médicas según enfermedades respiratorias. <i>Adaptado de [11]</i>	65
5.5.	Representación del descenso de gradiente. <i>Fuente: [12]</i>	68
5.6.	Efecto del dropout. A la izquierda una red con dos capas ocultas. A la derecha la red con dropout aplicado (las unidades tachadas son “apagadas”). <i>Adaptado de [13]</i>	69
5.7.	Diferencia entre normalizaciones, con N tamaño de batch, C número de canales o mapas de características, H y W alto y ancho de la imagen respectivamente. <i>Adaptado de [14]</i>	70
5.8.	Punto de early stopping monitorizando error en validación. <i>Fuente: [15]</i>	70
5.9.	Arquitectura de U-Net. Cada recuadro azul corresponde a un mapa de características, con el número de canales y las dimensiones arriba y a la izquierda respectivamente. <i>Fuente: [16]</i>	71
5.10.	Arquitectura de 3D U-Net. Las cajas azules representan los mapas de características, con el número de canales indicado arriba. <i>Fuente: [17]</i>	73
5.11.	Diferencia entre convolución 2D y 3D. <i>Fuente: [18]</i>	74
5.12.	Arquitectura de Attention U-Net. <i>Fuente: [19]</i>	74
5.13.	Arquitectura de la V-Net. <i>Fuente: [20]</i>	76
5.14.	Convoluciones (izquierda) con stride adecuado pueden reducir el tamaño de los datos, deconvoluciones (derecha) lo incrementan proyectando cada voxel a una región más amplia a través del kernel (cubo azul). <i>Fuente: [20]</i>	77
5.15.	Arquitectura de SegResNet. <i>Fuente: [21]</i>	78
5.16.	Arquitectura de UNETR. <i>Fuente: [22]</i>	79
5.17.	Arquitectura de Swin UNETR. <i>Fuente: [23]</i>	80
6.1.	Mensajes del notificador para el experimento “deep_aug”: (a) comienzo del <i>fold</i> 5 del entrenamiento y error y (b) actualización y aviso de entrenamiento completado. <i>Elaboración propia</i>	88
7.1.	UNet 4M. Monitorización del entrenamiento	99
7.2.	UNet 19M. Monitorización del entrenamiento	100
7.3.	UNet 70M. Monitorización del entrenamiento	101
7.4.	DynUNet. Monitorización del entrenamiento	101
7.5.	Attention U-Net. Monitorización del entrenamiento	102
7.6.	SegResNet. Monitorización del entrenamiento	103
7.7.	SegResNet NoLRS. Monitorización del entrenamiento	103
7.8.	V-Net. Monitorización del entrenamiento	104
7.9.	UNETR. Monitorización del entrenamiento	105
7.10.	Swin UNETR 100. Monitorización del entrenamiento	105
7.11.	Swin UNETR 300. Monitorización del entrenamiento	106

7.12. Predicción completa del mejor caso en test. <i>Elaboración propia</i>	111
7.13. Predicción ampliada del mejor caso en test. <i>Elaboración propia</i>	111
7.14. Cortes con contornos de predicción y máscara en el mejor caso de test. <i>Elaboración propia</i>	112
7.15. Reconstrucción 3D del mejor caso de test. La figura humana indica la orientación. <i>Elaboración propia</i> mediante Slicer [6] .	113
7.16. Predicción completa del peor caso en test. <i>Elaboración propia</i>	114
7.17. Predicción ampliada del peor caso en test. <i>Elaboración propia</i>	114
7.18. Cortes con contornos de predicción y máscara en el peor caso de test. <i>Elaboración propia</i>	115
7.19. Reconstrucción 3D del peor caso de test. La figura humana indica la orientación. <i>Elaboración propia</i> mediante Slicer [6] .	116
A.1. Resultados en Google Sheet de experimentos propios a fecha del 9 de mayo de 2025. <i>Elaboración propia</i>	130

Índice de cuadros

2.1. Resumen del estado del arte en segmentación de páncreas	10
2.2. Resumen del estado del arte en segmentación de PDAC	12
2.3. Resumen del estado del arte en segmentación de vasos	14
4.1. Distribución de tipos de tumor en casos de entrenamiento	40
4.2. Distribución de tipos de tumor agrupados (categoría “Otro”) en casos de entrenamiento	40
4.3. Distribución de fases de TC en casos de entrenamiento.	41
4.4. Distribución de fases de TC realmente utilizadas en casos de entrenamiento.	41
4.5. Distribución de dificultad de segmentación manual en casos de entrenamiento.	41
4.6. Estadísticas de spacing de véxoles (mm)	42
4.7. Estadísticas de número de cortes	43
4.8. Proporción de cada clase sobre el volumen total de TC	45
4.9. Diferencia de tamaño entre dataset original y preparado	58
7.1. Distribución de tipos de tumor en split de entrenamiento y validación	91
7.2. Resultados de U-Net 2D	98
7.3. Resultados de DeepLabV3	98
7.4. Resultados de U-Net 3D	99
7.5. Resultados de UNet 4M	100
7.6. Resultados de UNet 19M	100
7.7. Resultados de UNet 70M	101
7.8. Resultados de DynUNet	102
7.9. Resultados de Attention U-Net	102
7.10. Resultados de SegResNet	103
7.11. Resultados de SegResNet NoLRS	104
7.12. Resultados de V-Net	104
7.13. Resultados de UNETR	105
7.14. Resultados de Swin UNETR 100	106
7.15. Resultados de Swin UNETR 300	106

7.16. Comparativa de Dice (resultados destacables resaltados) . . .	107
7.17. Resultados de “SegResNet No LRS” en test	107

Acrónimos

IA	Inteligencia Artificial
DL	Deep Learning
TC	tomografía computarizada
TAC	tomografía axial computarizada
DICOM	Digital Imaging and Communications in Medicine
NIfTI	Neuroimaging Informatics Technology Initiative
NIH	National Institutes of Health
MSD	Medical Segmentation Decathlon
PDAC	adenocarcinoma ductal pancreático (por sus siglas en inglés)
TNE	tumor neuroendocrino
AMS	arteria mesentérica superior
VMS	vena mesentérica superior
CNN	red neuronal convolucional (por sus siglas en inglés)
FCN	red completamente convolucional (por sus siglas en inglés)
MONAI	Medical Open Network for Artificial Intelligence
HU	unidades Hounsfield

Capítulo 1

Introducción

1.1. Contexto

El cáncer de páncreas es una enfermedad mortal con un pronóstico desalentador, presentando una supervivencia global a 5 años de menos del 10 %. Es importante remarcar que se espera que se convierta en la segunda causa principal de muerte por cáncer en Europa en un futuro cercano [24].

El adenocarcinoma ductal pancreático (PDAC) es el tipo más común de cáncer de páncreas. Afecta al páncreas exocrino, que supone la mayor parte del tejido pancreático (aproximadamente 98 %) y contiene varias células encargadas de sintetizar y secretar las enzimas necesarias para el correcto desarrollo del proceso digestivo. Los tumores neuroendocrinos pancreáticos, aunque presentes, son menos frecuentes.

La alta letalidad de esta enfermedad se atribuye principalmente a dos factores:

1. Diagnóstico tardío. Las fases iniciales son asintomáticas, de modo que hasta un 30 %-35 % de los pacientes se diagnostican cuando la enfermedad está localmente avanzada y un 50 %-55 % lo hace ya en fase metastásica [25].
2. Escasez de opciones terapéuticas y tratamientos limitados. En la actualidad, la única opción con potencial curativo es la resección quirúrgica combinada con quimioterapia adyuvante, es decir, primero se extirpa el tumor mediante cirugía y posteriormente se aplica quimioterapia para reducir el riesgo de que vuelva a aparecer. Sin embargo, muchos pacientes no son candidatos a este procedimiento debido a la invasión tumoral de estructuras vasculares críticas tales como la arteria o vena mesentérica superior (AMS y VMS) o a metástasis.

La tomografía computarizada (TC) juega un papel esencial tanto en el diagnóstico precoz como en la planificación quirúrgica, puesto que ofrece imágenes detalladas del abdomen de los pacientes que permiten a los radiólogos identificar los tumores, la etapa en la que se encuentran y realizar segmentaciones.

La segmentación de imágenes médicas es el proceso de asignar a cada píxel (elemento de una imagen 2D) o voxel (elemento de un volumen 3D) una etiqueta que permita distinguir e identificar regiones anatómicas o patológicas de interés. Sin embargo, la segmentación manual de las distintas estructuras como páncreas, tumor y vasos adyacentes es laboriosa y lleva mucho tiempo.

En este contexto clínico crítico, este Trabajo de Fin de Grado (TFG) se enmarca en un proyecto más amplio titulado “Impacto clínico de los modelos digital e impreso en 3D basados en fusión de CT y RM en cirugía oncológica de páncreas”¹, cuyo objetivo principal es determinar el impacto clínico de los modelos digitales e impresos en 3D en pacientes con cáncer de páncreas, tanto en el periodo peroperatorio como intraoperatorio, con el objetivo de mejorar la evaluación clínica y reducir el tiempo de planificación quirúrgica.

El proyecto se realiza en colaboración con el área de radiología del Hospital Universitario Clínico San Cecilio de Granada, quienes proporcionan acceso a las imágenes médicas y supervisión clínica para la validación de los resultados.

Actualmente, la creciente disponibilidad de imágenes médicas en alta resolución y el avance de la Inteligencia Artificial (IA) han propiciado el uso de algoritmos de aprendizaje profundo o *Deep Learning* (DL) para tareas de segmentación automática de órganos y lesiones. En particular, las redes neuronales convolucionales (CNN) se han consolidado como la técnica de referencia para extraer de forma automática las estructuras anatómicas relevantes en estudios de imagen médica, a partir de ejemplos anotados por radiólogos.

Numerosos de ellos se han centrado en órganos sanos, incluyendo el páncreas, y tumores, incluyendo PDAC, y en menor medida en la AMS y VMS, pero no se ha profundizado en este enfoque combinando todas estas estructuras para la creación de un modelo 3D que ayude verdaderamente a cirujanos y radiólogos.

¹Proyecto DIGIPanCa (TED2021-132438B-I00)

1.2. Motivación

La motivación central de este TFG deriva de la necesidad clínica y tecnológica de acelerar y mejorar la precisión de la segmentación de estructuras pancreáticas en imágenes de TC. En el contexto del proyecto hospitalario, la segmentación manual no solo representa un cuello de botella por el tiempo necesario, sino que además la variabilidad interobservador hace que la reproducibilidad sea limitada. Por ello, a nivel clínico:

- Reducción del tiempo de planificación quirúrgica: disponer de segmentaciones automáticas fiables permitiría que el equipo quirúrgico obtenga los modelos 3D antes, sin depender de que un radiólogo lo haga manualmente, lo cual es crítico en tumores desarrollados.
- Mejorar el diagnóstico: la delimitación precisa de la extensión tumoral y su relación con los vasos como AMS, VMS, vena porta y tronco celíaco es esencial para decidir el tratamiento que debe recibir el paciente.
- Facilitar la transición hacia la Industria 4.0, una industria que reúne sistemas físicos, digitales y biológicos en una red de producción inteligente, de colaboración e interacción. Combinando modelos digitales con medicina se facilita el enfoque quirúrgico, aumentando las posibilidades de éxito de la operación al minimizar la improvisación en procedimientos de alta complejidad.

Desde el punto de vista de la investigación en IA y DL, se abordan varios retos fundamentales:

- Variabilidad anatómica y bajo contraste tisular. El páncreas presenta grandes variaciones en tamaño, forma y posición entre pacientes, y el tejido pancreático sano y el tumoral suelen diferir muy poco dentro de la imagen TC, lo que dificulta la delimitación precisa de los contornos.
- Pequeño tamaño del tumor. Los tumores pancreáticos ocupan normalmente menos del 0,1 % del volumen total de la TC, lo que produce un fuerte desequilibrio entre el tumor y el fondo. Esto es importante porque puede provocar que los modelos de DL tiendan a ignorar regiones tan pequeñas.
- Escasez de datos. Aunque existen conjuntos públicos como *Pancreas-CT* de National Institutes of Health (NIH) [26] y *Pancreas* del Medical Segmentation Decathlon (MSD) [27], con 82 y 420 casos respectivamente, la heterogeneidad de protocolos de adquisición (distintos escáneres, fases de contraste utilizadas, etc) y la falta de segmentaciones precisas de tumor y vasos limitan la generalización de los modelos. Contar

con datos reales proporcionados por el hospital aumenta la variedad y permiten probar la robustez en un entorno clínico real.

De este modo, este TFG no solo aporta un avance técnico en la aplicación de Deep Learning a la segmentación médica, sino que también se alinea con una necesidad clínica real, encaminada a mejorar la calidad asistencial y optimizar los recursos hospitalarios en el manejo de una enfermedad tan compleja como letal.

1.3. Objetivos

A partir del contexto establecido en la Sección 1.1 y la motivación expuesta en la sección anterior, en este Trabajo de Fin de Grado (TFG) se propone abordar desafíos críticos en el análisis de imágenes médicas para el diagnóstico y planificación del cáncer de páncreas. El objetivo principal es el siguiente:

Objetivo General (OG): desarrollar y evaluar métodos avanzados de análisis de imágenes médicas basados en Deep Learning que permitan mejorar la precisión y eficiencia en la segmentación de estructuras anatómicas relevantes (páncreas, tumores y vasos adyacentes), en colaboración con especialistas clínicos del Hospital Universitario Clínico San Cecilio.

Este objetivo general se articula en torno a la consecución de los siguientes objetivos específicos:

- **OE1:** realizar un estudio exhaustivo del estado del arte en la segmentación automática de estructuras pancreáticas en imágenes médicas mediante técnicas de aprendizaje profundo, identificando arquitecturas, metodologías y desafíos actuales.
- **OE2:** analizar las características del conjunto de datos proporcionado por el hospital, incluyendo su variabilidad anatómica y el desequilibrio de clases, y aplicar técnicas avanzadas de preprocesamiento para su preparación óptima.
- **OE3:** diseñar e implementar un marco experimental robusto para el entrenamiento y evaluación de modelos, explorando redes neuronales convolucionales 2D y 3D adaptadas a la tarea.
- **OE4:** validar el rendimiento de los modelos con los datos reales utilizando métricas estándar en imagen médica (como el coeficiente Dice), contrastando los resultados con las necesidades clínicas.

Además de los objetivos específicos del proyecto, a nivel personal se persiguen las siguientes metas:

- Adquirir conocimientos fundamentales en un área completamente nueva para un ingeniero informático, como es la medicina, y en particular sobre el cáncer de páncreas.
- Colaborar en un entorno multidisciplinar, enfrentando una problemática real y participando en la búsqueda de soluciones con impacto clínico.

1.4. Estructura de la memoria

Esta sección describe la organización y los contenidos de la presente memoria, que ha sido estructurada para facilitar una comprensión clara y progresiva del trabajo realizado.

En el **Capítulo 1** se ha establecido el contexto del problema abordado, detallado la motivación que impulsa el proyecto y definidos los objetivos generales y específicos que guían la investigación en análisis de imágenes médicas con Deep Learning para el cáncer de páncreas. El resto de la memoria se ha estructurado como sigue:

- En el **Capítulo 2** se presenta una revisión exhaustiva de la literatura existente en el campo de la segmentación de imágenes médicas mediante Deep Learning, centrándose en el páncreas, tumores pancreáticos y vasos adyacentes.
- En el **Capítulo 3** se profundiza en el dominio del problema, se describe el conjunto de datos clínicos utilizado y se abordan aspectos del proyecto como la metodología y las propuestas elaboradas.
- En el **Capítulo 4** se presenta un análisis detallado del conjunto de datos y se describen las técnicas de preprocesamiento aplicadas.
- En el **Capítulo 5** se establecen brevemente los fundamentos teóricos del Deep Learning aplicados a la segmentación de imágenes médicas y se detallan las arquitecturas consideradas.
- En el **Capítulo 6** se describe la implementación del marco experimental, detallando las estrategias de reproducibilidad y alguna automatización de apoyo desarrollada.
- En el **Capítulo 7** se evalúan los modelos propuestos, definiendo métricas y métodos de validación empleados, y se presenta el análisis de los resultados experimentales.
- En el **Capítulo 8** se exponen las conclusiones y posibles trabajos futuros en esta línea de investigación.

Tras el contenido principal se incluye la bibliografía utilizada durante la investigación y desarrollo del proyecto, y los anexos.

Capítulo 2

Estado del Arte

Este capítulo presenta una revisión exhaustiva de las investigaciones previas en el campo de la segmentación automática de imágenes médicas mediante técnicas de Deep Learning, centrándose específicamente en el páncreas, los tumores pancreáticos y las estructuras vasculares adyacentes (arterias y venas). Se analizarán las principales metodologías, arquitecturas de redes neuronales y desafíos abordados en la literatura, con el fin de identificar las debilidades y definir la contribución de este proyecto.

2.1. Metodología de Revisión

La revisión del estado del arte se ha estructurado siguiendo un enfoque sistemático para identificar y analizar las contribuciones más relevantes en la segmentación de páncreas, tumor y vasos en imágenes médicas utilizando técnicas de DL.

Para la búsqueda en bases de datos académicas y repositorios específicos de este ámbito se han utilizado palabras clave como “pancreas segmentation deep learning”, “PDAC segmentation deep learning” y “vessel segmentation deep learning” principalmente, aunque se han empleado otras como “tumor segmentation deep learning” para dar con estudios relativos a otras tareas tales como *Brain Tumor Segmentation* (segmentación de tumores cerebrales) pero que en definitiva tengan la misma finalidad: segmentar tumores en imágenes médicas.

Las bases de datos consultadas, tanto directamente como a través de

Google Scholar¹, han sido: PubMed², Springer³, Elsevier⁴, Nature⁵, BMC Medical Imaging⁶, phiRO (*Physics and Imaging in Radiation Oncology*)⁷, AJR (*American Journal of Roentgenology*)⁸, IEEE Xplore⁹ y Frontiers¹⁰. En la bibliografía se puede observar que algunas citas mencionan publicaciones en arXiv. Esto se debe a que se han consultado en su estado *preprint*, lo que quiere decir aún no han pasado el proceso editorial de alguna revista académica, pero que ha permitido estudiar trabajos más recientes.

Como criterio de selección se han priorizado trabajos recientes y con enfoques basados en redes neuronales convolucionales (CNN) y sus variantes, arquitecturas híbridas y basadas en *transformers*, así como modelos de dos etapas orientados a la optimización de la localización y segmentación de estructuras complejas.

2.2. Estado del Arte en Segmentación del Páncreas

La segmentación automática del páncreas ha sido siempre una tarea difícil para los métodos computacionales debido a su alta variabilidad en forma, y al bajo contraste con los tejidos circundantes en las tomografías computarizadas (TCs) [28]. Métodos de segmentación tradicionales basados en regiones, umbralización o atlas presentaban limitaciones [29]. Sin embargo, el Deep Learning ha demostrado ser muy prometedor para las tareas de segmentación de imágenes médicas [30].

2.2.1. Arquitecturas de Redes Neuronales

- Las redes neuronales convolucionales (CNN) y, en particular las arquitecturas U-Net, han sido ampliamente adoptadas para la segmentación del páncreas [31, 32, 33].
- Attention U-Net ha demostrado mejorar la precisión de la predicción en numerosas tareas, y en particular en la segmentación del páncreas, al permitir que la red se enfoque en regiones relevantes [19, 31].

¹<https://scholar.google.es/>

²<https://pubmed.ncbi.nlm.nih.gov/>

³<https://link.springer.com/>

⁴<https://www.sciencedirect.com/>

⁵<https://www.nature.com/>

⁶<https://bmcmedimaging.biomedcentral.com/>

⁷<https://www.phiro.science/>

⁸<https://ajronline.org/>

⁹<https://ieeexplore.ieee.org/Xplore/home.jsp>

¹⁰<https://www.frontiersin.org/>

- Estudios como [34] compararon diferentes arquitecturas U-Net 3D, incluyendo Basic U-Net, Dense U-Net, Residual U-Net y Residual Dense U-Net, en un conjunto de datos compuesto por 1006 TACs abdominales. Residual Dense U-Net obtuvo los mejores resultados en validación interna, con un Coeficiente Dice (DSC) de 0,842.
- Modelos como MOMUNet han logrado mejoras significativas en la precisión del Coeficiente Dice (DSC) sobre modelos de última generación en el conjunto de datos *Pancreas-CT* del NIH [26] (2,56 % de mejora) y *Pancreas* del MSD [27] (2,97 % de mejora), siendo además una red ultraligera, pues tiene únicamente 1,31 millones de parámetros [33].
- Auto-nnU-Net es un método autoconfigurable que genera configuraciones de U-Net 2D, 3D y 3D en cascada, logrando un rendimiento de vanguardia o *state-of-the-art* [31].

2.2.2. Enfoques de Segmentación de Dos Etapas

- Debido al pequeño tamaño y los límites borrosos del páncreas, los métodos de dos etapas (grueso a fino o *coarse-to-fine*) son comunes para la segmentación del páncreas. Estos enfoques primero localizan una región de interés (etapa gruesa) y luego realizan una segmentación más fina dentro de esa región [31, 32, 33].
- Como ejemplo de este enfoque se encuentra el uso de YOLOv5 para la localización inicial de las regiones del páncreas, seguido de modelos como U-Net, VNet, SegResNet o HighResNet para la segmentación, destacando esta última [29].
- También se ha utilizado la localización del páncreas mediante subvolúmenes 3D LBP (*Local Binary Patterns*) y redes neuronales convolucionales 3D [30].

2.2.3. Conjuntos de Datos

- Los datasets públicos más utilizados para la segmentación del páncreas incluyen *Pancreas-CT* del NIH [26], el cual proporciona escaneos de 82 pacientes sanos junto con las anotaciones del páncreas, y *Pancreas (Task 7)* del MSD, compuesto por 420 volúmenes TC que además de páncreas incluye anotaciones de tumores [27].
- Otros datasets multiorgánicos como AbdomenCT-1k [35], que combina datos de múltiples centros y proveedores e incluye páncreas, hígado, bazo y riñones también son relevantes [31].

- La cantidad de datos es crucial para el rendimiento de los modelos de DL en segmentación médica [34]. Sin embargo, la escasez de grandes datasets es una gran limitación, y el páncreas generalmente ocupa una porción muy pequeña del volumen de la TC (menos del 0,5 % en 3D) [29, 30, 31, 34].

2.2.4. Resumen

A continuación se presenta un cuadro resumiendo las contribuciones y limitaciones de los estudios analizados para segmentación de páncreas:

Cuadro 2.1: Resumen del estado del arte en segmentación de páncreas

Ref.	Contribuciones clave	Limitaciones
[19]	Demostró mejorar la precisión al permitir que la red se enfoque en regiones relevantes. Pioneros en uso de atención probabilística (<i>Soft-Attention</i>) que simplifica el entrenamiento	Reducción de resolución de imagen por limitaciones de memoria y centrado en órganos abdominales sanos
[34]	Pioneros en aplicar DL en un dataset abdominal de gran tamaño. Residual Dense U-Net demostró el rendimiento más alto	Dependencia del conjunto de datos: peores resultados por diferencias en las imágenes. Segmentación semiautomatizada: los volúmenes se recortaron manualmente debido a limitaciones en memoria RAM y GPU
[36]	Método autoconfigurable con rendimiento decente, flexible a recursos. Analiza la importancia de hiperparámetros	Dependiente del conjunto de datos y aleatoriedad
[31]	Ánálisis detallado de arquitecturas DL	No probado en datos adicionales a los datasets públicos
[30]	Empleo de LBP para localización y CNNs 3D para segmentar (enfoque de dos etapas)	Tuvieron que usar arquitecturas 2D para segmentar debido al tamaño limitado del dataset y el alto coste computacional
[29]	Superioridad de YOLOv5 en localización y HighResNet en segmentación (enfoque de dos etapas)	CNNs 2D no pueden preservar información contextual volumétrica y 3D requieren más recursos

Continúa en la siguiente página

Cuadro 2.1 – continuación

Ref.	Contribuciones clave	Limitaciones
[33]	Método de agrupación de muestras para solventar variaciones entre tamaños del páncreas	No aborda el problema de diferenciar tumor de tejido sano: une la máscara del páncreas y la máscara del tumor en una sola estructura para la segmentación, esto es, no se proporciona una segmentación separada entre órgano y lesión

2.3. Estado del Arte en Segmentación de Tumores de Páncreas

La segmentación de tumores pancreáticos es aún más compleja que la del páncreas sano, ya que los tumores son más pequeños (a menudo menos del 0,1% del volumen total de la TC), sus límites son más difusos y el contraste con el tejido circundante es bajo [30, 31]. Los estudios sobre la segmentación de tumores pancreáticos son limitados en comparación con los de órganos grandes. Los resultados reportados muestran una precisión relativamente baja, con coeficientes Dice que oscilan entre 0,52 y 0,71, dependiendo del tipo de tumor y las características del dataset [30].

2.3.1. Métodos y Resultados Específicos

- Deep LOGISMOS combina el aprendizaje contextual profundo con el algoritmo LOGISMOS para la segmentación de tumores 3D, mejorando el coeficiente Dice y la diferencia de volumen relativa (RVD) [29].
- Transformer Guided Progressive Fusion Network (TGPFN) ha demostrado ser efectivo para la segmentación de masas tumorales pancreáticas, logrando un Dice del 0,74 y una precisión de detección del 0,92 en un conjunto de datos privado de TC [29].
- Se han propuesto métodos de segmentación multiescala grueso a fino para la detección de adenocarcinoma ductal pancreático (PDAC), reportando un coeficiente Dice para la masa tumoral del 0,57. También se propone un enfoque basado en CNN con *deep supervision* y *attention gates* para tumores de hígado y páncreas, alcanzando un Dice del 0,55 para la segmentación de tumores pancreáticos [30].
- La red guiada por temperatura se ha utilizado para la segmentación conjunta 3D del páncreas y los tumores [37].

12 2.3. Estado del Arte en Segmentación de Tumores de Páncreas

- Dentro del MONAI Model Zoo se encuentra el modelo preentrenado *Pancreas CT DiNTS segmentation* [38], el cual haciendo uso de DiNTS busca la topología óptima para la red a emplear en el problema. Con este modelo reportan un Dice medio global de 0,62, es decir, en páncreas y tumor (no se aporta por clase). Este modelo también ha sido entrenado usando el dataset de *Pancreas* del MSD [27, 39].
- El reciente trabajo [40] emplea amplificación central y otras técnicas para aumentar la información de las características de la lesión y balancear el número de muestras. Implementan el mecanismo de atención ECSA, que es efectivo para reconocer objetivos pequeños sin elevar la complejidad computacional. Consiguieron mejorar el Dice medio en un 7,43 % respecto de U-Net base.

2.3.2. Resumen

La siguiente tabla resume las contribuciones y limitaciones vistas para este caso:

Cuadro 2.2: Resumen del estado del arte en segmentación de PDAC

Ref.	Contribuciones clave	Limitaciones
[29]	Diversas técnicas de aumento de datos para solventar problema de desequilibrio y variabilidad de datos e integración de convolución 2.5D que mejora el rendimiento	Dependencia del conjunto de datos privado y costo computacional de la convolución 2.5D
[30]	Empleo de LBP para localización y CNNs 3D para segmentar (enfoque de dos etapas)	Tamaño limitado del dataset y alto coste computacional
[37]	Red guiada por temperatura para segmentación conjunta 3D de páncreas y tumores	Dependiente del conjunto de datos y aleatoriedad. Segmentación deficiente cuando el tumor se encuentra en el área del borde del páncreas
[38]	Búsqueda de topología óptima para la red	Elevado consumo de memoria RAM/GPU y rendimiento no reportado por clase
Continúa en la siguiente página		

Cuadro 2.2 – continuación

Ref.	Contribuciones clave	Limitaciones
[40]	Amplificación central y otras técnicas para aumentar información de lesiones y complejidad computacional baja	Bloques de amplificación pueden tener un rendimiento deficiente en segmentación de vasos y riesgo de desequilibrar aún más el tamaño de la muestra

2.4. Estado del Arte en Segmentación de Vasos Adyacentes

La segmentación precisa de las estructuras vasculares adyacentes al páncreas, tales como la arteria y la vena mesentérica superior, el tronco celiaco y la vena porta, es crucial para establecer relaciones anatómicas entre el tumor y estas estructuras, lo cual influye directamente en las decisiones clínicas. La planificación quirúrgica requiere imágenes de alta resolución de TC para evaluar esta relación, por lo que los modelos de segmentación de vasos específicos de pacientes son esenciales para una visualización 3D preoperatoria [41, 42].

2.4.1. Métodos y Resultados

- Los estudios sobre la segmentación de vasos abdominales son limitados, no reciben la misma atención que otros órganos.
- En [43] se utilizó una red completamente convolucional (FCN) para la segmentación de la arteria abdominal empleando un método basado en parches y consiguiendo de media un Dice de 87,1%.
- Más en línea con este proyecto, la propuesta en [30] se centró en la segmentación del PDAC y los vasos circundantes: AMS y VMS. Debido al pequeño tamaño de su muestra, tan solo 19 casos, utilizaron redes preentrenadas (Attention U-Net, TAU-Net y un modelo híbrido) que fueron ajustadas por medio de *fine-tuning* para esta tarea. En su caso, el modelo híbrido mostró el mejor rendimiento en la segmentación de vasos.

2.4.2. Resumen

A continuación se presenta un resumen de las contribuciones y limitaciones de los estudios analizados respecto a la segmentación de vasos:

Cuadro 2.3: Resumen del estado del arte en segmentación de vasos

Ref.	Contribuciones clave	Limitaciones
[43]	Centrado en segmentación completamente automatizada de la arteria abdominal con alto rendimiento. Manejo efectivo del desequilibrio de volumen (la arteria ocupa poco espacio de la TC) mediante generación de parches para entrenar	Enfoque exclusivo en arterias abdominales y clasificación errónea de venas como arterias (reconocido por los autores)
[30]	Además del empleo de LBP para localización y CNNs 3D para segmentar, se centró en segmentación del PDAC y vasos circundantes (AMS y VMS)	En cuanto a vasos se refiere, se basó en una muestra muy pequeña (solo 19 casos), requirió redes preentrenadas

2.5. Desafíos y Direcciones Futuras

Los estudios analizados coinciden en que, a pesar de los avances recientes en segmentación automática del páncreas y tumor mediante técnicas de DL, el problema aún no está resuelto. Entre los principales desafíos recurrentes en la literatura destacan:

- Escasez y heterogeneidad de los datos: la falta de grandes datasets anotados y disponibles de manera pública es una de las mayores limitaciones. Los datos existentes normalmente varían en calidad, protocolos de adquisición (por ejemplo diferentes escáneres TC, fases de contraste...) y tipos de anotaciones, lo que dificulta la generalización de los modelos. Preparar nuevos datasets anotados es costoso y además presenta desafíos éticos y de privacidad. El estudio [34] reconoce que modelos evaluados con datasets pequeños pueden no ser fiables.
- Desequilibrio de clases y tamaño de los objetivos: si bien el reducido tamaño de los tumores es una dificultad inherente del problema, puede llevar a que los modelos de DL tiendan a ignorar o segmentar deficientemente regiones pequeñas, limitando el rendimiento de los algoritmos existentes.
- Bajo contraste y límites difusos: siendo nuevamente un factor intrínseco del problema, la similitud en las intensidades de vértices entre

el páncreas y los órganos circundantes, y especialmente entre el tejido pancreático sano y el tumoral, resulta en límites poco claros que complican y dificultan una segmentación precisa.

Sin embargo, la mayor laguna encontrada es la ausencia de estudios que se enfoquen en el problema de combinar la segmentación de páncreas, tumor, arterias y venas simultáneamente para proporcionar modelos 3D útiles en medicina, tal y como se anticipaba en la Sección 1.1.

La literatura identifica como posibles líneas de trabajo futuras:

- Validación clínica en escenarios reales: es fundamental que los modelos de Deep Learning no solo demuestren un alto rendimiento en conjuntos de datos de referencia (públicos), sino que también sean clínicamente aplicables y robustos en entornos del mundo real con datos heterogéneos. Las mejoras obtenidas deben tener un impacto real en los pacientes [44].
- Generación de datos sintéticos: una dirección prometedora es la generación de datos sintéticos para aumentar la cantidad y diversidad de los datasets de entrenamiento sin comprometer ni privacidad ni estándares éticos [45].
- Modelos de Aprendizaje Federado: incorporar técnicas de aprendizaje federado para aprovechar la información de múltiples instituciones o centros sin necesidad de compartir los datos brutos, lo que podría reducir las preocupaciones de privacidad y la escasez de datos [31, 46].

2.6. Conclusión del Estado del Arte

La segmentación automática del páncreas, los tumores y los vasos adyacentes es una tarea de vital importancia para el diagnóstico temprano y la planificación quirúrgica del cáncer de páncreas, como se ha mencionado previamente. Aunque las técnicas de DL, especialmente las arquitecturas basadas en U-Net y los enfoques de dos etapas, han logrado avances significativos en la segmentación del páncreas, en el caso de tumores y, en menor medida de vasos, sigue siendo un desafío debido a su pequeño tamaño, límites borrosos y la escasez de datos anotados de alta calidad.

Es por ello que, mediante este proyecto y en colaboración con el Hospital Universitario Clínico San Cecilio de Granada, se busca abordar estas limitaciones desarrollando un sistema de IA robusto. Explorando técnicas avanzadas de preprocesamiento, redes neuronales y experimentación con funciones de pérdida y arquitecturas, se pretende mejorar la precisión en la identificación de tumores y la delimitación de estructuras vasculares. La contribución

de este proyecto reside en la aplicación y adaptación de estas técnicas a un contexto clínico real, utilizando imágenes proporcionadas por el hospital y buscando una segmentación automática con una precisión y rapidez que la acerquen a la práctica clínica, a la vez que se mejora la generalización del modelo. Esto es crucial para facilitar la visualización 3D en la planificación quirúrgica, un objetivo clave del proyecto original del hospital, con la finalidad de reducir el tiempo de segmentación manual y permitir un diagnóstico más preciso y una intervención más eficaz.

Capítulo 3

Descripción y planteamiento del problema

3.1. Anatomía del páncreas

El páncreas es un órgano glandular profundo y vital en el sistema digestivo, ubicado en el abdomen. Su anatomía presenta una gran variabilidad en tamaño, forma y posición entre los diferentes pacientes. Típicamente, tiene una forma de “J” y se subdivide principalmente en:

- Cabeza: se aloja en la curva del duodeno.
- Cuello: estrechamiento posterior al tronco celíaco y anterior a la vena porta, que une el cuerpo a la cabeza.
- Cuerpo: se extiende hacia la izquierda, posterior al estómago.
- Cola: es la prolongación del cuerpo y se aproxima al bazo.

La Figura 3.1 muestra su forma característica y su localización anatómica relativa.

El tamaño promedio en adultos es de 15–22 cm de longitud, 50–125 g de peso y con un volumen aproximado de 70–80 cm³ [47]. Está compuesto por dos tipos de glándulas:

- Exocrinas (98 % del tejido): secretan enzimas digestivas en una red de conductos que se unen al conducto pancreático principal, que se extiende a lo largo del páncreas.
- Endocrinas (2 % del tejido): secretan hormonas en el torrente sanguíneo. Entre ellas se encuentra la insulina, esencial para la regulación de los niveles de azúcar en la sangre.

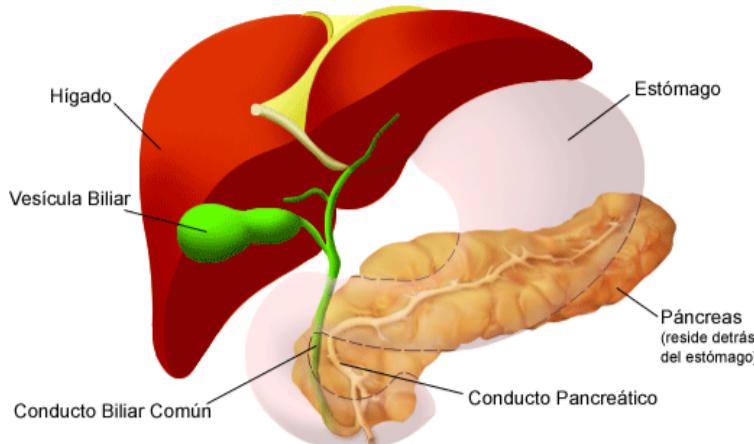


Figura 3.1: Anatomía del páncreas. *Fuente:* [1]

Debido a su función digestiva y su localización profunda, los síntomas iniciales del cáncer de páncreas suelen pasar desapercibidos. El síntoma más común es la pérdida de peso, generalmente acompañado de dolor epigástrico (entre el esternón y el ombligo). Otros signos incluyen náuseas, diarrea, fatiga, pérdida de apetito o debilidad general [47].

Esta compleja red anatómica se encuentra íntimamente relacionada con vasos sanguíneos mayores. La Figura 3.2 ilustra la irrigación arterial y el drenaje venoso del páncreas, elementos críticos en la evaluación de tumores.

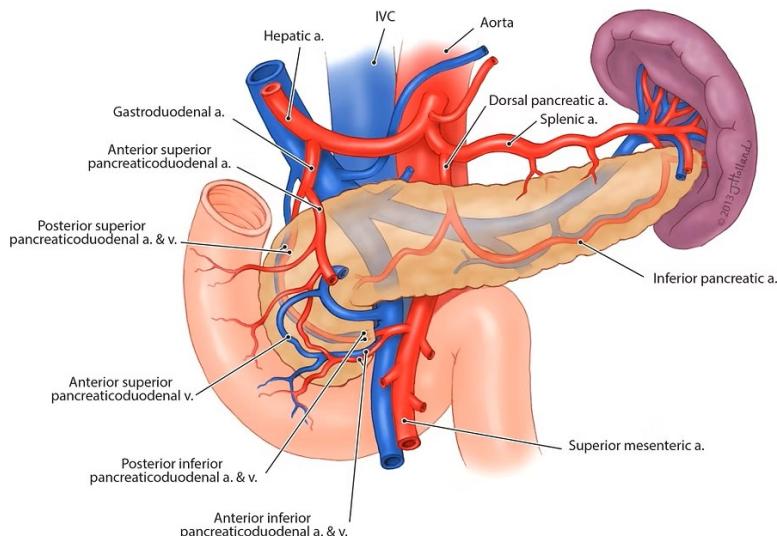


Figura 3.2: Irrigación arterial y drenaje venoso del páncreas (arterias en rojo y venas en azul). *Fuente:* [2]

Conocer la anatomía pancreática es fundamental no solo por su complejidad estructural, sino porque muchas patologías, como los tumores, presentan afectaciones específicas en regiones concretas del órgano. Además, al ser un órgano profundo hace que sea complicado detectar anomalías mediante exploración física, dificultando aún más el diagnóstico. A continuación, se analizan los distintos tipos de tumores pancreáticos y su relevancia clínica.

3.2. Tumores pancreáticos

El cáncer de páncreas, en especial el adenocarcinoma ductal pancreático (PDAC), presenta un comportamiento agresivo con una evolución clínica silenciosa, iniciada por la proliferación celular en el tejido pancreático. Para establecer un tratamiento adecuado, la primera evaluación crítica tras el diagnóstico por imagen es determinar la **estadificación** del tumor, es decir, identificar la localización, extensión y grado de avance de la enfermedad. Aunque clásicamente la estadificación se basa en el sistema TNM (Tumor, Nódulos, Metástasis), en la práctica clínica del cáncer pancreático la **resecabilidad** es un criterio más útil para la toma de decisiones.

3.2.1. Resecabilidad

La resecabilidad es un sistema de clasificación clínica que agrupa a los tumores en función de la posibilidad de ser extraídos mediante cirugía de manera eficaz y segura. Este criterio se establece principalmente según la relación anatómica del tumor con las estructuras vasculares adyacentes al páncreas (AMS, VMS, tronco celiaco y vena porta) [48]. Se distinguen tres grandes categorías:

- **Resecable:** el tumor está localizado exclusivamente en el páncreas o con mínima extensión y se puede extraer por completo. No hay metástasis ni afectación vascular detectable por imagen, especialmente de la VMS o vena porta [48].
- **Resecable borderline** (también llamado “resecable limítrofe”): el tumor muestra un contacto limitado con vasos sanguíneos, como una afectación menor de 180º de la circunferencia de la AMS, lo que podría permitir la resección tras tratamiento neoadyuvante (quimioterapia previa), aunque con mayor riesgo quirúrgico [48].
- **Irresecable:** presentan imposibilidad de ser extraídos por completo mediante cirugía. Incluye dos subtipos:
 - *Localmente avanzado:* el tumor rodea o infiltra completamente

vasos críticos (por ejemplo, afectación de la AMS mayor de 180° o afectación del tronco celiaco), impidiendo una resección segura.

- **Metastásico:** el cáncer se ha propagado a órganos distantes. En este caso, la cirugía puede usarse con intención paliativa, pero no curativa.

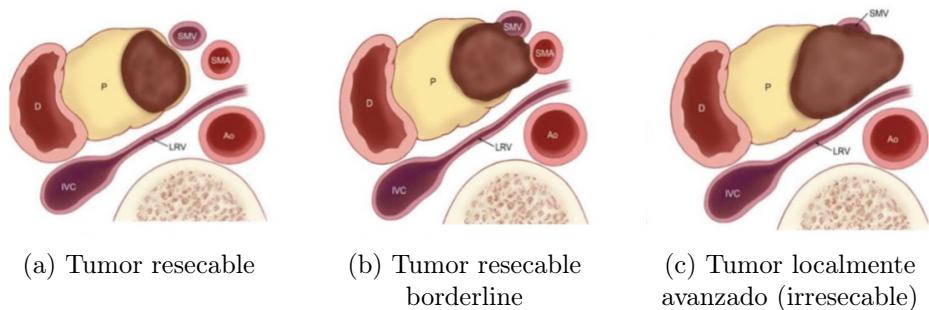


Figura 3.3: Resecabilidad de tumores (masa marrón). Fuente: [3]

Cabe destacar que algunos tumores inicialmente considerados resecables por imagen (TC) resultan irresecables durante la cirugía. Además, la evaluación de la afectación vascular mediante imagen tiene limitaciones importantes: se han documentado tasas de falsos positivos de hasta el 40 % [48].

Este enfoque basado en la resecabilidad es crucial en la planificación terapéutica y representa un reto técnico importante. Precisamente por ello, el objetivo de este proyecto se centra en facilitar la detección automática y precisa de las estructuras anatómicas implicadas, apoyando la toma de decisiones clínicas desde fases tempranas.

3.2.2. Tipos de tumores

En el contexto del cáncer de páncreas, la clasificación se realiza según el tipo de célula pancreática en la que se origina:

- Células exocrinas: son el origen de la gran mayoría de los tumores pancreáticos.
 - Adenocarcinoma ductal pancreático (PDAC): representa más del 90 % de los tumores malignos. Tiene mal pronóstico debido a su diagnóstico tardío, crecimiento infiltrativo y tendencia a invadir vasos y generar metástasis hepáticas o peritoneales [47].
 - Otros tipos menos comunes: cistoadenocarcinoma (formación de quistes), pancreatoblastoma (más frecuente en la infancia), carcinoma adenoescamoso (alta agresividad) y carcinoma coloide o mucinoso (rodeado de moco) [47].

- Células endocrinas: dan lugar a los tumores neuroendocrinos (TNE) pancreáticos. Son poco frecuentes y pueden ser funcionales o no funcionales, con evolución generalmente más lenta [47].

Además, existen lesiones benignas o precancerosas (que pueden convertirse en cáncer con el paso del tiempo si no son tratados), como la neoplasia mucinosa papilar intraductal (NMPI o IPMN por sus siglas en inglés), cuyo tratamiento quirúrgico trata de evitar la posible degeneración maligna [49].

Estas consideraciones establecen el contexto necesario para entender la importancia de contar con representaciones anatómicas precisas, como las que se pretenden obtener mediante segmentación aplicada a imágenes médicas, que se abordará en el siguiente apartado.

3.3. Segmentación semántica

La segmentación semántica es una tarea esencial en visión por computador (VC) que consiste en asignar una etiqueta a cada píxel (en 2D) o voxel (en 3D) de una imagen digital, clasificando cada región según su clase, como “páncreas” o “tumor” en el caso de imágenes médicas. A diferencia de la clasificación o la detección de objetos, que trabajan a nivel global o con cuadros delimitadores (*bounding boxes*), la segmentación semántica proporciona un nivel de detalle más fino clave para tareas clínicas.

Se diferencia de la *segmentación de instancias* en que no distingue entre objetos individuales de la misma clase. Por ejemplo, dos tumores en una imagen serían etiquetados como “tumor” en segmentación semántica, mientras que la segmentación por instancias los identificaría como entidades separadas (“tumor 1” y “tumor 2”). En medicina, la segmentación semántica suele ser suficiente para fines clínicos como el diagnóstico o la planificación quirúrgica, donde lo importante es la delimitación precisa de las estructuras, no la distinción entre múltiples instancias separadas.

En la Figura 3.4 se muestra una comparativa entre estos tipos de tareas que aborda la visión por computador.

La segmentación semántica (Figura 3.4c) resulta especialmente útil en medicina, donde las imágenes obtenidas mediante tomografía computarizada (TC), resonancia magnética (RM) u otras modalidades necesitan ser analizadas con alta resolución espacial para detectar estructuras y anomalías.

Tradicionalmente, la segmentación se realizaba de forma manual por expertos, lo que es un proceso costoso, sujeto a variabilidad interobservador y difícil de escalar. El desarrollo de algoritmos automáticos, y en especial el uso de técnicas de aprendizaje profundo, ha supuesto un gran avance en la segmentación semántica. Modelos como *U-Net*, *DeepLab* o *3D U-Net* han

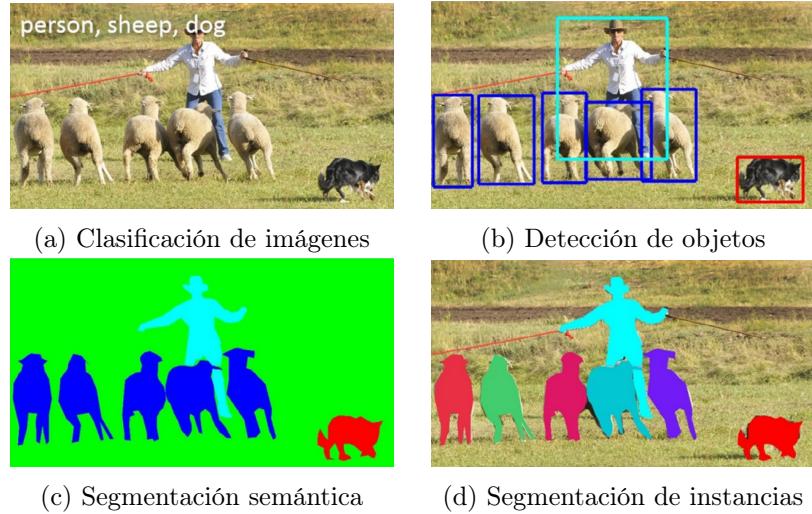


Figura 3.4: Comparativa de tareas de visión por computador. *Fuente:* [4]

demonstrado una alta eficacia en distintos retos de segmentación médica, convirtiéndose en herramientas clave para el análisis de imagen clínica [16].

Desde el punto de vista computacional, la segmentación semántica puede plantearse como una tarea de clasificación a nivel de píxel. A partir de una imagen de entrada de tamaño $W \times H \times C$ (ancho, alto, canales), el objetivo es generar un mapa de segmentación de tamaño $W \times H \times 1$, donde cada píxel contiene la etiqueta de clase a la que pertenece, con n clases posibles.

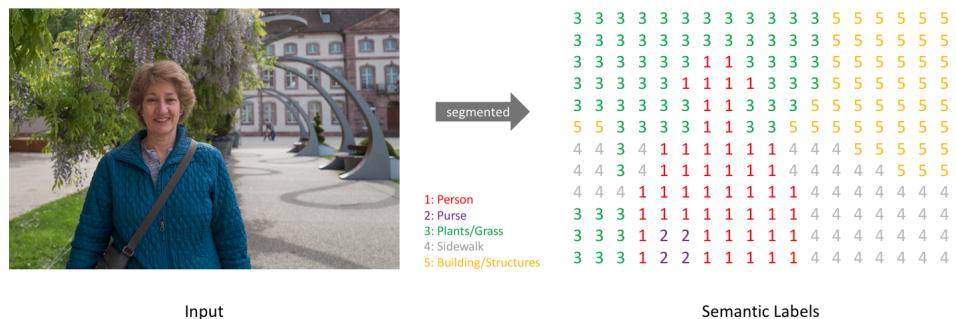


Figura 3.5: Imagen de entrada y segmentación. *Fuente:* [5]

Durante el entrenamiento, estas etiquetas suelen representarse mediante codificación *one-hot*, generando una máscara binaria para cada clase, lo que produce una salida de tamaño $W \times H \times n$. El mapa de segmentación final puede obtenerse aplicando la operación *argmax* sobre la dimensión de clases.

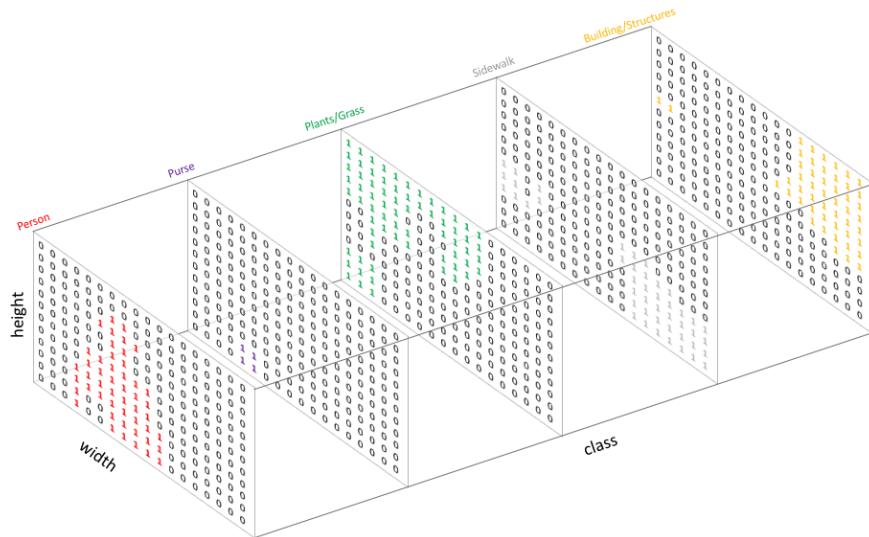


Figura 3.6: Codificación one-hot de la segmentación. *Fuente:* [5]

Al ser una tarea supervisada, el entrenamiento requiere una imagen de entrada y su correspondiente mapa de segmentación de referencia (*ground truth*). Cada canal de esta anotación puede entenderse como una máscara que, al superponerse sobre la imagen original, permite visualizar qué regiones corresponden a una determinada clase anatómica.

En la siguiente sección se describen algunas de las principales aplicaciones clínicas de esta técnica en el ámbito del cáncer pancreático.

3.4. Relevancia y aplicaciones reales

Una vez definida la segmentación semántica como la asignación de una etiqueta a cada píxel o voxel de una imagen, su utilidad en el ámbito médico, especialmente en el cáncer de páncreas, se traduce en múltiples aplicaciones prácticas. Estas aplicaciones no solo optimizan la eficiencia clínica, sino que también pueden impactar directamente en el pronóstico y la calidad de vida del paciente.

Modelado 3D para planificación quirúrgica y evaluación de resecabilidad

La segmentación precisa del páncreas, los tumores y los vasos adyacentes a partir de imágenes de tomografía computarizada (TC) permite la creación de modelos tridimensionales, tanto virtuales como impresos, específicos para cada paciente. Estos modelos transforman la comprensión espacial de

la anatomía pancreática y, en particular, de la relación entre el tumor y las estructuras vasculares circundantes como la arteria y vena mesentérica superior, el tronco celíaco o la vena porta.

La visualización 3D facilita a los cirujanos la evaluación preoperatoria de la resecabilidad tumoral y permite planificar la intervención quirúrgica con mayor precisión. La Figura 3.7 muestra un ejemplo de modelado tridimensional basado en segmentaciones anatómicas obtenidas mediante TC.

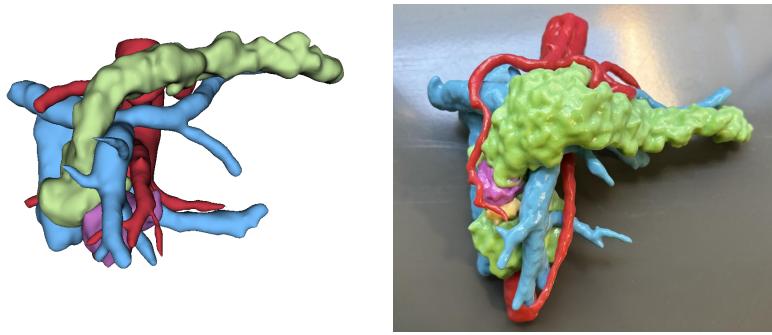


Figura 3.7: Modelado 3D a partir de segmentaciones. Páncreas en *verde*, tumor en *morado*, arterias en *rojo* y venas en *azul*. No corresponden al mismo paciente. *Fuentes:* a) elaboración propia con 3D Slicer [6], b) Proyecto DIGIPanCa¹ (privada)

Este tipo de modelado constituye una de las principales aplicaciones clínicas actuales de la segmentación semántica en oncología pancreática, pero no la única.

Monitorización de la evolución del tratamiento y reducción del daño a tejidos sanos

La segmentación semántica volumétrica permite cuantificar con precisión el volumen del tumor y otros tejidos de interés a lo largo del tiempo. Realizando segmentaciones en distintos momentos, es posible monitorizar la respuesta del tumor a tratamientos como la quimioterapia o la radioterapia [30].

Esta capacidad facilita la evaluación objetiva de si el tumor está encogiéndose, creciendo o si hay cambios en su relación con estructuras críticas. Además, al delimitar de forma precisa las zonas a tratar, se puede minimizar la exposición de estructuras sanas, reduciendo el daño colateral y los efectos adversos del tratamiento. Esta información objetiva también permite a

los equipos médicos decidir con mayor fundamento si continuar, ajustar o cambiar la estrategia terapéutica [50].

Entrenamiento y educación médica

Los modelos anatómicos generados a partir de segmentaciones también son herramientas muy útiles en entornos formativos. Tanto los modelos virtuales como los impresos en 3D permiten estudiar con precisión las variaciones anatómicas, la localización de tumores y la planificación de procedimientos quirúrgicos complejos. Esto resulta especialmente valioso para la formación de nuevos radiólogos, cirujanos o personal en formación en disciplinas relacionadas con la anatomía quirúrgica [50].

Como se ha visto, la segmentación semántica permite aplicaciones clínicas avanzadas con impacto directo en la planificación quirúrgica, la monitorización del tratamiento y la educación médica. Sin embargo, todas estas aplicaciones dependen de la disponibilidad de imágenes médicas de alta calidad, especialmente de tomografía computarizada, que constituye la principal fuente de datos en el cáncer de páncreas. Por ello, en el siguiente apartado se introducen los fundamentos técnicos de la TC, base esencial para comprender cómo se adquieren y procesan las imágenes utilizadas en este proyecto.

3.5. Fundamentos de tomografía computarizada

La tomografía computarizada (TC) es una técnica de imagen médica basada en la adquisición de múltiples proyecciones de rayos X desde diferentes ángulos, que mediante algoritmos de reconstrucción computacional genera cortes axiales detallados del cuerpo humano [51]. A lo largo de las últimas décadas, la TC se ha convertido en una herramienta fundamental en el diagnóstico y estadificación de enfermedades abdominales.

Planos anatómicos

Además de generar imágenes en forma de cortes o secciones del cuerpo en el plano axial (horizontal), también es habitual realizar reconstrucciones en otros planos ortogonales a partir del volumen tridimensional, como el plano coronal (frontal) y el plano sagital (lateral) mediante técnicas de reconstrucción multiplanar (*MPR*), como refleja la Figura 3.8.

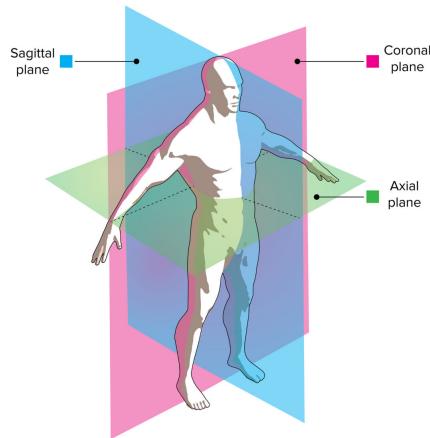


Figura 3.8: Planos anatómicos de imagen de TC. *Fuente:* [7]

Sistema de coordenadas

Las imágenes de TC se almacenan siguiendo sistemas de coordenadas anatómicos, que definen la orientación espacial del volumen en relación con el cuerpo del paciente. Estos sistemas se basan en tres ejes ortogonales:

- X (Left–Right): de izquierda a derecha.
- Y (Posterior–Anterior): de la espalda hacia el pecho.
- Z (Inferior–Superior): de los pies hacia la cabeza.

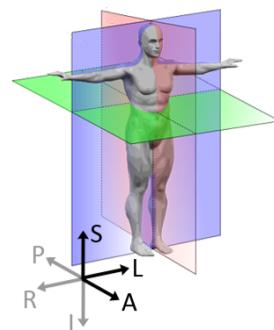


Figura 3.9: Sistema de coordenadas anatómicas. *Fuente:* adaptado de [8]

Dependiendo del fabricante o del protocolo de adquisición, los datos pueden almacenarse con diferentes orientaciones, como RAS, LPS, etc. Es necesario tener en cuenta este aspecto a la hora de trabajar con este tipo de datos.

Unidades Hounsfield y ventana de visualización

Cuando se hace una TC a un paciente, realmente se está midiendo la transmisión de los rayos X a través de él en múltiples proyecciones, por lo que los valores que se le asignan a cada píxel de las imágenes están directamente relacionados con la atenuación en el tejido correspondiente por la densidad física del mismo. Es por ello que el principio básico de su funcionamiento es la medida de las intensidades de estos rayos antes y después de pasar por el paciente [51].

Las imágenes de TC se cuantifican en unidades Hounsfield (HU). Los valores en esta escala se expresan con la atenuación del agua a temperatura ambiente como base:

$$HU = \frac{\mu_{material} - \mu_{agua}}{\mu_{agua}} \times 1000 \quad (3.1)$$

Donde μ representa el coeficiente de atenuación lineal. El agua se ha establecido como 0 HU, pues $\mu_{material} = \mu_{agua}$, mientras que el aire tiene un valor de -1000 HU, pues $\mu_{material} = 0$. Debido a esto se puede decir que cada incremento de 1 HU se asocia con un incremento del 0,1 % de la atenuación del material. Para un hueso, este valor puede superar las 1000 HU [51].

Como las imágenes contienen varios miles de tonos de grises, normalmente se habla de “Ventana de HU” para referirse a los valores de esta escala en que se enfoca, que permite ajustar el contraste y el brillo en función del objetivo. Con el nivel de ventana o *window level* (WL) se establece el punto central a mostrar, mientras que con el ancho de ventana o *window width* (WW) se define el rango de valores HU a mostrar. Una ventana más amplia permite visualizar un rango mayor de densidades, mientras que una más estrecha aumenta el contraste. Es otro factor a tener en cuenta esto a la hora de tratar este tipo de datos.

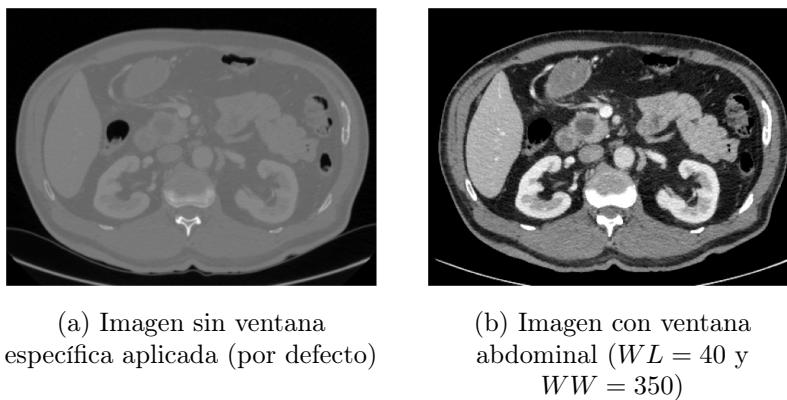
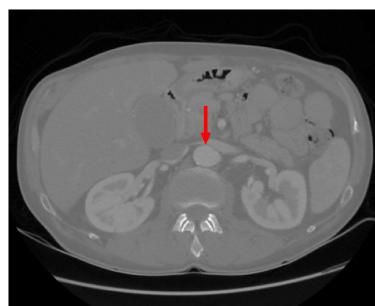


Figura 3.10: Diferencia en la visualización según HU. *Elaboración propia*

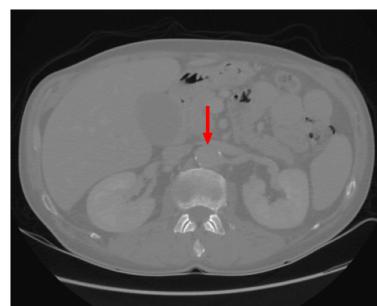
Fases de contraste

Para mejorar la calidad de la imagen y diferenciar mejor tejidos sanos de lesiones tumorales, se emplea el “realce de contraste”. Una TC con contraste intravenoso se realiza en diferentes fases temporales, mientras circula la sustancia suministrada por la sangre. Las dos fases principales y con las que ha trabajado el Hospital Universitario Clínico San Cecilio son:

- **Fase arterial** (20–35 segundos tras la inyección de contraste): resalta principalmente las arterias, como la aorta y sus ramas, y permite identificar lesiones hipervasculares y metástasis hipervasculares.
- **Fase portal venosa** (60–90 segundos tras la inyección): proporciona un equilibrio entre realce de órganos sólidos y los vasos centrales.



(a) Imagen TC en fase arterial



(b) Imagen TC en fase portal venosa

Figura 3.11: Diferencia entre fases. La flecha roja señala la arteria.
Elaboración propia

Resumen. Los principios físicos de la TC, la interpretación de las fases de contraste y la comprensión de las unidades *Hounsfield* específicas del páncreas sano y tumoral constituyen la base para analizar el conjunto de datos de este proyecto de segmentación automática.

3.6. Descripción del conjunto de datos y aspectos legales

El conjunto de datos ha sido proporcionado por el Hospital Universitario Clínico San Cecilio de Granada. Consta de un total de 116 tomografías computarizadas, recibidas en dos partes: inicialmente 88 pacientes y los 28 restantes el día 25 de abril de 2025. Es por ello que se ha considerado el

primer lote de 88 pacientes como conjunto de entrenamiento (entrenamiento y validación) y el segundo lote como conjunto de prueba o *test*.

Dado que este TFG nace para apoyar el proyecto del hospital, a continuación se detallan algunos aspectos del mismo como criterios seguidos para obtención de datos y procesamientos aplicados.

Criterios de inclusión

Se incluyen en el conjunto de datos del proyecto pacientes tratados en el Hospital Universitario Clínico San Cecilio de Granada o en el Hospital Universitario La Fe de Valencia entre 2018 y 2024 con diagnóstico de:

- Adenocarcinoma ductal de páncreas (PDAC): tumores resecables y *borderline* resecables, e irresecables por metástasis.
- Tumor neuroendocrino de páncreas.
- Tumor neuroendocrino con metástasis hepáticas que sean resecables.

Criterios de exclusión

De esta serie se excluirán aquellos casos sin estudios de imagen, con imágenes defectuosas, incompletos o con ausencia de informes biológicos relevantes para el diagnóstico, además de casos de cáncer irresecable por afectación vascular.

Edad de la población

La población afectada con cáncer de páncreas será mayor de 18 años al momento del diagnóstico.

Procesamiento de los datos

Una vez son realizados los escaneos en el hospital, el volumen TC es sometido a una serie de procesos para construir el conjunto de datos a utilizar y para obtener los modelos impresos en 3D, realizados por un radiólogo experto. El flujo de trabajo es el siguiente:

1. Segmentación manual. Páncreas, tumor, arterias y venas se segmentan de manera manual por medio de la aplicación privada de *Philips*

*Portal Intellispace*². Genera una segmentación del paciente en formato DICOM.

2. Exportación a modelo 3D. La segmentación se exporta como modelo 3D, decidiendo la suavidad con la que se renderizan las estructuras. Esta puede ser sutil (con pérdida de exactitud mínima), exacta (para estructuras complejas) y alta (gran superficie).
3. El DICOM obtenido se anonimiza con el software *Slicer 3D* [6, 52], resultando en una nueva segmentación en formato NIfTI.
4. Continuando con la parte del modelado 3D, se refina el modelo que se va a imprimir, con el objetivo de suavizar, unir partes para que no queden flotando, etc. Esto se hace mediante *Meshmixer*³.
5. Con el software *GrabCAD*⁴ se eligen los colores, material de impresión, etc, y se imprime el modelo segmentado.

Atendiendo exclusivamente a la preparación y construcción del dataset, sólo son necesarios los puntos 1 y 3, mientras que el resto de pasos detallan el flujo completo desde que se realiza TC a un paciente hasta que se tiene un modelo 3D real de sus órganos y estructuras de interés.

3.6.1. Legalidad

Los datos médicos en formato NIfTI utilizados en este proyecto han sido proporcionados por el Hospital Universitario Clínico San Cecilio tras un proceso de conversión desde DICOM y anonimización llevado a cabo por el equipo clínico de dicha institución. Este proceso ha eliminado todos los identificadores personales directos e indirectos, conforme a las directrices del *Considerando 26* del *Reglamento (UE) 2016/679 (RGPD)*⁵, que establece que los datos anonimizados de forma irreversible quedan fuera del ámbito de aplicación del *RGPD*. En consecuencia, y de acuerdo con la *Ley Orgánica 3/2018, de Protección de Datos Personales y garantía de los derechos digitales*, específicamente su *Disposición adicional decimoséptima, apartado 2, letra e)*⁶, no existe ninguna probabilidad razonable de reidentificación de los sujetos, lo que garantiza el cumplimiento de la normativa vigente en materia de protección de datos. Esto también está respaldado por la *Agencia*

²<https://www.philips.es/healthcare/medical-specialties/healthcare-informatics>

³<https://apps.autodesk.com/FUSION/en/Detail/Index?id=4108920185261935100>

⁴<https://grabcad.com/>

⁵<https://gdpr-text.com/es/read/recital-26/>

⁶<https://www.boe.es/buscar/act.php?id=BOE-A-2018-16673>

Española de Protección de Datos (AEPD)⁷.

3.7. Metodología

La metodología de este proyecto se enmarca dentro de un enfoque de investigación empírica, tratando de resolver un problema práctico tanto en el ámbito de la ingeniería informática como de la medicina.

Software

El desarrollo del código se ha llevado a cabo utilizando Python 3.9.12⁸, un lenguaje de programación interpretado flexible y ampliamente utilizado en el ámbito de la Inteligencia Artificial, con bibliotecas preparadas para todo tipo de problemas.

Para implementar los modelos, se ha hecho uso de PyTorch 1.13.1⁹, una biblioteca de código abierto para Deep Learning y que permite el uso de GPUs (*Graphics Processing Units*, unidades de procesamiento de gráficos) para acelerar las operaciones con tensores además de una gran flexibilidad. Se ha consultado su documentación, disponible en [53].

El desarrollo del proyecto se ha dividido en dos fases. En el primer enfoque de trabajo se ha realizado una implementación manual de todo el ecosistema: clases para el conjunto de datos, preprocesamiento de datos, aumento de datos (*data augmentations*, que consiste en generar nuevos datos mediante transformaciones aleatorias sobre los existentes), modelos, métricas, entrenadores y evaluadores, visualizaciones, etc. Gracias a esta etapa, se ha comprendido mejor la naturaleza del problema (manejo y preparación de datos, metadatos de los archivos, contexto clínico...) y demostrada la necesidad de utilizar técnicas más avanzadas en todos los aspectos para mejorar el rendimiento de los modelos.

Es por ello que, para la segunda etapa se ha seguido un enfoque más especializado y optimizado, incorporando el uso del *framework* MONAI (*Medical Open Network for Artificial Intelligence*)¹⁰ [54]. En concreto se ha utilizado la versión 1.4.0. Se trata de un framework basado en PyTorch para Deep Learning aplicado en imagen médica, por lo que contiene innumerables implementaciones optimizadas y beneficiosas, principalmente en lo que a transformaciones de datos y modelos se refiere. Gracias a esto se ha podido

⁷<https://www.aepd.es/prensa-y-comunicacion/blog/anonimizacion-y-seudonimizacion>

⁸<https://www.python.org/>

⁹<https://pytorch.org/>

¹⁰<https://monai.io/>

optimizar todo el proceso y mejorar los resultados obtenidos.

Además, se han utilizado otras bibliotecas de Python, entre las que se incluyen:

- NiBabel 5.3.2¹¹ para trabajar con datos en formato NIfTI.
- Matplotlib 3.9.4¹² para visualizaciones 2D y 3D de imágenes y segmentaciones.
- NumPy 1.26.4¹³ para operaciones sobre matrices multidimensionales.
- Albumentations 2.0.7¹⁴ para data augmentation en 2D.
- TorchIO 0.20.8¹⁵ para data augmentation en 3D.
- scikit-learn 1.6.1¹⁶ y scikit-image 0.24.0¹⁷ para otras utilidades.

El desarrollo se ha llevado a cabo con el editor de código Visual Studio Code¹⁸ junto con Jupyter¹⁹ (tanto integrado en VS Code como en su versión propia) para realizar pruebas y visualizaciones rápidas en *Jupyter Notebooks*.

Todo el código desarrollado se encuentra disponible en el siguiente repositorio de GitHub: <https://github.com/Alvarorlz12/TFG>.

Para complementar el proceso, se ha hecho un uso exhaustivo de 3D Slicer [6], que permite visualizaciones, procesamiento, segmentaciones y análisis de imágenes médicas.

Hardware

El entorno principal para el desarrollo de las implementaciones ha sido un HP Laptop 15s-fq2042ns con las siguientes especificaciones:

- Procesador (CPU): Intel® Core™ i7-1165G7 a 2.80GHz.
- Memoria RAM: 16 GB.

¹¹<https://nipy.org/nibabel/>

¹²<https://matplotlib.org/>

¹³<https://numpy.org/>

¹⁴<https://albumentations.ai/>

¹⁵<https://torchio.readthedocs.io/>

¹⁶<https://scikit-learn.org/stable/>

¹⁷<https://scikit-image.org/>

¹⁸<https://code.visualstudio.com/>

¹⁹<https://jupyter.org/>

Para las tareas de entrenamiento y evaluación de los modelos de Deep Learning, se ha contado con acceso a los servidores de cómputo de alto rendimiento (NGPU) proporcionados por la Universidad de Granada (UGR), cuyas máquinas se encuentran ubicadas en CPD Santa Lucía.

La infraestructura NGPU de la UGR cuenta con un clúster de nodos de cómputo que incluyen diversas configuraciones de GPUs como GTX Titan X Pascal, GTX Titan Xp, RTX 2080 Ti, Titan RTX, Tesla V100 (32GB) y Tesla A100 (40GB). Estas han permitido entrenar tanto modelos 2D como 3D.

3.8. Planificación y diario de trabajo

En esta sección se presenta la planificación y el diario de trabajo del proyecto. Se ha dividido en tareas, desarrolladas y completadas a lo largo del tiempo. Estas tareas son:

- Aprendizaje clínico o médico. Al comienzo del proyecto, fundamental para familiarizarse con el dominio del problema y tecnologías: comprender el campo médico, sus terminologías procedimientos, especialmente para el cáncer de páncreas.
- Revisión de la literatura. Aunque iniciada al comienzo, ha sido una tarea intermitente.
- Implementación propia. Fase inicial del desarrollo del software, iniciada una vez se tenían los datos a disposición.
- Implementación con MONAI. Evolución o mejora de la implementación inicial, coincidiendo temporalmente con el desarrollo de la implementación propia.
- Experimentación. Desde que se tenía el ecosistema inicial implementado (propio) y los primeros modelos, hasta las últimas pruebas (con MONAI).
- Memoria. Redacción de la memoria, comenzando con las partes conocidas (contexto médico, teoría, etc), y dejando la experimentación y la maquetación para las últimas semanas.

Gracias a este enfoque, se ha permitido una ejecución de experimentos al mismo tiempo que se continuaba con el desarrollo de las propuestas y se revisaba el estado del arte para justificar y encaminar las decisiones.

A continuación, se muestra un diagrama de Gantt de la planificación llevada a cabo en este proyecto.

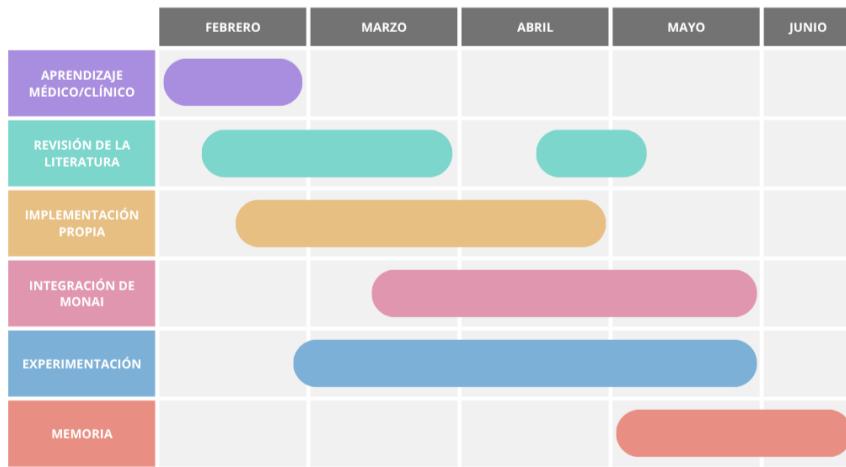


Figura 3.12: Diagrama de Gantt del proyecto, correspondiente al periodo Febrero–Junio de 2025

Para llevar un mejor control del trabajo y las subtareas a realizar, especialmente durante la implementación propia, ya que implica inherentemente una mayor cantidad de subtareas, se ha utilizado Notion²⁰. Este espacio de trabajo permite llevar un registro del estado de las tareas y clasificarlas por características específicas (tipo de tarea, ficheros que afecta, prioridad, fecha límite, etc), como por ejemplo en la Figura 3.13.

Fix: shape incorrecto con None en transforms y augment

Responsable: ALVARO RUIZ LOPEZ | Estado: Listo | Plazo: 18/03/2025

Comentarios: Añadir un comentario...

Descripción de la tarea

Corregir error que produce el emplear `transform=None` y `augment=None`, donde la salida del `getitem` no contiene canal de color y se produce error en el entrenamiento.

Se propone modificar dicho método para que permita el uso correcto de `transforms` y `augmentations` de manera flexible y dinámica, permitiendo una ejecución con estas transformaciones definidas en el fichero de configuración `.yaml` o no (`None`).

Figura 3.13: Ejemplo de tarea en Notion

²⁰<https://www.notion.com/es-es>

Además de tareas, permite llevar un recuento de las horas dedicadas. Con esto, la distribución final se refleja en la siguiente figura:



Figura 3.14: Estimación de horas dedicadas al proyecto según registro en Notion

Como aclaración, las horas introducidas en Notion son una estimación realizada al final de cada jornada o tras la finalización de una tarea significativa. Por tanto, pueden presentar una ligera desviación respecto al tiempo real empleado, aunque se consideran lo suficientemente precisas para ofrecer una visión general de la carga de trabajo. También hay que destacar que gran parte del tiempo invertido ha sido en revisión de literatura y contextualización clínica, pues requería más dedicación al ser ajeno al conocimiento propio del grado.

Una vez aclarado esto, se puede justificar que se han cubierto las 300 horas de trabajo estipuladas para 12 créditos ECTS, correspondientes al Trabajo de Fin de Grado.

El uso de Notion ha supuesto una mejora en la organización y gestión del tiempo durante el desarrollo del proyecto y principalmente al comienzo del mismo.

3.9. Presupuesto

En esta sección se detallará el presupuesto necesario para llevar a cabo el proyecto. Incluye horas del autor (alumno) y del supervisor (tutor), y de

infraestructura (cómputo en los servidores). Por el contrario, se excluyen costes personales propios como portátil, alquiler o luz.

Entonces, por parte del alumno se ha trabajado 415 horas y por parte del tutor se han dedicado 25 horas. La duración total de los experimentos ha sido 670 horas aproximadamente. Las fuentes para los costes son Glassdoor²¹, SalaryExpert²² y Google Cloud²³ respectivamente. El siguiente cuadro recoge el coste final:

En total, el coste atribuido al desarrollo del proyecto es de 7.026,6 €.

Sostenibilidad

Además del presupuesto monetario, es necesario velar por una informática sostenible, por lo que a continuación se presenta la sostenibilidad de este proyecto.

A partir de las GPUs utilizadas del servidor, se ha estimado el consumo medio energético en 0,25 kW. Por tanto, el consumo total estimado de los experimentos es:

$$670 \text{ h} \cdot 0,25 \text{ kW} = 167,5 \text{ kWh}$$

De acuerdo con el estudio de la AIB (*Association of Issuing Bodies*) sobre el *mix residual* Europeo de 2024²⁴, el factor de emisión fue 0,2922 kg CO₂e/kWh. Entonces:

$$167,5 \text{ kWh} \cdot 0,2922 \text{ kg CO}_2\text{e/kWh} = 48,94 \text{ kg}$$

Como conclusión, en base a las estimaciones realizadas, los experimentos realizados para este proyecto han producido una emisión de 48,94 kg de CO₂.

3.10. Descripción de las propuestas elaboradas

La propuesta se centra en el uso de Redes Neuronales Convolucionales (CNN) bidimensionales y tridimensionales. A diferencia de los modelos 2D, los modelos 3D son capaces de capturar dependencias espaciales y temporales en volúmenes y preservar la información contextual intercorte.

Se explorarán y adaptarán diversas arquitecturas de CNN, incluyendo tanto enfoques 2D como 3D, reconociendo la importancia de ambos en la investigación y desarrollo de segmentación médica.

²¹https://www.glassdoor.es/Sueldos/junior-software-engineer-sueldo-SRCH_K00%2C24.htm

²²<https://www.salaryexpert.com/salary/job/software-developer/spain>

²³Promediado de Google Cloud: <https://cloud.google.com/compute/gpus-pricing>

²⁴<https://www.aib-net.org/facts/european-residual-mix>

Las arquitecturas 2D consideradas son:

- U-Net. Una arquitectura referente en segmentación médica, con una distintiva forma de “U” y gran capacidad para combinar información de bajo y alto nivel, lo que mejora la precisión. Se basa en el uso de un codificador para extraer características y un decodificador para reconstruir la segmentación, unidos por conexiones de salto que fusionan características de diferentes niveles.
- DeepLabV3. Otra arquitectura destacada para la segmentación semántica, diseñada para abordar la variación de escala de los objetos en la imagen. Sus características principales incluyen la Convolución *Atrous* (atrosa o dilatada) que expande el campo de visión sin aumentar el número de parámetros, y el *Atrous Spatial Pyramid Pooling* (ASPP) que aplica convoluciones atrous en paralelo con diferentes tasas de dilatación para capturar contexto a múltiples escalas. Utiliza ResNet-101 como backbone.

Entre las arquitecturas 3D consideradas se incluyen:

- U-Net 3D. Una extensión directa de U-Net 2D, adaptando las operaciones convolucionales y de *pooling* al espacio tridimensional.
- Attention U-Net. Integra Puertas de Atención (*Attention Gates*) para permitir que la red se enfoque en regiones relevantes, suprimiendo aquellas irrelevantes.
- DynUNet. Una implementación de MONAI basada en la filosofía nnU-Net, adaptable y robusta, que configura automáticamente la profundidad y resolución del modelo para distintas tareas de segmentación médica volumétrica.
- V-Net. Diseñada específicamente para segmentación 3D, utilizando convoluciones tridimensionales y deconvoluciones en su arquitectura de codificador-decodificador en forma de “V”.
- SegResNet. Una arquitectura asimétrica con un codificador más grande y un decodificador más pequeño, que incorpora bloques residuales y normalización de grupo, originalmente propuesta para segmentación de tumores cerebrales pero adaptable a otros contextos.
- UNETR (UNet TRansformers). Un modelo que combina un codificador basado en *transformers* para capturar el contexto global y dependencias de largo alcance, con un decodificador basado en CNN y conexiones de salto para el contexto local, diseñado para imágenes médicas 3D.

- Swin UNETR. Una mejora de UNETR que aborda la limitación de resolución fija en el codificador, extrayendo características multiescala mediante *Swin Transformers* y atención autodesplazada.

Para abordar la heterogeneidad de los datos (variabilidad en el grosor de corte, desequilibrio de clases, bajo contraste entre tejidos) y optimizar el rendimiento del modelo, se implementará un pipeline de preprocesamiento robusto utilizando la biblioteca MONAI. Las técnicas clave incluyen:

- Homogeneización. Reorientación de los volúmenes a una orientación estándar (RAS o RPS) y homogeneización del espaciado de los véxoles a una resolución isotrópica de 1 mm, lo que mejora la generalización y coherencia espacial.
- Normalización de intensidades. Aplicación de una ventana de intensidades a un rango específico ($[-150, 250]$ HU), basado en recomendaciones clínicas y la distribución observada en el conjunto de datos, seguida de una normalización al rango $[0, 1]$.
- Aumento de datos (*Data augmentation*). Para mitigar la escasez de datos y prevenir el sobreajuste, se aplicarán transformaciones aleatorias dinámicas durante el entrenamiento, incluyendo deformaciones elásticas tridimensionales, rotaciones, *zooms*, suavizado gaussiano, escalados y desplazamientos de intensidades, y *flips* (o volteos). La estrategia de muestreo por parches o subvolúmenes se guiará por máscaras dilatadas morfológicamente para asegurar un contexto anatómico suficiente y priorizar el muestreo de clases minoritarias como el tumor.

Con estas técnicas avanzadas de preprocesamiento de datos médicos se garantiza entrenar modelos de manera coherente y eficiente.

Para el entrenamiento de los modelos, se emplearán distintas funciones de pérdida para guiar el aprendizaje, como la *Dice Loss* (robusta ante el desequilibrio de clases), *Cross-Entropy* (fundamental para clasificación a nivel de píxel) y *Focal Loss* (diseñada para resolver el desbalance de clases, enfocando el aprendizaje en las muestras difíciles), así como sus combinaciones ponderadas. Los algoritmos de optimización incluirán *Adam* (o *AdamW*) y variantes del *Mini-Batch Gradient Descent*, con estrategias de ajuste de la tasa de aprendizaje (*LR Schedulers*) para optimizar la convergencia. Técnicas de regularización como *Dropout*, normalización de capas (*Normalization*), y *Early Stopping* se aplicarán para evitar el sobreajuste y mejorar la estabilidad del entrenamiento.

Capítulo 4

Análisis de los datos y preprocesamiento

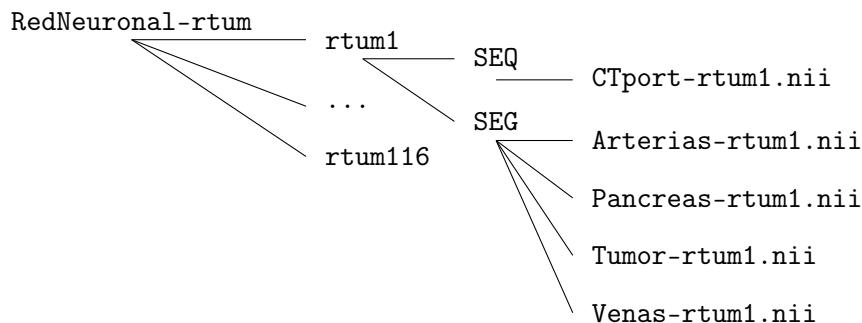
En este capítulo, se presenta un análisis exhaustivo de los datos proporcionados por el Hospital Clínico Universitario San Cecilio de Granada y las técnicas de preprocesamiento que se le han aplicado.

4.1. Descripción detallada y análisis de datos

Como se mencionaba en la Sección 3.6, el dataset completo está compuesto por 116 volúmenes TC, esto es, pacientes.

El formato de estos datos es `.nii` (NIfTI), con una resolución de 512×512 píxeles y un grosor de corte variable, entre 1 mm y 2,5 mm. La variabilidad en el grosor de corte hace que por paciente se tengan desde 40 hasta 160 cortes o *slices* aproximadamente. Así, la dimensión de cada TC es $512 \times 512 \times D$ vóxeles, siendo D la profundidad o número de cortes.

El dataset está estructurado de la siguiente manera:



`RedNeuronal-rtum` contiene todos los pacientes divididos por carpetas,

donde `rtumN` corresponde a la carpeta del paciente N. Para cada caso, se encuentra en `SEQ` la secuencia o volumen TC. y en `SEG` las segmentaciones de páncreas, tumor, arterias y venas. Independientemente de la fase en la que se observen mejor, todas las secuencias están nombradas como `CTport-rtumN.nii` para facilitar la experimentación y los entrenamientos.

4.1.1. Estadísticas del dataset

En este apartado se muestran una serie de estadísticas sobre el conjunto de datos de entrenamiento, es decir, el primer lote de 88 volúmenes TC.

Tipo de tumor

La siguiente tabla muestra la distribución de los tipos de tumor presentes en los casos de entrenamiento:

Cuadro 4.1: Distribución de tipos de tumor en casos de entrenamiento

Tipo de tumor	Número de casos	Proporción (%)
PDAC	44	50,00 %
PDAC borderline	31	35,23 %
TNE	9	10,23 %
NMPI	1	1,14 %
Ampuloma	1	1,14 %
Gist duodenal	1	1,14 %
Colangiocarcinoma	1	1,14 %

De cara a tener divisiones del conjunto en subconjuntos de entrenamiento y validación, los casos con tipo de tumor NMPI, ampuloma, gist duodenal y colangiocarcinoma han sido englobados en una misma categoría denominada “Otros”, de modo que al ser casos no representativos del dataset no influyen en los *splits*. Con esta condición, la distribución resultante se presenta en la siguiente tabla:

Cuadro 4.2: Distribución de tipos de tumor agrupados (categoría “Otro”) en casos de entrenamiento

Tipo de tumor	Número de casos	Proporción (%)
PDAC	44	50,00 %
PDAC borderline	31	35,23 %
TNE	9	10,23 %
Otro	4	4,55 %

Fase de TC

La distribución de fases de la tomografía computarizada para los 88 pacientes se muestra en la siguiente tabla:

Cuadro 4.3: Distribución de fases de TC en casos de entrenamiento.

Fase TC	Número de casos	Proporción (%)
Doble	42	47,73 %
Portal	27	30,68 %
Arterial	19	21,59 %

Los volúmenes etiquetados con fase “Doble” indican buen reconocimiento y visualización tanto en la fase portal como arterial, sin embargo y como se ha mencionado anteriormente, se utilizan para todos los casos CTport-rtumN, de modo de estos 42 casos también se está tomando la fase portal en lugar de la arterial. Por tanto, la distribución efectiva de las fases utilizadas queda como sigue:

Cuadro 4.4: Distribución de fases de TC realmente utilizadas en casos de entrenamiento.

Fase TC	Número de casos	Proporción (%)
Portal	69	78,41 %
Arterial	19	21,59 %

Dificultad de segmentación manual

Se dispone de una estimación de la dificultad de segmentación manual de cada uno de los pacientes del conjunto de entrenamiento. Para cada paciente anotado (segmentado) durante el procedimiento descrito en la Sección 3.6, el equipo clínico del Hospital Universitario Clínico San Cecilio ha otorgado una puntuación en el rango [1, 3], donde 1 equivale a “Fácil” y 3 a “Difícil”. La distribución porcentual se puede observar en la siguiente tabla:

Cuadro 4.5: Distribución de dificultad de segmentación manual en casos de entrenamiento.

Dificultad de segmentación	Número de casos	Proporción (%)
1 (<i>Fácil</i>)	48	54,55 %
2 (<i>Media</i>)	35	39,77 %
3 (<i>Difícil</i>)	5	5,68 %

4.1.2. Análisis exploratorio

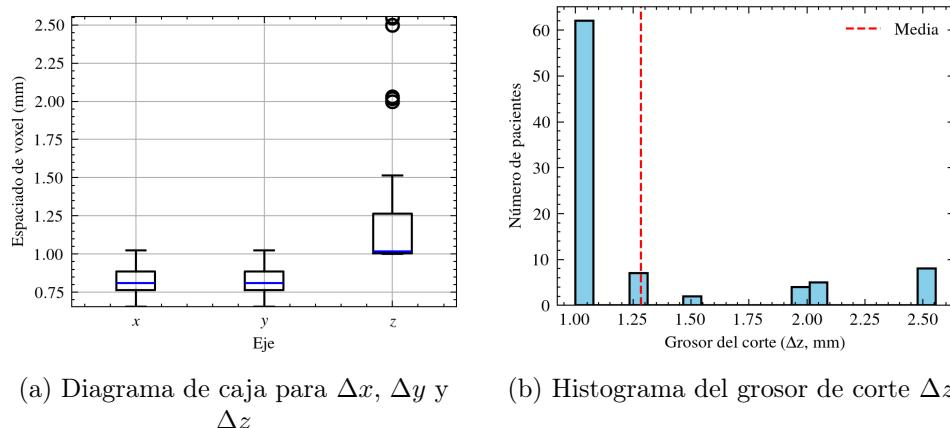
Espaciado de véxeles

En imágenes médicas volumétricas, cada unidad elemental tridimensional se denomina *vóxel* (de *volumetric pixel* en inglés). A diferencia de los píxeles en imágenes 2D, que tienen dimensiones fijas y uniformes, los véxeles pueden tener distintos tamaños en cada dirección. El **espaciado de véxel** (*voxel spacing*) define la distancia física real (en milímetros) entre véxeles adyacentes en los ejes x , y y z , lo que equivale a la resolución espacial del volumen en cada dimensión. El espaciado en z (grosor de corte) suele ser mayor y más variable que en los planos $x-y$, debido a las limitaciones del escáner y las decisiones clínicas al adquirir la imagen.

Esta información, almacenada en los encabezados del archivo NIfTI, permite interpretar correctamente las dimensiones físicas de las estructuras anatómicas representadas. En este trabajo se ha extraído el espaciado ($\Delta x, \Delta y, \Delta z$) de todos los casos del conjunto de entrenamiento (88 pacientes). Los resultados se resumen a continuación.

Cuadro 4.6: Estadísticas de spacing de véxeles (mm)

Eje	Media	Mediana	Desv. Típica	Mínimo	Máximo
x	0,8272	0,8096	0,0864	0,6535	1,0231
y	0,8272	0,8096	0,0864	0,6535	1,0231
z	1,2819	1,0175	0,4980	1,0000	2,5556



(a) Diagrama de caja para Δx , Δy y Δz (b) Histograma del grosor de corte Δz

Figura 4.1: Visualizaciones de la distribución del espaciado de véxeles.
Elaboración propia

De acuerdo con los resultados reflejados en el Cuadro 4.6 y apoyados por la Figura 4.1 demuestran que el espaciado en los ejes x e y es consistente. Por el contrario, en el grosor de corte se tienen entre 20–30 pacientes con un valor alejado del más frecuente.

Por tanto, será necesario considerar el espaciado, principalmente en el eje z , a la hora de preprocesar los datos para el entrenamiento.

Número de cortes por volumen

Para conocer la variabilidad en la cantidad de cortes por volumen (profundidad, D), se ha extraído el número de slices por paciente (D_i) junto con el espaciado en z (Δz_i) puesto que están directamente relacionados (el número de cortes depende del grosor de corte). En la Figura 4.2 se muestran los resultados obtenidos:

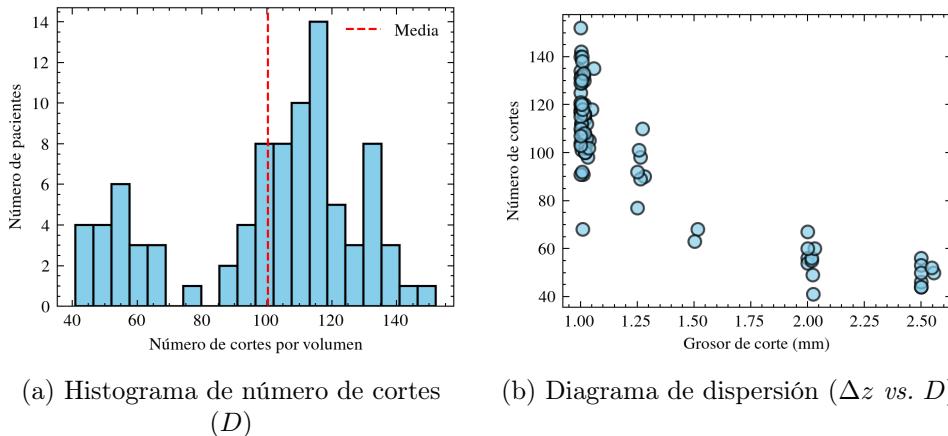


Figura 4.2: Distribución y dependencia del número de cortes por volumen.
Elaboración propia

De forma más estadística:

Cuadro 4.7: Estadísticas de número de cortes

Media	Mediana	Desv. Típica	Mínimo	Máximo
100,39	108,00	28,40	41,00	152,00

Como se puede observar en la Figura 4.2 y el Cuadro 4.7, y resulta obvio, a mayor grosor de corte menor número de cortes. Esto confirma que es necesario algún preprocesamiento con el objetivo de homogeneizar los datos.

Dimensiones de las máscaras segmentadas

Para analizar el desequilibrio entre clases y caracterizar el volumen relativo de cada estructura, se ha calculado el número de vóxeles etiquetados por clase en cada paciente del conjunto de entrenamiento. Aunque esto no representa el volumen físico real (en mm³), esta métrica permite realizar comparaciones proporcionales consistentes entre pacientes y clases, ya que no se ve afectada por la variabilidad del grosor de corte.

La Figura 4.3 muestra la proporción global de cada clase (páncreas, tumor, arterias y venas) sobre el total segmentado, excluyendo el fondo.

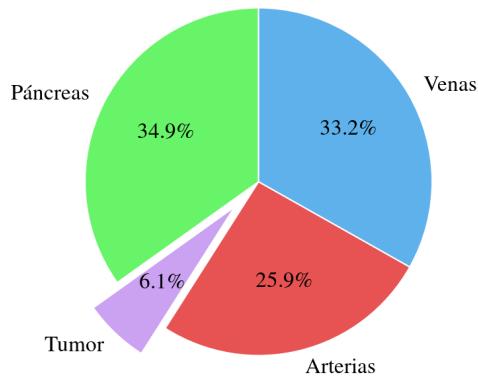


Figura 4.3: Proporción relativa de cada clase segmentada. *Elaboración propia*

La Figura 4.4 muestra la distribución de proporciones por clase por paciente. Destaca la alta variabilidad entre individuos, especialmente en el volumen relativo del páncreas.

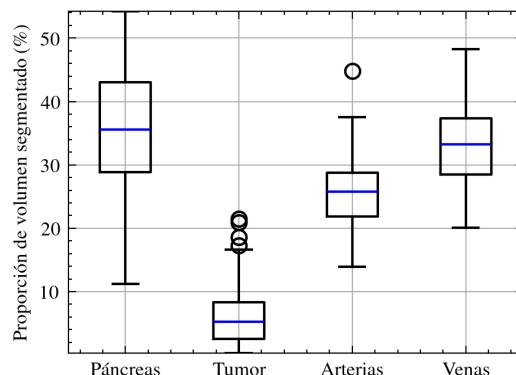


Figura 4.4: Distribución por paciente de la proporción de cada clase. *Elaboración propia*

Considerando ahora el volumen completo de la TC (incluido el fondo), las estructuras segmentadas suponen menos del 1 %. La Figura 4.5 resume esta relación.

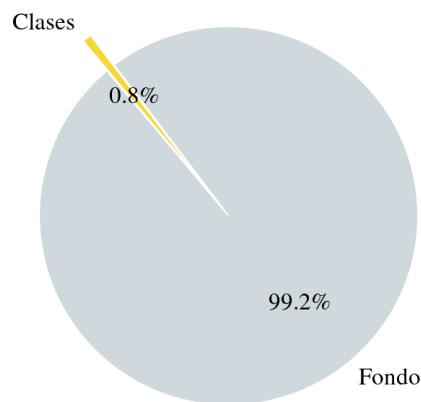


Figura 4.5: Proporción de clases sobre volumen total de TC. “Clases” representa la unión de páncreas, tumor, arterias y venas. *Elaboración propia*

La Tabla 4.8 detalla la proporción media por clase. El tumor pasa de constituir el 6 % de los véxoles segmentados (Figura 4.3) a apenas alcanzar el 0,05 % respecto del total del volumen de la tomografía computarizada.

Cuadro 4.8: Proporción de cada clase sobre el volumen total de TC

Clase o estructura	Proporción (%)
Páncreas	0,2793 %
Tumor	0,0506 %
Arterias	0,1987 %
Venas	0,2562 %

Este análisis confirma que el conjunto de datos presenta un fuerte desequilibrio de clases, tanto en proporción relativa como absoluta dentro del volumen total. La baja presencia del tumor, especialmente frente al fondo, constituye uno de los principales retos para la segmentación automática, y justifica la necesidad de aplicar técnicas específicas de balanceo durante el entrenamiento.

Distribución de intensidades (HU)

El análisis de la distribución de intensidades en unidades HU permite estudiar el contraste entre las distintas estructuras anatómicas segmentadas, aspecto clave en tareas de segmentación automática. Para ello, se han extraído los valores de intensidad en voxels etiquetados como páncreas, tumor, arterias, venas y fondo, y se han representado mediante histogramas normalizados de modo que el área total bajo la curva sea 1 (densidad de probabilidad por unidad de HU). Se han recortado las intensidades al rango $[-200, 400]$ HU, habitual en imágenes abdominales.

La Figura 4.6 incluye todas las clases y el fondo. En ella puede observarse un pico muy pronunciado a la izquierda, correspondiente a regiones de baja densidad como el aire (alrededor de -1000 HU), que ocupa la mayor parte del volumen total de la imagen.

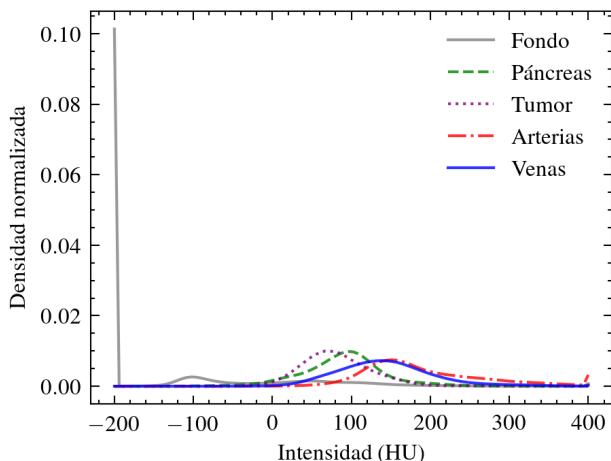


Figura 4.6: Distribución de intensidades por clase, incluyendo el fondo.

Elaboración propia

Este comportamiento justifica el recorte aplicado en el preprocesamiento a un rango de intensidades adecuado que permita centrarse en los tejidos relevantes para el diagnóstico.

Para facilitar el análisis comparativo entre clases anatómicas, la Figura 4.7 muestra la misma distribución sin incluir el fondo. En ella se aprecia un claro solapamiento entre las distribuciones de páncreas y tumor, ambos concentrados principalmente entre 20 y 160 HU, lo que anticipa la dificultad de diferenciarlos utilizando únicamente la intensidad. Las arterias y venas presentan distribuciones más desplazadas hacia intensidades altas, especialmente las arterias, como resultado del contraste intravenoso administrado en la fase arterial o portal venosa.

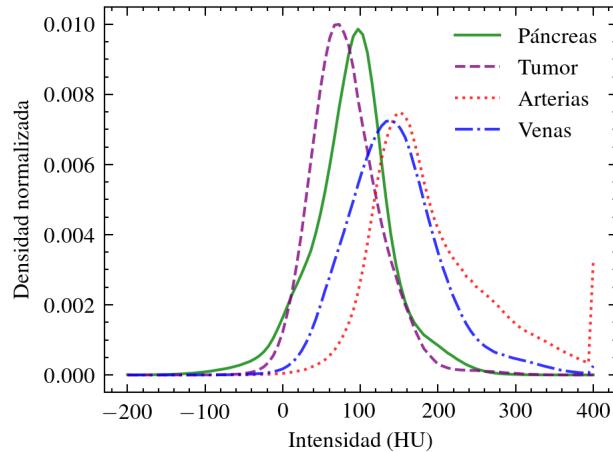


Figura 4.7: Distribución de intensidades por clase, excluyendo el fondo.

Elaboración propia

Este análisis revela un importante solapamiento entre las clases relevantes y una gran disparidad de representación del fondo respecto al resto, tal y como se veía en el apartado anterior. Estos factores refuerzan la necesidad de normalizar las intensidades, recortar los valores extremos y emplear modelos que no dependan exclusivamente de la información por voxel, sino también de su contexto espacial y anatómico.

4.1.3. Visualización

La visualización de los datos volumétricos y sus segmentaciones es fundamental para comprender el conjunto de datos, validar las anotaciones manuales y detectar posibles anomalías o sesgos antes del entrenamiento. En esta sección se presentan diferentes estrategias de visualización empleadas para inspeccionar los volúmenes de tomografía computarizada (TC) y sus respectivas máscaras de segmentación.

Visualización del volumen corte a corte

Una forma de visualizar un volumen 3D es mediante la secuenciación de las imágenes que lo conforman en uno de sus planos. En este caso, la Figura 4.8 las imágenes originales de la TC de un paciente en el plano axial, complementado con la Figura 4.9 que muestra las mismas imágenes junto con las segmentaciones superpuestas mediante contornos coloreados para las clases de interés: páncreas, tumor, arterias y venas. Esta visualización permite:

- Evaluar la calidad y precisión de las segmentaciones manuales.
- Observar la distribución espacial y el tamaño relativo de las estructuras en cada corte.
- Detectar posibles inconsistencias en las anotaciones.

El uso de contornos en lugar de máscaras opacas facilita la inspección simultánea de la intensidad original (HU) y la ubicación precisa de las regiones segmentadas. Se han tomado 12 slices que recorren el plano axial de manera equidistante, por lo que la evolución entre ellas es representativa del volumen.

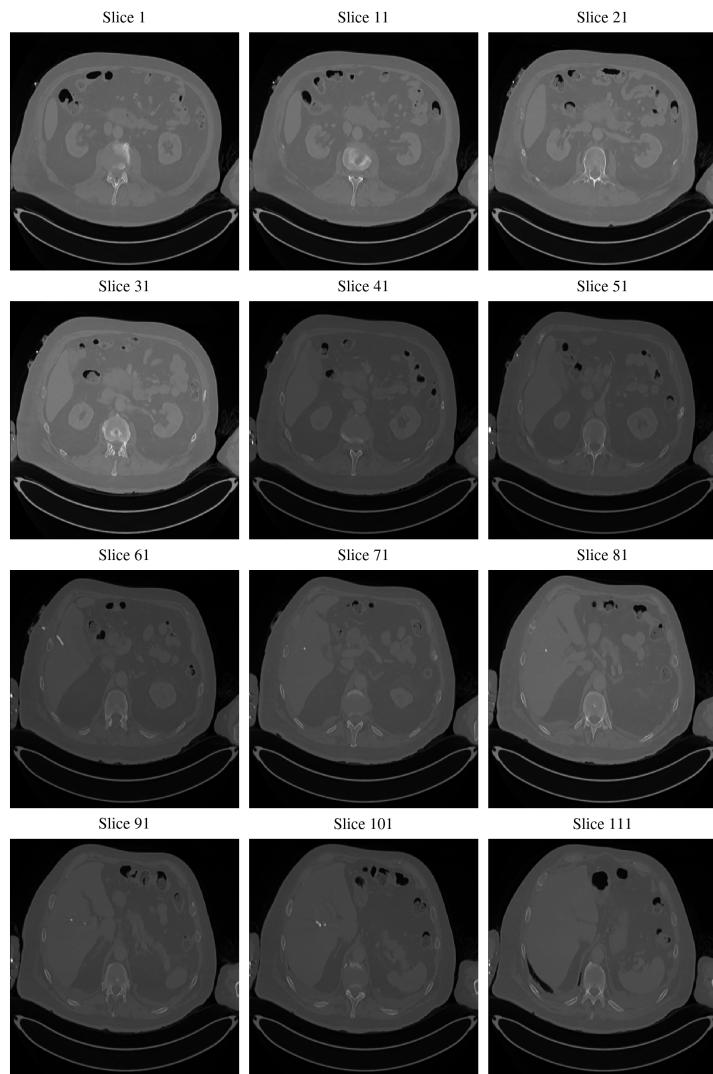


Figura 4.8: Visualización de TC corte a corte. *Elaboración propia*

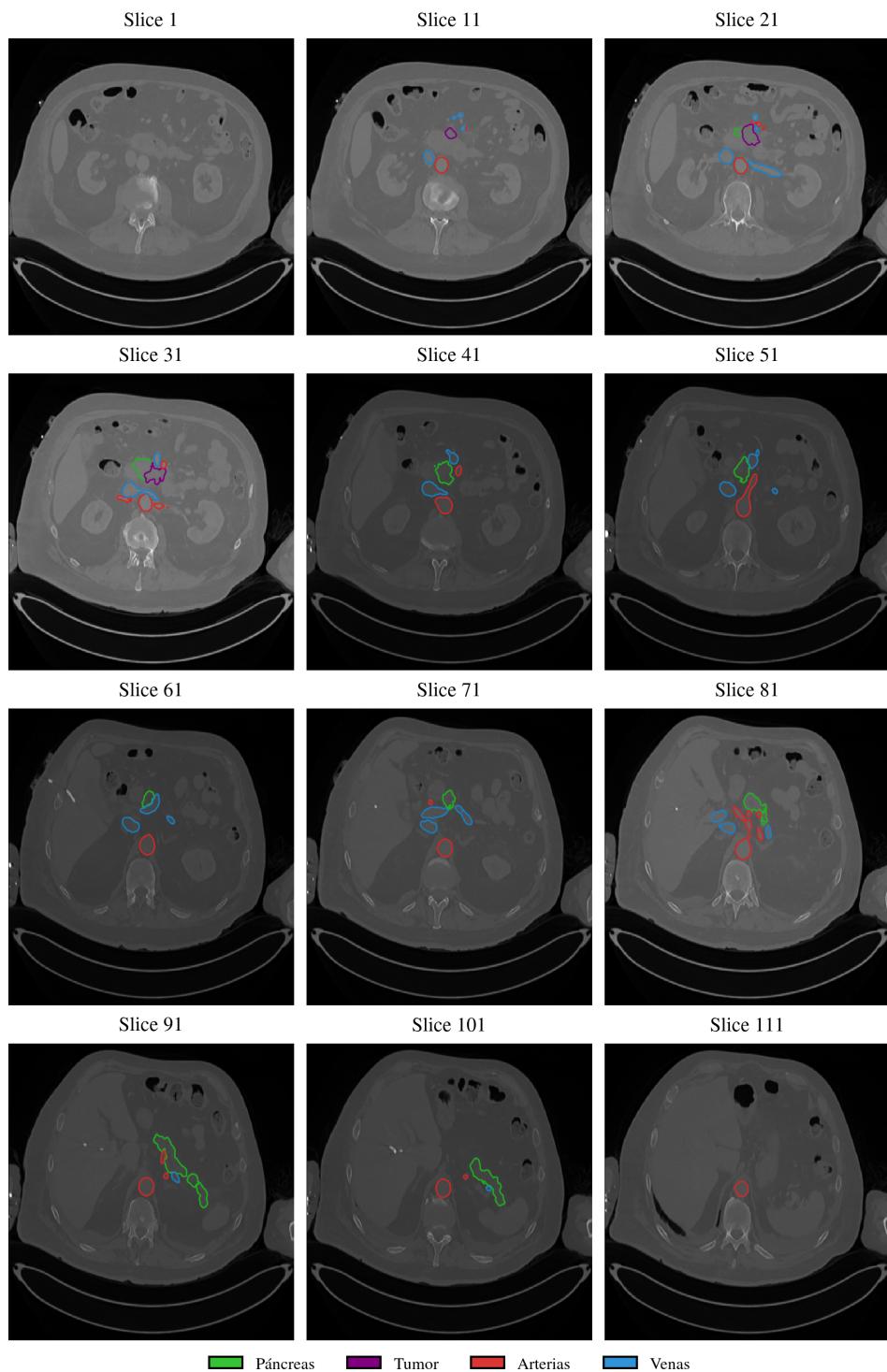


Figura 4.9: Visualización de TC y segmentación corte a corte. *Elaboración propia*

Visualización multiplanar

La visualización multiplanar, que incluye los planos axial, coronal y sagital, permite una comprensión tridimensional más intuitiva. Esto puede facilitar, por ejemplo, la identificación de posibles irregularidades en las segmentaciones que podrían pasar desapercibidas en una única vista y la comprensión de la coherencia anatómica entre diferentes planos.

A continuación se encuentran dos figuras, conteniendo la Figura 4.10 las imágenes originales y la Figura 4.11 las máscaras superpuestas usando los colores habituales:

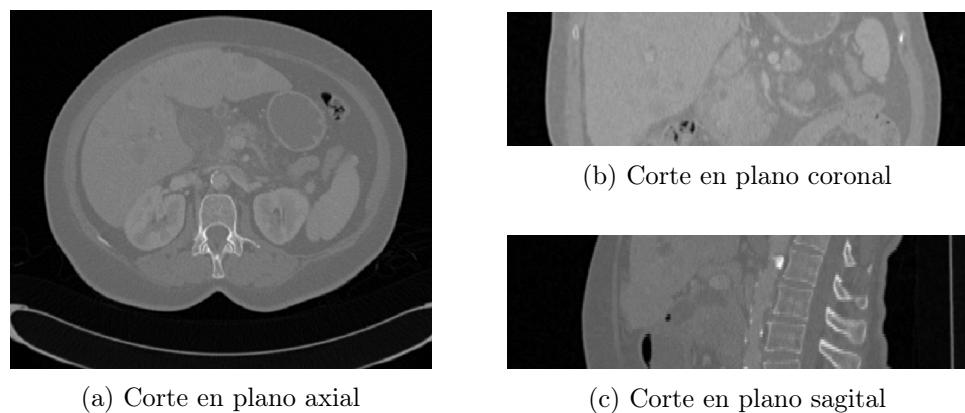


Figura 4.10: Visualización multiplanar de corte centrado en TC. *Elaboración propia*

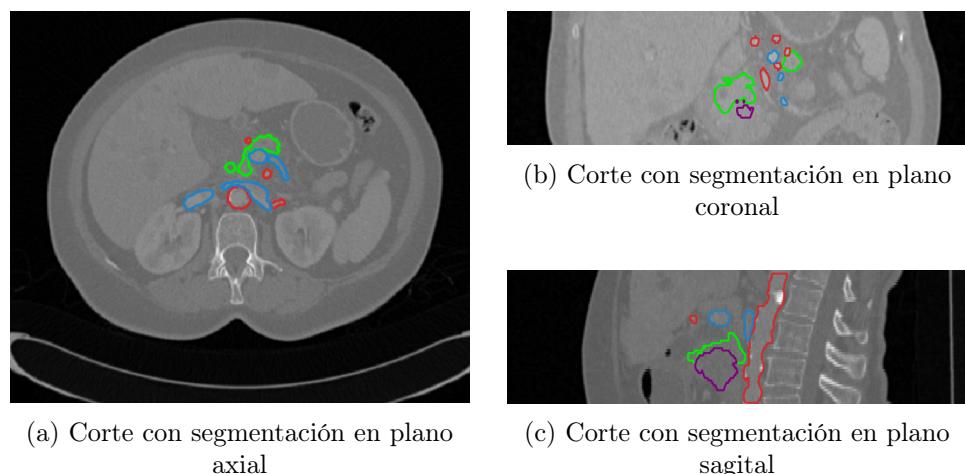


Figura 4.11: Visualización multiplanar de corte centrado en TC con contornos de máscaras. *Elaboración propia*

Intersección de planos volumétricos y segmentación 3D

Además de las vistas multiplanarias, se pueden visualizar las intersecciones de los planos para evaluar la continuidad espacial y la precisión tridimensional de las segmentaciones. En este apartado se muestran imágenes con y sin superposición de la segmentación en 3D, lo que aporta una visión global y detallada simultáneamente. Estas visualizaciones permiten obtener una percepción espacial de las estructuras y analizar las formas volumétricas de cada una de ellas. Los colores empleados son los habituales.

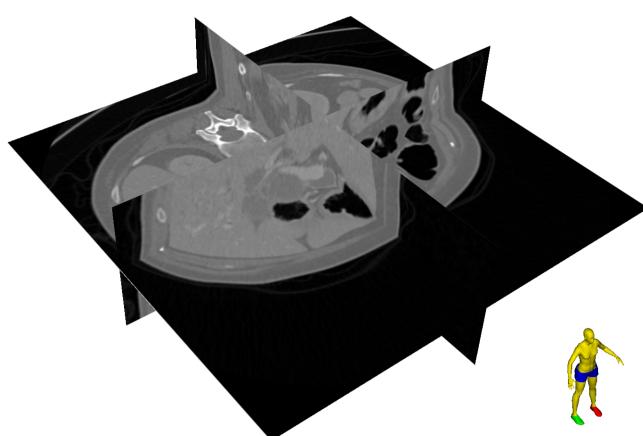


Figura 4.12: Intersección de planos en el centro de TC. La figura humana indica la orientación. *Elaboración propia* usando Slicer [6]

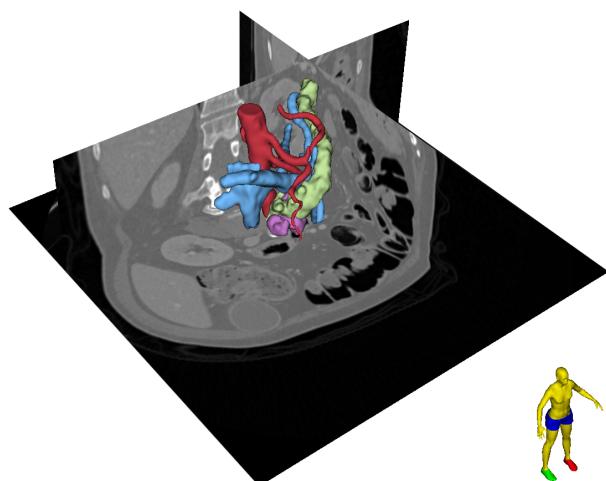


Figura 4.13: Intersección de planos junto con segmentación en 3D. La figura humana indica la orientación. *Elaboración propia* usando Slicer [6]

Ampliación en zonas de interés

Dado que las estructuras patológicas (tumor) suelen ocupar regiones pequeñas y delicadas, como se ha demostrado en secciones previas, en este apartado se presenta un caso en el que se aplica zoom y recorte en el área de interés de una imagen axial. La región de interés se ha determinado como aquel espacio que contiene la segmentación en su totalidad, es decir, que contiene todas las máscaras.

La siguiente figura muestra la imagen original junto con el zoom aplicado:

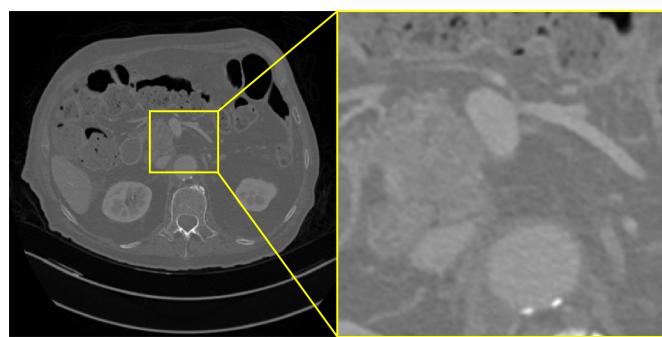
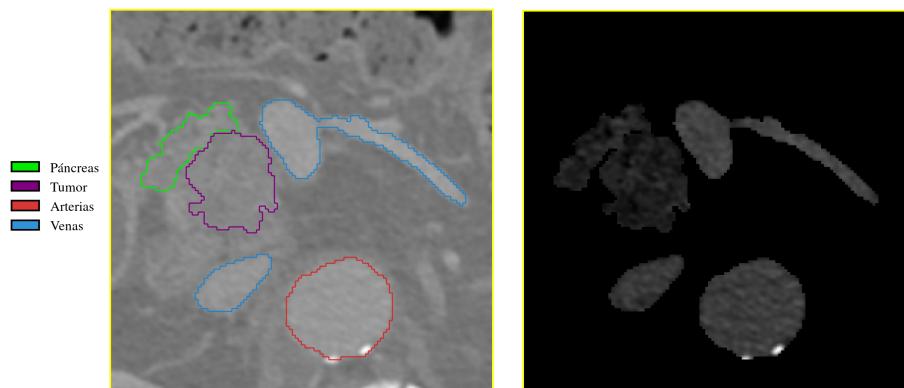


Figura 4.14: Zoom en región de interés. *Elaboración propia*

Sobre la región ampliada se muestran las estructuras delimitadas con un contorno (Figura 4.15a) y las estructuras recortadas, esto es, eliminando el fondo (Figura 4.15b). Esta se ve más oscura debido a que el rango de la escala de grises se ha adaptado a las intensidades (HU) de las estructuras.



(a) Región ampliada con contornos

(b) Región ampliada con fondo eliminado

Figura 4.15: Identificación de estructuras en zona ampliada. *Elaboración propia*

Visualización tridimensional de segmentaciones

Las visualizaciones volumétricas de este apartado tienen la finalidad de permitir observar la morfología completa del tumor y órganos relacionados, y comprender la relación espacial entre la lesión y las estructuras vasculares.

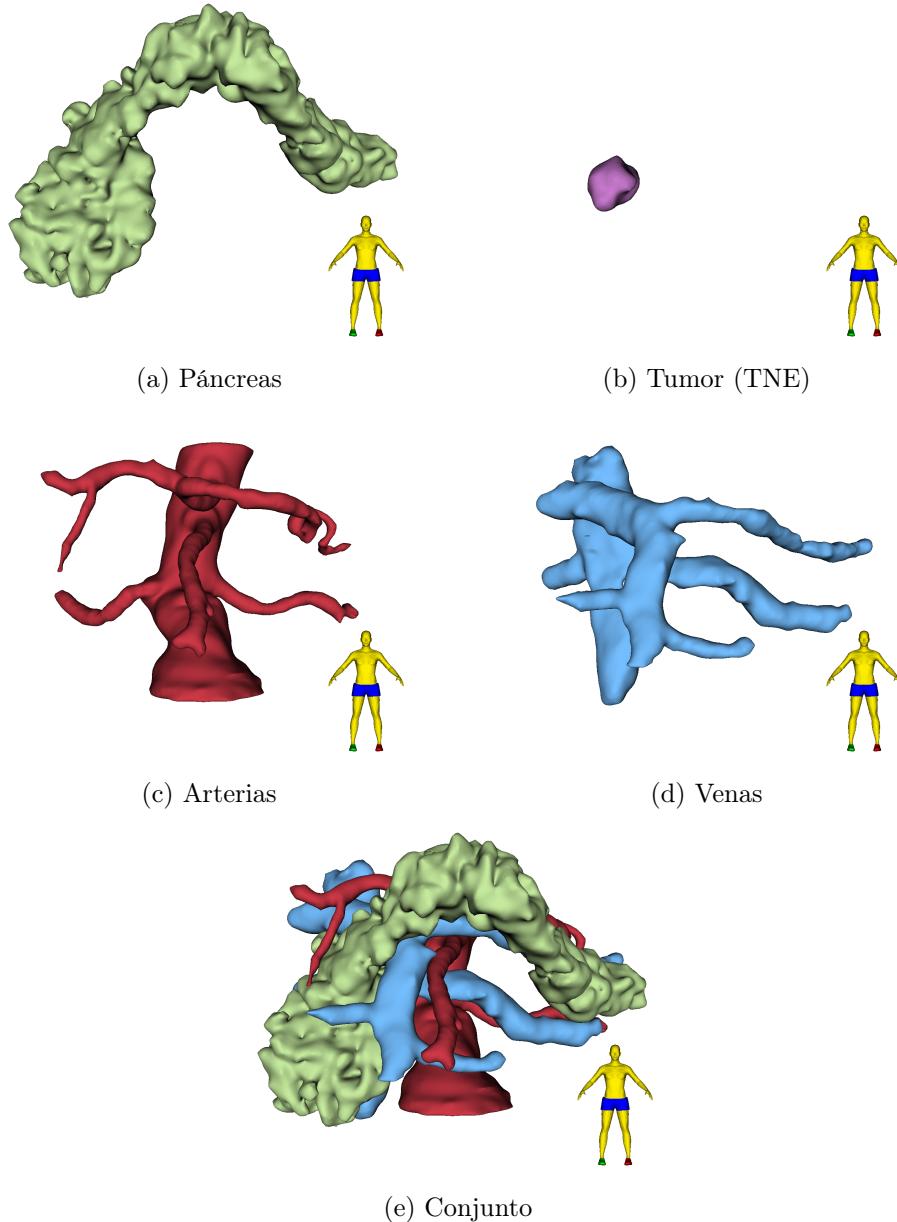


Figura 4.16: Reconstrucción 3D de segmentaciones de paciente con tumor neuroendocrino (TNE). La figura humana indica la orientación.
Elaboración propia usando Slicer [6]

Además de la reconstrucción de la Figura 4.16, cuyo tumor es neuroendocrino, y siguiendo con el objetivo del proyecto de facilitar la clasificación por resecabilidad del tumor, la siguiente figura muestra dos casos de adenocarcinoma ductal pancreático, (PDAC), uno clasificado como resecable y otro como resecable borderline.

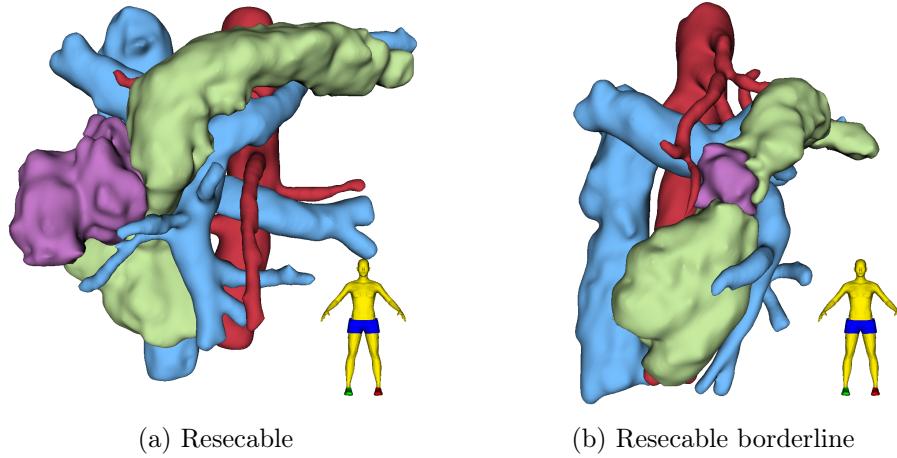


Figura 4.17: Visualización 3D de tumor resecable y resecable borderline. La figura humana indica la orientación. *Elaboración propia* usando Slicer [6]

Para visualizar mejor la relación entre la lesión y las estructuras vasculares, la Figura 4.18 muestra los mismos modelos 3D excluyendo la segmentación del páncreas.

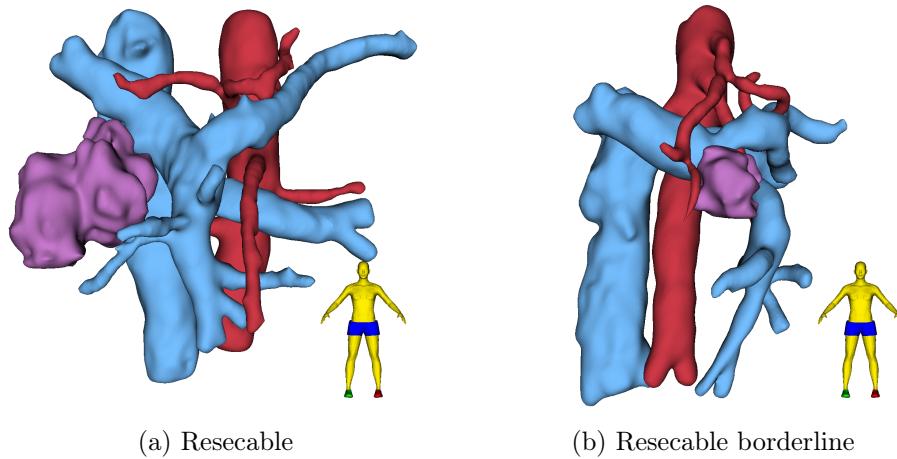


Figura 4.18: Visualización 3D sin páncreas de tumor resecable y resecable borderline. La figura humana indica la orientación. *Elaboración propia* usando Slicer [6]

A continuación se muestran dos vistas más de cada caso: una en el plano axial (inferior a superior, equivalente a ver al paciente desde arriba) y otra adaptada para reflejar mejor la diferencia.

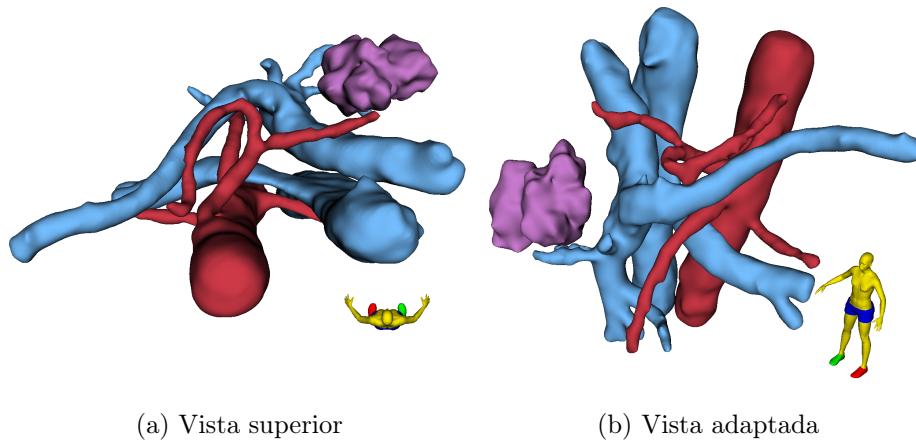


Figura 4.19: Visualización 3D de tumor resecable en distintas perspectivas. La figura humana indica la orientación. *Elaboración propia* usando Slicer [6]

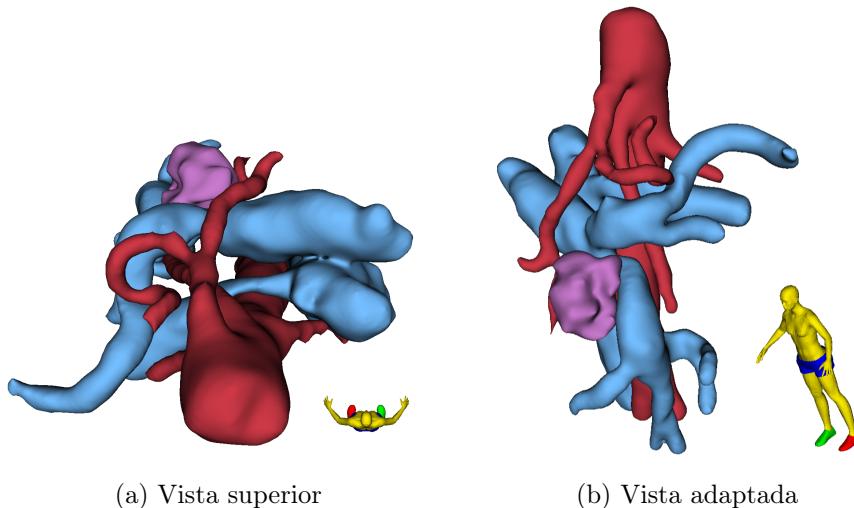


Figura 4.20: Visualización 3D de tumor resecable borderline en distintas perspectivas. La figura humana indica la orientación. *Elaboración propia* usando Slicer [6]

Viendo ambas figuras se comprende mejor la importancia de la relación entre las estructuras para el tratamiento de esta enfermedad. Un tumor que esté alejado de las venas y arterias será fácilmente resecable (Figura 4.19).

Sin embargo, si está muy próximo y contactando con los vasos específicos (Figura 4.20), dificulta el tratamiento quirúrgico puesto que se compromete la seguridad de la resección.

4.2. Técnicas de preprocesamiento de datos

El preprocesamiento de los datos es una etapa crítica en la construcción de modelos de aprendizaje profundo para imágenes médicas, ya que garantiza la homogeneidad de los datos de entrada, mejora la calidad de los mismos y facilita la generalización del modelo. En este trabajo, se han seguido dos enfoques principales para el preprocesamiento, que evolucionaron en función del grado de automatización y la integración con librerías especializadas.

- **Preprocesamiento manual:** esta estrategia fue la utilizada inicialmente, en la que todas las operaciones se realizaban de forma explícita fuera del entrenamiento. Incluye transformaciones estáticas (como el reescalado del espaciado de las imágenes, reorientación, recorte, almacenamiento por slices o subvolúmenes, etc.), así como transformaciones aplicadas dinámicamente durante el entrenamiento (aplicación de ventana y normalización de intensidad, y técnicas de data augmentation específicas tanto para 2D como para 3D).
- **Preprocesamiento automatizado con MONAI:** posteriormente, se optó por un enfoque más estructurado y reproducible utilizando la librería MONAI. Este enfoque facilita una integración directa con el proceso de entrenamiento, reduce errores manuales y permite la aplicación de técnicas avanzadas de aumento de datos.

A continuación, se detallan los pasos específicos llevados a cabo en cada uno de los enfoques mencionados, incluyendo tanto las transformaciones estáticas previas al entrenamiento como las dinámicas aplicadas durante el mismo.

4.2.1. Preprocesamiento manual

Este enfoque se utilizó en fases iniciales del proyecto, cuando el pipeline aún no estaba completamente automatizado. Las operaciones se dividieron en dos categorías: transformaciones estáticas, realizadas antes del entrenamiento, y transformaciones dinámicas, aplicadas durante el entrenamiento.

Transformaciones estáticas (previas al entrenamiento)

- Lectura y combinación de máscaras: se cargan las máscaras correspondientes a las diferentes clases y se combinan en una única máscara multiclas.
- Reescalado del spacing: se ajusta el espaciado a (0.8, 0.8, 1.0) para homogeneizar la resolución espacial entre distintos volúmenes. Estos valores se basan en las estadísticas obtenidas en el análisis de datos y expuestas en el Cuadro 4.6.
- Reorientación del volumen: se fuerza una misma orientación (RPS en este caso) y se rotan los volúmenes para que coincidan visualmente con Slicer [6], lo que facilita la visualización y la corrección de posibles errores o la interpretación de resultados.
- Recorte al ROI: se realiza un recorte a una región de interés (ROI) centrada en las estructuras a segmentar, con tamaño fijo de 256×256 píxeles por slice. Antes del ajuste de espaciado se utilizaba otro ROI que comprometía parte del contenido de la imagen.
- Almacenamiento de datos:
 - 2D: se almacenan los slices individuales del volumen en formato .npy (NumPy) y las máscaras en formato .png.
 - 3D: se generan subvolúmenes con solapamiento (por ejemplo, tamaño 64 y desplazamiento o *stride* 32), almacenándolos como .npy.

Transformaciones dinámicas (durante el entrenamiento)

- Aplicación de ventana: se aplica una ventana de nivel 50 y ancho 400, correspondiente a una escala de intensidades de $[-150, 250]$, centrada en el rango habitual de tejidos blandos. El principal motivo por el que se escoge este rango y no otro es la recomendación directa del personal clínico del hospital. Además, se ajusta a la distribución obtenida en el análisis (véase la Figura 4.6 y Figura 4.7), y se aproxima a ventanas empleadas en el estado del arte para segmentar páncreas y tumor como en [32, 33], o coincide con otros estudios como [55].
- Normalización de intensidades: las intensidades (HU) se normalizan para ajustarse al rango $[0, 1]$.
- Data augmentation:

- En entrenamientos 2D: se utilizan técnicas de Albumentations, incluyendo transformaciones como `Affine`, `ElasticTransform`, `GaussianBlur`, `RandomBrightnessContrast` y `GridDistortion`.
- En entrenamientos 3D: se emplea TorchIO con transformaciones como `RandomAffine`, `RandomFlip`, `RandomNoise` y `RandomBlur`.

4.2.2. Preprocesamiento con MONAI

Con el objetivo de no solo estructurar y reproducir fácilmente el preprocesamiento, sino también incorporar transformaciones más avanzadas y útiles en una tarea tan compleja, posteriormente se adaptó el uso de la librería MONAI.

Preparación previa al entrenamiento

- Combinación de máscaras: a partir de ficheros `.nii` independientes por clase, tal y como se describía la estructura del dataset en la Sección 4.1, se combinan en una única máscara multiclase y se almacenan como `.nii.gz`, al igual que los volúmenes. Aprovechar la posibilidad de comprimir, almacenando el volumen y las máscaras como `.nii.gz`, no supone una limitación para MONAI ni para la mayoría de lectores de archivos NIfTI como puede ser NiBabel, pero sí que supone un gran beneficio a nivel de ocupación de disco. La diferencia de tamaño del dataset completo se resume en el siguiente cuadro:

Cuadro 4.9: Diferencia de tamaño entre dataset original y preparado

Partición	Casos	Tamaño original (GB)	Tamaño preparado (GB)
Train	88	27,20 GB	2,48 GB
Test	28	7,58 GB	0,84 GB

Preprocesamiento

El preprocesamiento para MONAI tiene algunas diferencias respecto del desarrollado en el apartado anterior, principalmente porque deriva del aprendizaje durante la fase de uso de implementaciones propias, fase que permitió comprender la necesidad de incorporar técnicas más avanzadas y específicas, tanto de preprocesamiento como de aumento de datos, que mejoren la generalización de los modelos y prevengan el sobreajuste debido a la escasa cantidad de datos disponibles.

Durante el pipeline de transformaciones determinísticas, primero se garantiza que las imágenes y máscaras tengan una misma orientación, en este

caso RAS que es una orientación estándar. A continuación, se homogeneiza el espaciado de los véxeles a una resolución isotrópica de 1 mm en cada eje, esto es, $(1 \times 1 \times 1)$ mm. El uso de un espaciado isotrópico (uniforme) frente a uno anisotrópico (no uniforme, como el del apartado anterior) está justificado en diversos estudios como [43, 50, 56, 57, 22] entre otros, afirmando que mejora el rendimiento y la generalización de los modelos, y es más coherente con las segmentaciones o ground truth y el análisis. Se especifica interpolación bilineal para los volúmenes y de vecino más cercano (*nearest neighbor*) para las máscaras, garantizando una alineación espacial coherente entre las distintas muestras sin perjudicar la calidad de la segmentación por medio de artefactos introducidos por la interpolación.

Posteriormente, se aplica la ventana de intensidades. Aquí no hay diferencia, el rango escogido es $[-150, 250]$, esto es, ventana de nivel 50 y ancho 400. Se ha planteado el uso del rango empleado por MONAI en su modelo del MONAI Model Zoo *Pancreas CT DiNTS segmentation* [38], $[-87, 199]$, pero al tratarse exclusivamente de páncreas y tumor se ha decidido mantener la ventana sugerida por el equipo clínico del hospital, apoyada en numerosos estudios como se ha indicado en el apartado anterior.

Una parte clave del preprocesamiento consiste en copiar la máscara original y aplicar una dilatación morfológica de cinco iteraciones mediante la función `binary_dilation` de `ndimage`. Extender ligeramente las estructuras asegura que los parches tomados para entrenar tengan suficiente contexto anatómico. Esta versión dilatada de la máscara se utiliza únicamente como guía para recortar regiones de interés más amplias alrededor de las clases objetivo. El resultado de esta operación se puede ver en la siguiente figura:

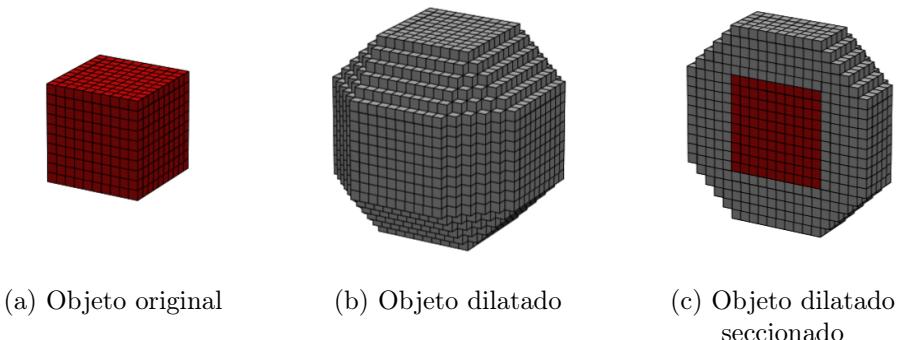


Figura 4.21: Efecto de dilatación binaria con 5 iteraciones. *Elaboración propia*

Debido a la variabilidad observada en el número de cortes (D) de los volúmenes, se aplica un *padding* espacial para garantizar que todos ellos cubran como mínimo un parche de entrenamiento, es decir, unas dimensiones

mínimas de $96 \times 96 \times 96$ por muestra, utilizando modos de relleno distintos: reflexión (*reflect*) para el volumen y constante para las máscaras.

Finalmente, antes de ser movidos a la GPU, los datos se convierten a tensores y se desactivan los metadatos de MONAI, innecesarios para el entrenamiento.

En cuanto a las transformaciones aleatorias para aumento de datos, se emplea primeramente una estrategia de muestreo mediante la función `RandCropByLabelClassesd`, que extrae aleatoriamente dos parches de $96 \times 96 \times 96$ por muestra a partir de la máscara dilatada. De forma simple, la función escoge una clase aleatoria y, para esa clase, escoge un véxel aleatorio. Este véxel aleatorio será el centro del parche. Los ratios de selección de cada clase se establecen a 1 para fondo, 2 para páncreas, arterias y venas, y 3 para tumor. De este modo, se prioriza el tumor sobre el resto por ser la clase más compleja y menos representada, y se priorizan páncreas, arterias y venas sobre el fondo. Sin embargo, aquí es donde se refleja la utilidad de la máscara dilatada, pues al ser esta la guía que tiene la función para escoger los centros, se consigue principalmente ampliar el área donde el muestreo de recortes es válido (hay más representación “virtual”) y asegurar que los recortes no solo capturen una clase, sino su entorno anatómico. Por ejemplo, puede darse el caso que se escoja tumor y el véxel que tome como centro realmente pertenezca al páncreas, alguna estructura vascular o incluso fondo.

La combinación de `RandCropByLabelClassesd` con máscaras dilatadas como guías está basada en el MONAI Model Zoo *Pancreas CT DiNTS segmentation* [38] y en algunos estudios recientes como [40], donde lo definen como *label dilation* (“dilatación de etiquetas”).

El resto de técnicas de data augmentation incluidas son:

- Deformaciones elásticas tridimensionales que simulan variaciones anatómicas suaves, útil para mejorar la capacidad del modelo de adaptarse a diferencias morfológicas entre pacientes.
- Rotaciones aleatorias en torno a uno o más ejes, aportando invarianza rotacional.
- Zooks aleatorios: escalados aleatorios del volumen, ampliando o reduciendo ligeramente el campo visual, lo que ayuda al modelo a aprender representaciones robustas a distintos niveles de zoom.
- Suavizado gaussiano aleatorio que permite simular distintos grados de desenfoque, haciendo que el modelo sea menos sensible a pequeñas diferencias de resolución o ruido.
- Campos de bias aleatorios que simulan la presencia de campos de bias, un tipo de artefacto común en imágenes médicas. Aunque aparecen

sobre todo en resonancias magnéticas, se ha considerado su inclusión en este problema.

- Escalados aleatorios de las intensidades del volumen, emulando diferencias en el contraste entre imágenes.
- Desplazamientos aleatorios de las intensidades, simulando cambios sistemáticos en el nivel de gris de las imágenes.
- Ruido gaussiano aleatorio, lo que permite entrenar modelos más robustos al ruido inherente de las propias imágenes de TC.
- *Flips* o volteos aleatorios en los tres ejes espaciales, lo que introduce simetría y aumenta la variabilidad geométrica del conjunto de entrenamiento.

Como se ha expuesto, el preprocesamiento y las técnicas de aumento de datos implementadas con MONAI constituyen un enfoque más sólido y estructurado que el aplicado manualmente en las primeras fases del desarrollo. Esto ha sido posible gracias a la experiencia adquirida inicialmente, que permitió construir un pipeline más ajustado a las necesidades del problema.

Es importante destacar que cada una de las transformaciones incluidas ha sido seleccionada cuidadosamente en función tanto de su impacto en la variabilidad de las muestras como en su coherencia con la estructura tridimensional y la lógica anatómica inherente al problema. Del mismo modo, los parámetros específicos asociados a estas transformaciones, como rangos de aplicación, escalas o probabilidades, han sido definidos con atención para evitar generar muestras artificiales que no sean realistas o directamente anatómicamente incorrectas.

Conviene señalar que, aunque algunas transformaciones podrían parecer prescindibles en un primer análisis, han demostrado tener una utilidad significativa en la práctica. Un ejemplo claro es el caso de los flips o volteos aleatorios en los tres ejes espaciales. Dado que las imágenes de TC se reorientan a una orientación común, podría asumirse que estas transformaciones no aportarían variabilidad relevante, ya que, en teoría, la disposición espacial de los órganos se mantiene constante. Sin embargo, se ha observado que los flips introducen simetrías adicionales que ayudan a evitar el sobreajuste, dando lugar a una generalización más robusta. Estas simetrías artificiales pueden hacer que el modelo se enfoque menos en patrones posicionales específicos y más en las características morfológicas y de textura.

Ocurre lo mismo con las deformaciones elásticas o alteraciones de intensidades pues, si bien requieren un ajuste cuidadoso, pueden aumentar la diversidad sin comprometer la coherencia anatómica. Su utilidad reside precisamente en su capacidad para simular condiciones realistas pero variadas, contribuyendo también a la generalización del modelo.

Capítulo 5

Deep Learning

5.1. Fundamentos teóricos

El análisis avanzado de imágenes médicas para la detección de enfermedades como el cáncer de páncreas se ha visto revolucionado por la aplicación del Deep Learning (DL). Esta sección proporciona los fundamentos teóricos necesarios para comprender las técnicas empleadas en la segmentación automática de estructuras anatómicas y patológicas en imágenes médicas.

Redes neuronales artificiales

Las redes neuronales artificiales (ANN, por sus iniciales en inglés) son un componente fundamental del Deep Learning y fueron las primeras implementaciones en este campo. Se basan en el concepto de *feed-forward* (alimentación hacia adelante), donde la información se mueve en una única dirección, desde los nodos de entrada hacia las unidades de salida. Una ANN está compuesta por capas de neuronas (nodos) interconectadas: una capa de entrada, una o varias capas ocultas, y una capa de salida [58].

El entrenamiento de una red neuronal implica el ajuste de sus parámetros, es decir, pesos (*weights*) y sesgos (*bias*), para minimizar una función de pérdida que mide la diferencia entre las predicciones del modelo y las etiquetas de verdad fundamental (ground truth), normalmente mediante algoritmos de optimización basados en el gradiente. La retropropagación (*backpropagation*) es el algoritmo clave que permite calcular eficientemente los gradientes de la función de pérdida con respecto a los pesos de la red, propagándolos desde la salida hacia las primeras capas ocultas. Además, cada neurona aplica una función de activación a la suma ponderada de sus entradas, limitando o modificando el valor obtenido, introduciendo no linealidad y permitiendo a la red aprender patrones complejos.

Redes neuronales convolucionales

Las redes neuronales convolucionales (CNN) son una clase especializada de redes neuronales profundas, excepcionalmente eficaces para procesar imágenes. Su éxito reside en su capacidad para capturar dependencias espaciales y temporales mediante la aplicación de filtros relevantes. La arquitectura de una CNN le permite aprender de forma automática y adaptativa jerarquías espaciales de características, desde patrones de bajo nivel hasta patrones de alto nivel.

Los componentes clave de una CNN incluyen:

- Capas convolucionales: aplican filtros (*kernels*) a las imágenes de entrada para detectar características como bordes, líneas y esquinas. El resultado es un mapa de características que resalta la presencia de estas características en diferentes ubicaciones.

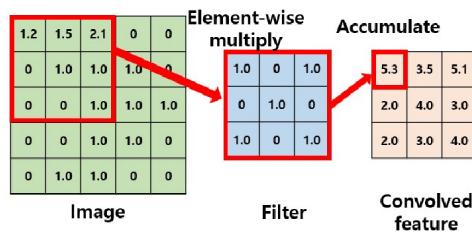


Figura 5.1: Operación de convolución. El resultado es la suma de la multiplicación elemento a elemento del filtro con la imagen. *Fuente: [9]*

- Funciones de activación: después de cada operación convolucional, se aplica una función de activación no lineal. Las más comunes incluyen la ReLU (por sus siglas en inglés, *Rectified Linear Unit*), la ReLU con fugas (Leaky ReLU) y las Unidades Lineales Exponenciales (ELU).

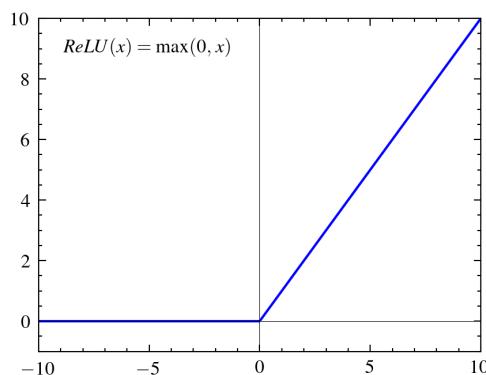


Figura 5.2: Función de activación ReLU con definición matemática.
Elaboración propia

- Capas de *pooling* (submuestreo): reducen la dimensionalidad espacial de los mapas de características (imágenes de entrada con filtros aplicados), disminuyendo la cantidad de parámetros y la complejidad computacional, ayudando a la red a ser más robusta a pequeñas variaciones en la ubicación de las características. El *max-pooling* es una operación común que selecciona el valor máximo dentro de una ventana deslizante, de modo que recorre la imagen o el mapa de características y reduce su dimensión tomando el máximo de la ventana. Con *average pooling* se toma la media de la ventana en lugar del máximo.

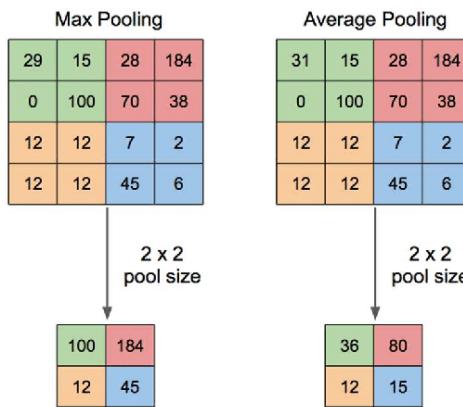


Figura 5.3: Comparación entre max pooling y average pooling. *Fuente:* [10]

- Capas completamente conectadas (*fully connected*): al final de la red, las capas convolucionales y de pooling suelen ser seguidas por una o más capas densamente conectadas que realizan la clasificación final o la regresión. Sin embargo, en la segmentación semántica, a menudo se reemplazan por capas convolucionales adicionales para producir un mapa de segmentación a nivel de píxel.

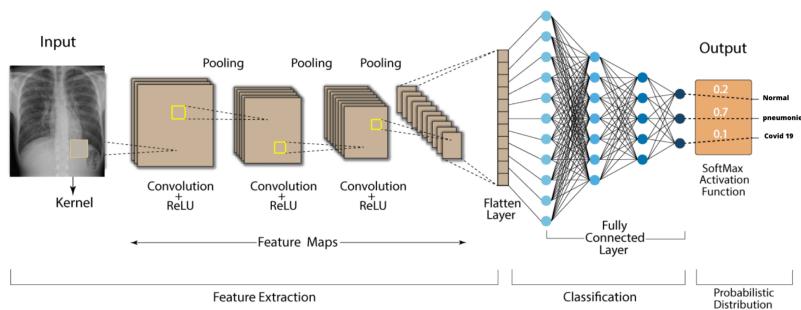


Figura 5.4: Red neuronal convolucional con capa fully connected para clasificar imágenes médicas según enfermedades respiratorias. *Adaptado de* [11]

Desafíos comunes en segmentación de imágenes médicas

Las imágenes médicas presentan desafíos únicos para las redes neuronales convolucionales que las distinguen de las aplicaciones de visión por computador en otros dominios. Además de los ya mencionados y expuestos como escasez de datos (véase la Sección 1.2) y desequilibrio de clases (véase la Figura 4.3 y la Figura 4.5), también destacan:

- Alta resolución de entrada: las imágenes de TC suelen tener una alta resolución, como las empleadas en este problema (512×512 píxeles por corte y decenas o centenas de cortes por volumen). Esto genera grandes volúmenes de datos (vóxeles) que pueden exceder la capacidad de memoria de las GPUs y aumentar significativamente el costo computacional para las redes 3D. Para manejar esto, se utilizan enfoques como el procesamiento por subvolúmenes (*patches*) o el reescalado a resoluciones más bajas, aunque esto último puede comprometer la información contextual.
- Coherencia espacial (3D): las imágenes médicas son inherentemente tridimensionales, esto es, volumétricas. Los modelos 2D, que procesan las imágenes corte por corte, son computacionalmente eficientes pero pierden información del contexto espacial intercorte, lo que puede llevar a segmentaciones discontinuas. Los modelos 3D abordan esto utilizando el volumen completo como entrada, capturando así la información espacial tridimensional, pero son computacionalmente mucho más costosos.

Funciones de pérdida

Las funciones de pérdida son cruciales para guiar el proceso de entrenamiento de la red. En segmentación, donde el objetivo es clasificar cada píxel (o vóxel), se utilizan pérdidas específicas, entre las que destacan:

- *Dice Loss*. Basada en el coeficiente de similitud Dice (DSC), una métrica común para evaluar la superposición entre dos conjuntos, que se detallará en capítulos futuros. La Dice Loss es ampliamente utilizada en segmentación médica debido a su robustez ante el desequilibrio de clases. Sin embargo, se ha observado que puede generar predicciones sobreconfiadas o mal calibradas [59]. Algunas variantes, como la *Generalized Dice Loss (GDL)*, buscan mitigar este problema aún más. Se define como:

$$\text{Dice Loss} = 1 - \frac{2 \cdot |\text{pred} \cap \text{gt}|}{|\text{pred}| + |\text{gt}|} \quad (5.1)$$

donde *pred* es la predicción y *gt* la ground truth [60].

- *Cross-Entropy* o Entropía cruzada. Es una función de pérdida estándar para problemas de clasificación. Mide la diferencia entre la distribución de probabilidad predicha por el modelo y la distribución de probabilidad real (ground truth). Aunque es efectiva, puede ser sensible al desequilibrio de clases en segmentación, ya que las clases mayoritarias (fondo) dominarán el gradiente de la pérdida. Se calcula como:

$$CE(p_t) = -\log(p_t) \quad (5.2)$$

donde p_t es la probabilidad estimada por el modelo para la clase verdadera. Si la etiqueta es 1 (clase positiva), $p_t = p$, con p la probabilidad predicha de la clase. Si la etiqueta es 0 (clase negativa), $p_t = 1 - p$. Se ha utilizado esta notación para compararla con *Focal Loss*.

- *Focal Loss*. Presentada en [61], se diseñó para resolver el problema del desbalance de clases, especialmente en detección de objetos. Se define como una modificación de la entropía cruzada en la que se reduce el peso de las muestras que son fáciles de clasificar correctamente (las que tienen alta probabilidad predicha), y enfocar el aprendizaje en las muestras difíciles (las que el modelo clasifica mal). Así, si una muestra es fácil, contribuirá menos a la pérdida, mientras que una muestra difícil contribuirá más. La función posee un parámetro clave, γ , que es el parámetro de enfoque. Cuanto mayor sea γ , mayor será la penalización por errores difíciles. Matemáticamente se define como:

$$FL(p_t) = -\alpha_t(1 - p_t)^\gamma \log(p_t) \quad (5.3)$$

donde α_t un factor opcional de peso para ponderar cada clase [60]. Como se puede apreciar, si $\gamma = 0$, entonces se cumple que $FL = CE$ (ignorando el parámetro α_t , que es opcional y también se puede incluir en CE).

- Combinaciones. Es una práctica común combinar múltiples funciones de pérdida para aprovechar sus beneficios complementarios. Por ejemplo, se puede usar una combinación ponderada de Dice Loss y Cross-Entropy o Dice Loss y Focal Loss.

Optimizadores y ajuste de la tasa de aprendizaje

Antes se mencionaba que el entrenamiento de una red neuronal conlleva ajustar sus parámetros para minimizar la función de pérdida. Los algoritmos fundamentales encargados de ello son los optimizadores. Son cruciales ya que aseguran que la red mejore gradualmente su precisión al reducir iterativamente los errores en las predicciones, por lo que la elección del optimizador puede influir significativamente en el rendimiento del modelo y en su tasa de convergencia [58]. Entre ellos se incluyen:

- Descenso de Gradiente (*Gradient Descent*). Es el método base para ajustar los parámetros de la red. Se basa en una propiedad fundamental del cálculo: el gradiente de una función diferenciable en un punto señala la dirección de máxima pendiente ascendente. Por lo tanto, para minimizar una función como la de pérdida, se avanza en la dirección opuesta al gradiente, es decir, en la de mayor descenso. Si el paso (determinado por la tasa de aprendizaje) es lo suficientemente pequeño, se garantiza una disminución del valor de la función en cada iteración, lo que aproxima al modelo a un mínimo local o global de la función de pérdida.

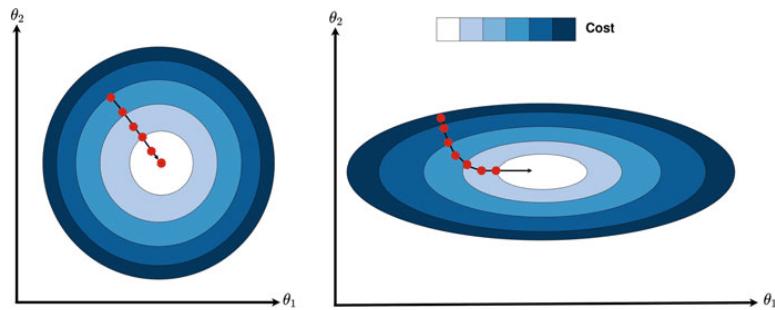


Figura 5.5: Representación del descenso de gradiente. Fuente: [12]

El descenso de gradiente se presenta en varias variantes:

- *Batch Gradient Descent* (BGD): calcula el gradiente de la función de pérdida para todo el conjunto de datos de entrenamiento antes de realizar una única actualización de los parámetros.
- *Stochastic Gradient Descent* (SGD): actualiza los parámetros utilizando el gradiente calculado a partir de una única muestra de entrenamiento en cada iteración, lo que lo hace más rápido pero con actualizaciones más ruidosas. También ayuda a evitar óptimos locales.
- *Mini-Batch Gradient Descent* (MBGD): representa un equilibrio entre BGD y SGD. Actualiza los parámetros utilizando un pequeño subconjunto del conjunto de datos de entrenamiento, denominado mini-batch. Este método equilibra la ventaja de una computación más rápida con el beneficio de una convergencia más estable [58].
- *Adam (Adaptive Moment Estimation)*: es uno de los optimizadores modernos más populares, que combina las ventajas de AdaGrad y RMSProp, calculando tasas de aprendizaje adaptativas individuales para diferentes parámetros [58]. Por un lado, AdaGrad adapta la tasa de

aprendizaje en función del gradiente acumulado: la disminuye para parámetros con gradientes grandes y la mantiene alta para aquellos con gradientes pequeños. Por otro lado, RMSProp mejora esta idea al utilizar una media móvil del cuadrado de los gradientes, lo que evita que la tasa de aprendizaje se vuelva excesivamente pequeña con el tiempo [62].

La tasa de aprendizaje (*learning rate*, η) controla cuánto se ajustan los parámetros en cada iteración del entrenamiento. Las estrategias de ajuste de la tasa de aprendizaje o *LR Schedulers* son mecanismos que la modifican a lo largo del entrenamiento, fundamental para mejorar el rendimiento y la convergencia de los modelos. En lugar de mantener una tasa de aprendizaje fija, estas estrategias permiten que la tasa varíe dinámicamente, lo que puede ayudar a superar mínimos locales y mejorar la estabilidad del entrenamiento.

Técnicas de regularización y mejora

Para mejorar el rendimiento y la generalización de los modelos de Deep Learning, especialmente en escenarios con datos limitados, se emplean diversas técnicas:

- *Dropout* (abandono). Es una técnica de regularización que previene el sobreajuste (*overfitting*). Durante el entrenamiento, el dropout “apaga” (establece a cero) aleatoriamente una fracción de las neuronas en una capa, lo que obliga a la red a aprender representaciones más robustas y menos dependientes de neuronas específicas.



Figura 5.6: Efecto del dropout. A la izquierda una red con dos capas ocultas. A la derecha la red con dropout aplicado (las unidades tachadas son “apagadas”). *Adaptado de [13]*

- Normalización. Busca estandarizar o normalizar las entradas de una capa para estabilizar el aprendizaje, reducir la cantidad de épocas necesarias y permitir el uso de tasas de aprendizaje más altas [62]. Existen

dos tipos de normalización a nivel de capa que son utilizados normalmente:

- Normalización por batch (*Batch Normalization*). Normaliza las entradas a una capa para cada minilote, mediante el cálculo de la media y la varianza sobre las características de todo el mini-batch.
- Normalización de instancia (*Instance Normalization*). A diferencia de la anterior, calcula las estadísticas de cada instancia individual y cada canal de la imagen de forma independiente, es decir, normaliza los datos dentro de cada muestra (o instancia) del batch.

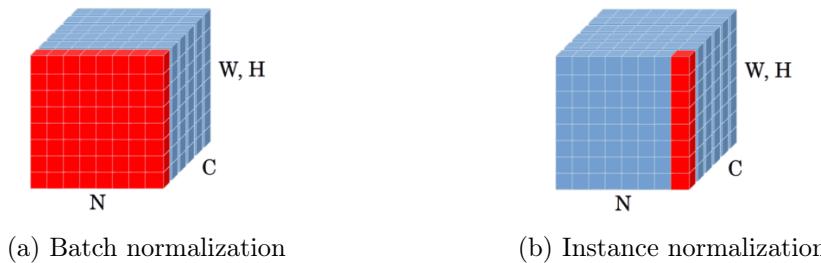


Figura 5.7: Diferencia entre normalizaciones, con N tamaño de batch, C número de canales o mapas de características, H y W alto y ancho de la imagen respectivamente. *Adaptado de [14]*

- *Early Stopping* (detención anticipada). Para prevenir el sobreajuste detiene el entrenamiento del modelo cuando el rendimiento en un conjunto de validación comienza a deteriorarse, es decir, no mejora, en lugar de continuar hasta que la pérdida en el conjunto de entrenamiento sea mínima (sobreajuste).

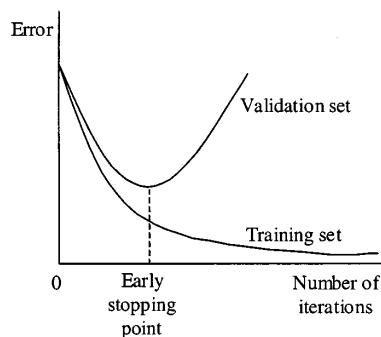


Figura 5.8: Punto de early stopping monitorizando error en validación.
Fuente: [15]

5.2. Técnicas 2D

En esta sección se describen las arquitecturas implementadas para segmentación 2D: U-Net y DeepLabV3. Estas técnicas procesan los datos tridimensionales corte a corte, como se ha mencionado con anterioridad.

5.2.1. U-Net 2D

La arquitectura U-Net, propuesta en [16], se ha convertido en un referente para segmentación médica. Su éxito se debe a la capacidad que posee para combinar información de bajo y alto nivel, lo que mejora la precisión. El nombre proviene de la arquitectura, pues tiene una distintiva forma de “U”:

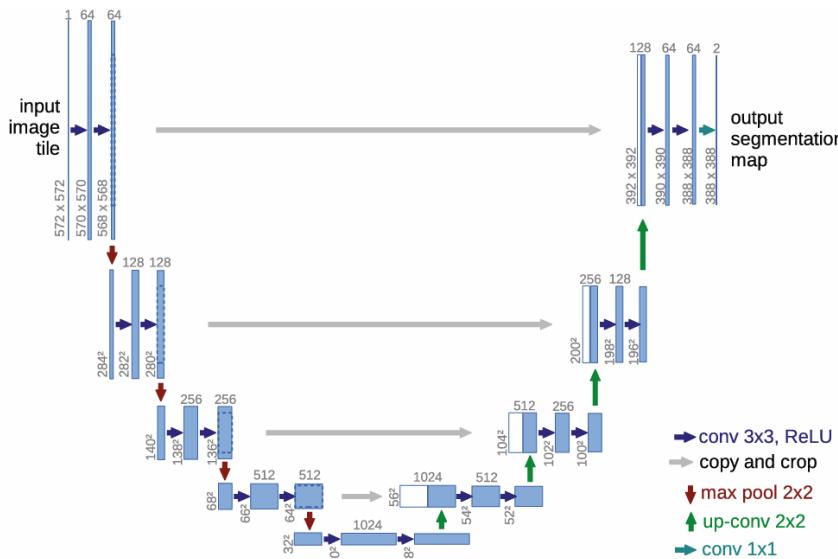


Figura 5.9: Arquitectura de U-Net. Cada recuadro azul corresponde a un mapa de características, con el número de canales y las dimensiones arriba y a la izquierda respectivamente. Fuente: [16]

Se compone de dos caminos principales:

- Camino de contracción (*encoder*). Esta parte de la red (izquierda de la Figura 5.9), similar a una red convolucional típica, extrae características de la imagen de entrada mediante una serie de capas convolucionales y de pooling. Cada nivel de resolución en esta ruta típicamente contiene dos capas convolucionales, seguidas de una función de activación (como ReLU) y una capa de max pooling para reducir la dimensionalidad.

- Camino de expansión (*decoder*). Esta parte se encarga de reconstruir la segmentación de alta resolución a partir de las características de alto nivel extraídas por el codificador. Utiliza capas de deconvolución o *upsampling* seguidas de capas convolucionales.

La innovación clave de la U-Net son las conexiones de salto (*skip connections*), que concatenan las características aprendidas en diferentes niveles del camino de contracción directamente con las capas correspondientes en el camino de expansión (flechas grises en la Figura 5.9). Esto permite que la información de alta resolución (detalles finos) de las primeras capas se combine con la información contextual (características de alto nivel) de las capas profundas, mejorando significativamente el rendimiento de la segmentación, especialmente en la localización de los límites [63].

5.2.2. DeepLabV3

Es otra arquitectura destacada para la segmentación semántica, propuesta en [64]. Se diseñó para abordar los desafíos de la segmentación, especialmente los relacionados con la variación de escala de los objetos en la imagen y sus características principales incluyen:

- Convolución *Atrous* (atrosa o dilatada): es un tipo de convolución que expande el campo de visión de los filtros sin aumentar el número de parámetros ni reducir la resolución espacial de las características, lo que permite capturar información contextual de múltiples escalas sin perder detalle.
- *Atrous Spatial Pyramid Pooling* (ASPP): inspirado en el *spatial pyramid pooling* (SPP), ASPP aplica convoluciones atrous en paralelo con diferentes tasas de dilatación sobre el mapa de características de entrada. Esto le permite capturar el contexto de las imágenes a múltiples escalas, crucial para segmentar objetos de diversos tamaños en la misma imagen.
- Utiliza ResNet-101 como backbone, que es una red profunda que ha demostrado ser efectiva en tareas de visión por computador [63].
- Eliminación del CRF denso (*DenseCRF*). A diferencia de versiones anteriores de DeepLab que utilizaban un DenseCRF para el posprocesamiento, en esta versión se elimina ya que no es necesario.

5.3. Técnicas 3D

En esta sección se describen las arquitecturas 3D empleadas en esta investigación, que van desde las clásicas como 3D U-Net y V-Net hasta técnicas basadas en transformers.

5.3.1. 3D U-Net

La 3D U-Net o U-Net 3D, propuesta en [17], es una extensión de la arquitectura U-Net para el procesamiento de volúmenes tridimensionales. El diseño es el mismo:

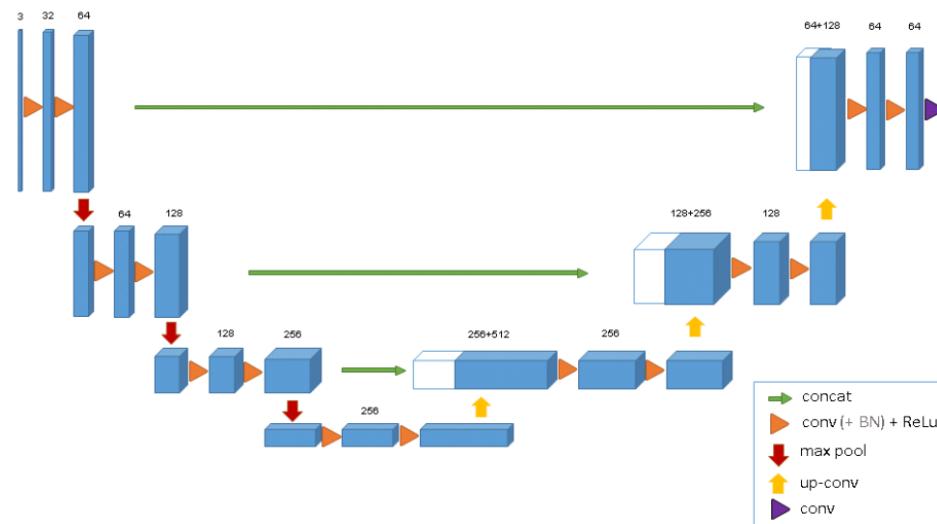


Figura 5.10: Arquitectura de 3D U-Net. Las cajas azules representan los mapas de características, con el número de canales indicado arriba. *Fuente:*

[17]

Al igual que la versión original 2D, esta se compone de un encoder que extrae características y un decoder que reconstruye la segmentación, pero todas las operaciones convolucionales y de pooling son 3D. Las conexiones de salto son igualmente fundamentales, permitiendo la concatenación de características de diferentes niveles en el espacio 3D. Algunas variantes de la 3D U-Net, como la implementada en MONAI, incorporan conexiones residuales para mejorar el entrenamiento de redes profundas [56] o conexiones densas (como en Recurrent 3D-DenseUNet) para una propagación de características más rica [65].

En resumen, para adaptar la U-Net original diseñada para datos 2D a datos 3D (3D U-Net), simplemente hay que reemplazar las operaciones 2D

por sus equivalentes 3D.



Figura 5.11: Diferencia entre convolución 2D y 3D. *Fuente:* [18]

5.3.2. Attention U-Net

La arquitectura Attention U-Net es un modelo propuesto en [19] para la segmentación de imágenes médicas que aprende automáticamente a enfocar estructuras objetivo de diversas formas y tamaños. Busca resolver la limitación que tienen las arquitecturas en forma de “U” (U-Net) al tratar con órganos que presentan gran variabilidad en forma y tamaño entre pacientes, como el páncreas, problema para el que fue definida.

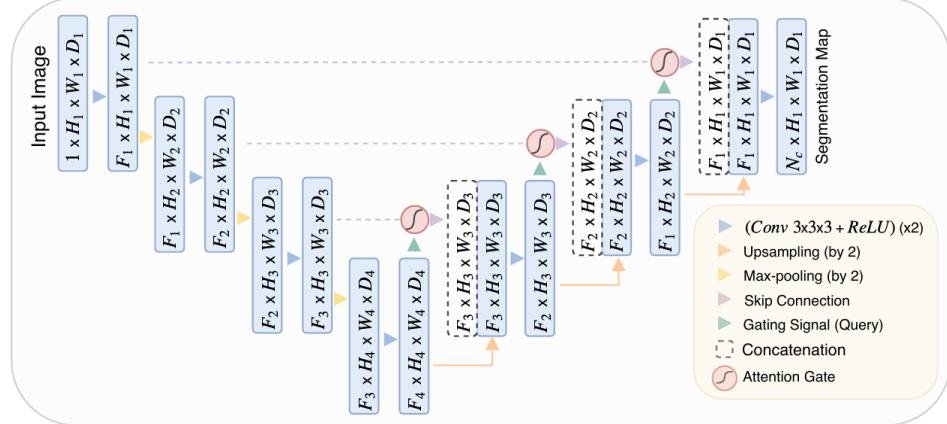


Figura 5.12: Arquitectura de Attention U-Net. *Fuente:* [19]

La innovación clave de Attention U-Net son las Puertas de Atención (AGs). Permiten que los modelos entrenados aprendan implícitamente a suprimir regiones irrelevantes en una imagen de entrada, mientras resaltan las características útiles para una tarea específica. Esto elimina la necesidad de módulos externos de localización de tejido u órgano.

Las AGs se integran en la arquitectura estándar de U-Net para resaltar las características que se pasan a través de las conexiones de salto. La infor-

mación extraída de la escala más gruesa se utiliza en el *gating* para eliminar respuestas irrelevantes y ruidosas en dichas conexiones. Al realizar esto justo antes de la concatenación, se garantiza que solo se fusionen las activaciones realmente relevantes.

Las Puertas de Atención no solo filtran las activaciones de las neuronas durante el paso hacia adelante, sino que también lo hacen durante el paso atrás o backpropagation, paso durante el cual los gradientes originados en las regiones de fondo se ponderan menos, permitiendo que los parámetros del modelo en capas más superficiales se actualicen de acuerdo con las regiones importantes para la tarea.

Para que funcione esta técnica, existen los coeficientes de atención, que no son más que valores que identifican regiones prominentes de la imagen y filtran las respuestas, de modo que el resultado de las AGs es la multiplicación elemento a elemento de los mapas de características de entrada y estos coeficientes.

De manera resumida, el modelo sigue un diseño de U-Net estándar, con una ruta de codificación (contracción) que progresivamente filtra y submuestra la imagen de entrada en cada escala, y una ruta de decodificación (expansión) que reconstruye el mapa de segmentación. Las conexiones de salto combinan características gruesas y finas, con puertas de atención insertadas para filtrar las características propagadas.

5.3.3. DynUNet

DynUNet (de *Dynamic U-Net*) es la implementación basada en nnU-Net [66, 67] (influenciada también por [68]) dentro de la librería MONAI, desarrollada para ofrecer una arquitectura adaptable, eficiente y robusta en tareas de segmentación médica volumétrica. Su filosofía central, encapsulada en su nombre *no-new net*, es que no es necesario crear arquitecturas de red completamente nuevas, sino que el éxito se logra mediante la adaptación sistemática y la configuración óptima de una arquitectura U-Net básica a diferentes conjuntos de datos y tareas de segmentación.

Los puntos clave de nnU-Net incluyen:

- Extracción automática de las propiedades del conjunto de datos como pueden ser el tamaño de la imagen, el espaciado de los vértices o las proporciones de las clases, que denominan *dataset fingerprint*, y un *pipeline fingerprint* que encapsula las decisiones de diseño del algoritmo de segmentación.
- Uso de reglas heurísticas que eliminan la necesidad de optimizar la configuración de forma manual.

- Enfocado en que una U-Net bien configurada es difícil de superar por variaciones complejas que incluyan conexiones residuales, densas, mecanismos de atención, etc.

Sin embargo, la versión de MONAI (DynUNet) está más limitada, compuesta por un bloque de entrada, n bloques de downsampling, un cuello de botella (bottleneck) y $n + 1$ bloques de upsampling, pero todos ellos configurables mediante secuencias de tamaños de kernel y strides, permitiendo ajustar su profundidad y resolución de manera automática. También permite conexiones residuales y tamaños de kernel distintos por capa (anisotrópicos).

Es por ello que se trata de un modelo más configurable que la mencionada U-Net, pero no incluye el componente de ajuste automático de nnU-Net.

5.3.4. V-Net

V-Net es una red neuronal completamente convolucional diseñada específicamente para segmentación de imágenes médicas en 3D, propuesta en [20]. Al igual que la U-Net, V-Net sigue una arquitectura de codificador-decodificador con conexiones de salto, y su nombre también proviene de su característica forma en “V”:

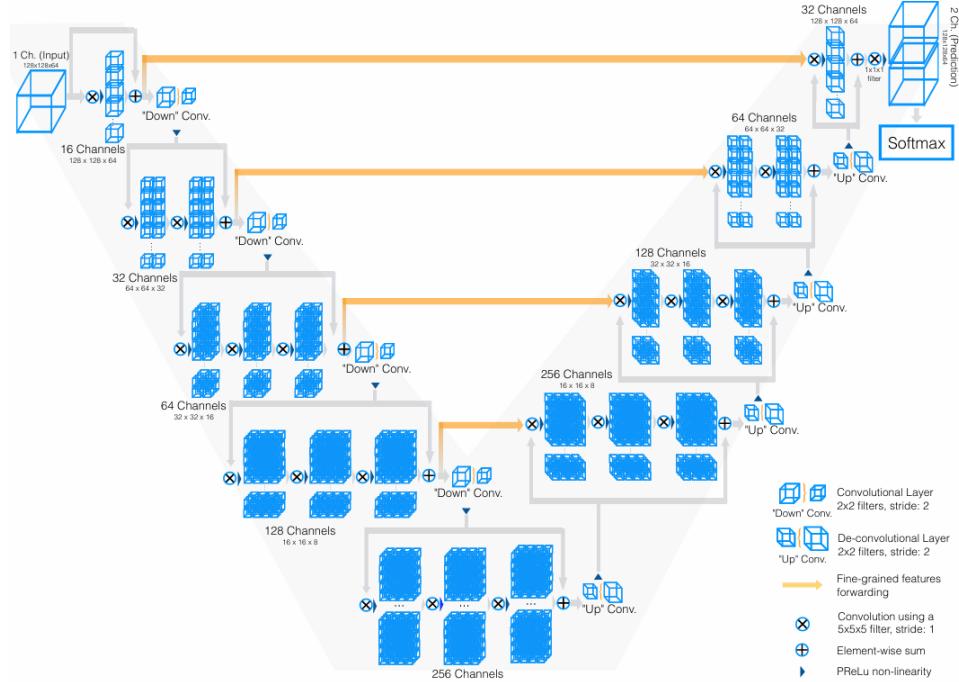


Figura 5.13: Arquitectura de la V-Net. Fuente: [20]

Las características de este modelo, además de las conexiones de salto, son:

- En el codificador cada etapa contiene de una a tres capas convolucionales. Utiliza convoluciones volumétricas con kernels de $5 \times 5 \times 5$ véxoles. Para reducir la resolución de los datos progresivamente se realizan convoluciones con kernels $2 \times 2 \times 2$ y un paso (stride) de 2, que reemplazan las operaciones de pooling. El número de canales de características se duplica en cada etapa de compresión [20].
- En el decodificador se expanden los mapas de características de menor resolución, aumentando el tamaño de los datos mediante deconvolución. La Figura 5.14 muestra la diferencia entre una convolución y una deconvolución (o convolución transpuesta) y cómo ayudan a reducir y aumentar respectivamente el tamaño de la imagen.
- Uso de la función de activación PReLU en toda la red.

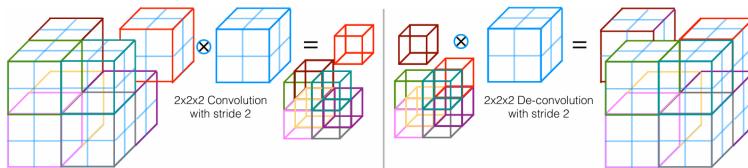


Figura 5.14: Convoluciones (izquierda) con stride adecuado pueden reducir el tamaño de los datos, deconvoluciones (derecha) lo incrementan proyectando cada véxol a una regió más amplia a través del kernel (cubo azul). *Fuente:* [20]

5.3.5. SegResNet

SegResNet es una arquitectura propuesta en [21], donde se trata la segmentación semántica de tumores cerebrales en imágenes de resonancia magnética. También sigue una arquitectura de codificador-decodificador basada en CNN pero, a diferencia de las anteriores, se caracteriza por ser asimétrica, teniendo un codificador más grande y un decodificador más pequeño. Posee una rama de autoencoder variacional (VAE) que reconstruye la imagen de entrada, sirviendo como una forma de regularización para el codificador compartido pero que sólo se utiliza durante el entrenamiento.

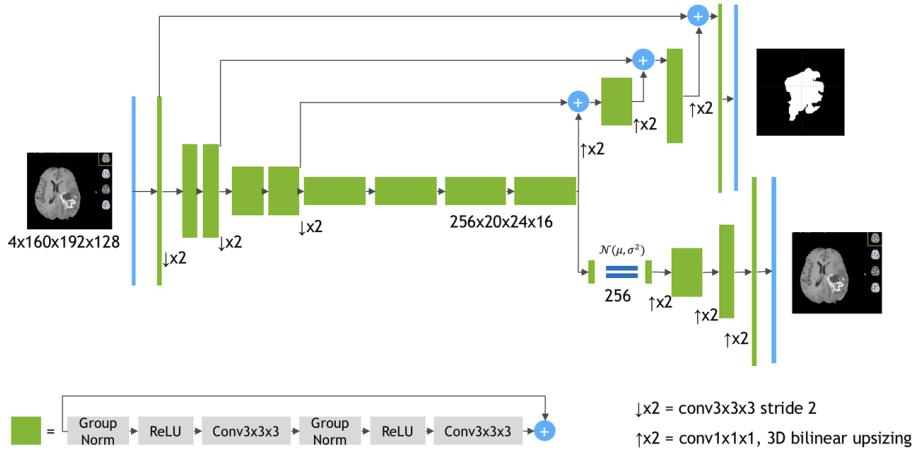


Figura 5.15: Arquitectura de SegResNet. Fuente: [21]

Profundizando en el codificador, utiliza bloques ResNet, donde cada uno contiene dos convoluciones con normalización y función de activación ReLU, seguidas de una conexión de salto aditiva de identidad. Para la normalización, se emplea la Normalización de Grupo (*Group Normalization*), que es una técnica intermedia entre las mencionadas anteriormente. Como es habitual, el codificador reduce progresivamente las dimensiones de la imagen a la mitad utilizando convoluciones y, al mismo tiempo, duplica el tamaño de las características. Todas ellas son $3 \times 3 \times 3$ con un número inicial de 32 filtros y en el punto final del codificador tiene un tamaño espacial 8 veces menor que en la entrada.

En cuanto al decodificador, su estructura se asemeja a la del codificador. Cada etapa inicia con un aumento de tamaño en el que se disminuye el número de características a la mitad y duplicando la dimensión espacial. Luego, se añade la salida del codificador del nivel espacial equivalente. El final del decodificador tiene el mismo tamaño que la imagen original, con un número de características igual al de entrada, seguido por otra convolución $1 \times 1 \times 1$ a 3 canales y una función sigmoide de acuerdo con el problema que pretendía resolver.

5.3.6. UNETR

La arquitectura UNETR (UNet TRansformers) es un modelo novedoso diseñado para la segmentación de imágenes médicas 3D por [22]. Está inspirado en transformadores (Procesamiento del Lenguaje Natural) y dirigido al aprendizaje de secuencias de largo alcance y captura de contexto global.

Sigue un diseño en forma de “U”, igualmente con un patrón de contracción

y otro de expansión, donde un codificador de transformador se conecta a un decodificador a través de conexiones de salto en diferentes resoluciones.

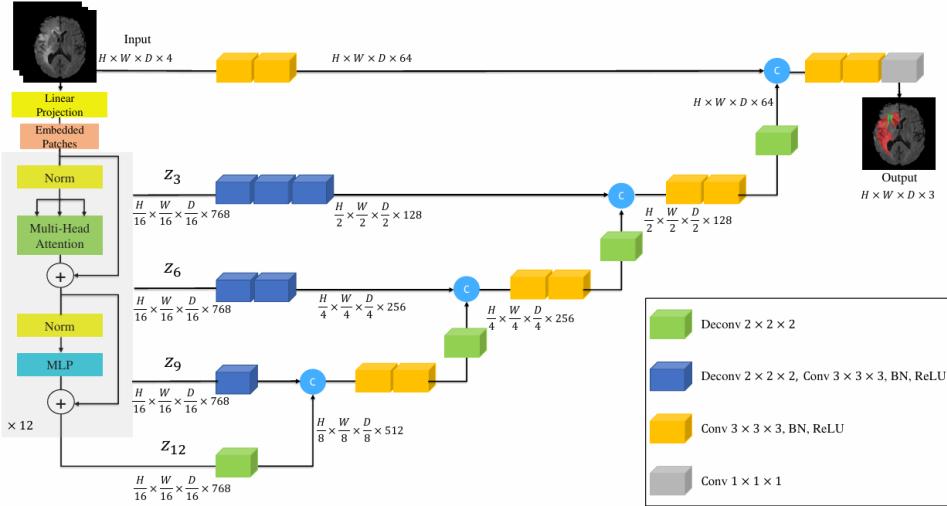


Figura 5.16: Arquitectura de UNETR. Fuente: [22]

El codificador de transformador (Transformer Encoder) procesa la entrada 3D de la siguiente manera: el volumen de entrada se divide en parches uniformes no solapados de igual tamaño y se aplana para crear una secuencia de todos los parches. Esta secuencia se proyecta mediante una capa lineal a un espacio de *embedding* (espacio vectorial de representaciones de los datos con el que trabajan los transformadores). El codificador utiliza una serie de bloques de transformador que comprenden subcapas de autoatención multicabeza (captura dependencias de largo alcance y modela el contexto global calculando los pesos de atención) y perceptrón multicapa (dos capas lineales con funciones de activación GELU). Antes de cada capa se aplica una normalización.

El decodificador sí está basado en CNN. Las salidas de las diferentes capas del transformador se convierten a tensores para que puedan ser tratados por el decodificador. En cada resolución, se aplican capas convolucionales $3 \times 3 \times 3$ consecutivas seguidas de capas de normalización (batch). En la salida de la última capa se aplica una deconvolución para duplicar la resolución y se concatena con la salida previa del transformador. La concatenación pasa por capas convolucionales $3 \times 3 \times 3$ consecutivas junto con una deconvolución al final para el upsampling. Este proceso se repite hasta llegar al tope del decodificador, cuando alcanza la resolución original y pasa por una capa convolucional $1 \times 1 \times 1$ con una función de activación softmax para generar las predicciones. Al igual que el modelo anterior, la salida está adaptada al tipo de problema.

En resumen, UNETR utiliza un transformador como codificador para capturar dependencias de largo alcance y contexto global y un decodificador basado en CNN y conexiones de salto para el contexto local.

5.3.7. Swin UNETR

Presentado en [23], surgió con la idea de abordar las limitaciones que presentaban los enfoques previos como UNETR [22], ya que, por ejemplo, tenían una resolución fija en el codificador. Swin UNETR extrae características multiescala para solventar esto. Su diseño también tiene forma de “U”:

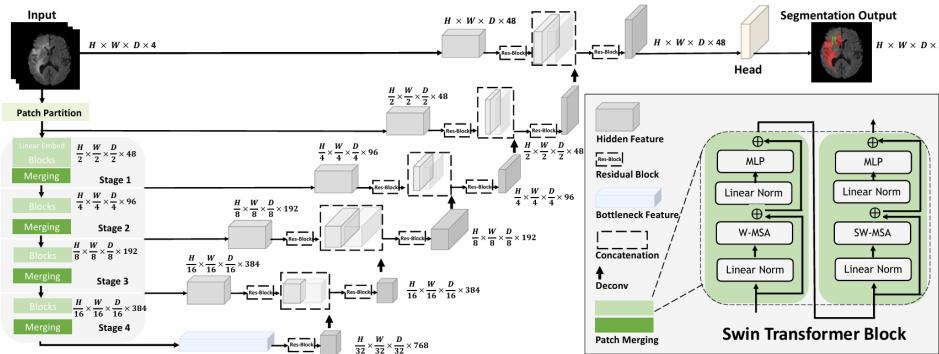


Figura 5.17: Arquitectura de Swin UNETR. Fuente: [23]

El codificador es un Swin Transformer. Inicialmente, se crean parches 3D uniformes no superpuestos del volumen de entrada, pero en lugar de ser aplazados como en UNETR pasan por una capa de partición de parches que crea una secuencia de tokens 3D. Estos tokens se proyectan en el espacio de embedding.

En este caso, la autoatención se calcula en ventanas no superpuestas. Las regiones de ventana particionadas se desplazan en las sucesivas capas del transformador, lo que permite capturar dependencias de largo alcance a la vez que se mantiene la eficiencia computacional. Sus capas son módulos de autoatención multicabeza (W-MSA y SW-MSA) y subcapas de perceptrones multicapa, junto con normalización por capas.

En el decodificador (basado en CNN), se incluyen bloques residuales consistentes en dos capas convolucionales $3 \times 3 \times 3$ normalizadas por capas de normalización de instancia. Haciendo uso de desconvoluciones se amplía la resolución de los mapas de características y las salidas resultantes se concatenan con las salidas de la etapa anterior del codificador. Al final, una capa convolucional de $1 \times 1 \times 1$ y una función de activación sigmoide generan la

segmentación. Como en modelos anteriores, esta característica es inherente al problema para el que se desarrolló (tumores cerebrales en resonancias magnéticas).

Capítulo 6

Desarrollo del marco experimental de trabajo

Como se ha expuesto en la Sección 3.7, el trabajo se ha dividido en dos etapas. La primera fase permitió entender profundamente cada parte del pipeline: cómo se comportaban los datos, cómo impactaban las transformaciones y qué limitaciones tenía el enfoque base. Gracias a esto, en la segunda fase, con MONAI, se han podido aplicar técnicas más avanzadas, optimizar el rendimiento y obtener resultados clínicamente más relevantes en base a un conocimiento firme y no por prueba y error. Además de las implementaciones necesarias para el problema, se han realizado otras con el objetivo de optimizar tanto tiempo como recursos. Estos aspectos se detallan en este capítulo.

6.1. Implementaciones

Etapa de implementación propia

Esta etapa tiene como objetivo comprender de manera profunda el problema y el flujo de datos, entrenamiento y evaluación mientras se trata de resolver el problema sin dependencias adicionales.

Las ventajas de este enfoque son:

- Control absoluto sobre cada paso.
- Aprendizaje sólido sobre el tipo de dato, las transformaciones y las métricas.
- Identificación de cuellos de botella, ya sea en la lectura y procesamiento de datos o en las fases del entrenamiento.

Aunque el resultado no haya sido óptimo, ha permitido entender por qué no funcionaba bien (no solo que no funcionaba) y reconocer las partes del pipeline que necesitaban mejoras, justificando la necesidad de incluir bibliotecas especializadas como MONAI.

En cuanto a las implementaciones, han seguido los patrones típicos del desarrollo de software, creando clases (módulos) interconectadas para cada necesidad encontrada, divididas en paquetes según funcionalidad. Los paquetes implementados son:

- **data**. Contiene Datasets (2D y 3D, con posibilidad de cargar o no los datos en memoria), preprocesamiento estático (lectura de volúmenes y escritura corte a corte para 2D y en subvolúmenes para 3D), creación de splits de entrenamiento, validación y test, transformaciones (ventana, orientación, etc) y aumento de datos (2D y 3D) entre otros.
- **evaluation**. Contiene el evaluador (**Evaluator**), una clase diseñada para evaluar un modelo entrenado en un determinado conjunto de datos.
- **inference**. Contiene el predictor (**Predictor**), una clase diseñada para inferir un paciente o conjunto de pacientes a partir de un modelo entrenado.
- **losses**. Incluye las pérdidas combinadas (Cross-Entropy con Dice Loss con y sin ponderación o Dice con Focal Loss), Dice Loss y Focal Loss.
- **metrics**. Incluye las métricas implementadas (Dice, intersección sobre la unión, precisión y recall) acumulables por época o no.
- **models**. Contiene una versión modificada de DeepLabV3 para recibir entrada de un canal (como son las TC), dropout la capa 3 del bloque ResNet y ASPP, con ResNet-101 como backbone y preentrenada¹, e implementaciones propias de U-Net 2D y 3D haciendo uso del paquete **nn** de PyTorch para la construcción de modelos capa a capa.
- **training**. Contiene el entrenador (**Trainer**) (encargado de entrenar un modelo y almacenar los resultados y pesos), los *callbacks* necesarios para el entrenamiento (gestión de puntos de control o *checkpoints* del entrenamiento y early stopping), y otras utilidades como establecimiento de semillas aleatorias en todos los generadores. También contiene el paquete **setup**, que sigue el patrón de diseño Factory, permitiendo crear los componentes del entrenamiento de manera flexible.
- **utils**. Utilidades de apoyo al resto de paquetes (manejo de datos, evaluaciones, exportaciones, *logging* o el notificador). Destacar el módulo

¹Documentación de PyTorch sobre los pesos de DeepLabV3 con ResNet-101

de configuración, que es el pilar fundamental de todo el desarrollo, pues lee y procesa un archivo de configuración YAML, y el subpaquete **visualization**, con implementaciones para visualizaciones.

A parte de lo anterior, perteneciente al paquete **src**, se han implementado *scripts* para cada una de las tareas: preprocessamiento (estático) de datos, entrenamiento, entrenamiento sin validación, evaluación de modelos y predicción o inferencia de datos nuevos.

La idea detrás de la implementación propia es tener archivos de configuración YAML, con la información específica de cada experimento, de modo que solo sea necesario cambiar estos archivos para generar nuevos experimentos, teniendo siempre un registro de las configuraciones utilizadas en cada uno de ellos y facilitando la reproducibilidad y la flexibilidad.

Es importante mencionar que para algunas de las implementaciones anteriores sí se ha hecho uso de MONAI internamente, pero siempre después de haberlo hecho de forma propia y adaptando los datos (como pueden ser la salida del modelo) al formato requerido por MONAI.

Etapa de implementación con MONAI

Durante esta etapa, el objetivo es aplicar las lecciones aprendidas con herramientas diseñadas específicamente para imágenes médicas, ofrecidas por MONAI.

A modo de resumen, aporta una serie de ventajas que incluyen Datasets acelerados, transformaciones avanzadas y pipelines reproducibles y reutilizables. Como resultado, se consigue una mejora significativa en los resultados.

Tal y como se recalca en el apartado anterior, se ha introducido MONAI en el ecosistema propio, pero lo realmente importante realizado durante esta etapa con este framework es el uso del MONAI *Bundle*², un paquete estructurado que agrupa todo lo necesario para una solución de Deep Learning en imágenes médicas:

- Pesos del modelo (como archivos de PyTorch entre otros).
- Configuraciones: archivos JSON o YAML que definen el pipeline completo, es decir, Datasets, preprocessamiento, entrenamiento, validación, inferencia y posprocesamiento.
- Metadatos: descripción del problema, licencias, esquemas de entrada y salida, versiones...

²https://docs.monai.io/en/stable/bundle_intro.html

La ventaja principal de este enfoque es la portabilidad y reproducibilidad ya que incluye todo lo necesario y permite reconstruir o modificar el pipeline completo únicamente con la configuración. Es la misma metodología que la aplicada en la implementación propia (véase el apartado anterior) pero mejorada y optimizada al tener todo el contenido de la configuración implementado en MONAI. Es el utilizado en el MONAI Model Zoo ³.

Dado que el proyecto no trata de publicar un bundle, se han omitido los archivos de metadatos y logging, creando simplemente los correspondientes a las configuraciones de cada experimento (o modelo). Para ejecutar un entrenamiento, basta con utilizar el siguiente comando:

```
python -m monai.bundle run -config_file .../config.yaml
```

Sin embargo, este paquete presenta algunas limitaciones, entre ellas el uso de early stopping, pues no puede ser definido en la configuración, y tampoco es una técnica muy utilizada en el MONAI Model Zoo. Para solventar esto, se ha desarrollado un script que lee y procesa la configuración haciendo uso del `ConfigParser` del bundle, crea el manejador de early stopping y lo agrega al entrenador. Por último, agrega el disparador de este manejador al evaluador, de forma que cuando se completa una validación se comprueba si debe detenerse el entrenamiento. Con esto se consigue optimizar aún más si cabe el proceso, evitando tener modelos entrenando indefinidamente a sabiendas de que no van a mejorar.

6.2. Reproducibilidad

De acuerdo con la sección anterior, la reproducibilidad de los experimentos se ha garantizado con los siguientes aspectos:

- Uso de archivos de configuración YAML. Tanto en la implementación propia como en el uso del bundle de MONAI, los experimentos se realizan siguiendo las especificaciones de un archivo de configuración `.yaml`.
- Establecimiento de semillas aleatorias: en la implementación propia se establecen las semillas que usan los generadores aleatorios de NumPy, `random` y PyTorch, además de CUDA junto con la activación del determinismo, según la documentación oficial de PyTorch para lograr reproducibilidad⁴. Con el bundle de MONAI se indica en la configuración la semilla a la que se establece el determinismo⁵.

³<https://monai.io/model-zoo.html#/>

⁴<https://docs.pytorch.org/docs/stable/notes/randomness.html>

⁵https://docs.monai.io/en/stable/utils.html#monai.utils.misc.set_determinism

A pesar del esfuerzo y el compromiso por asegurar la reproducibilidad de esta investigación, lamentablemente **no será posible reproducir los experimentos** debido a que el conjunto de datos es privado y no se puede liberar, como se explicaba en la Sección 3.6.

6.3. Automatizaciones

El uso de ficheros YAML para definir experimentos ya supone en sí mismo un tipo de automatización, pues no es necesario modificar ningún aspecto de la implementación, simplemente generar tantas configuraciones como experimentos se deseen.

No obstante, en esta sección se detallan otras dos automatizaciones que se han desarrollado, con el objetivo de optimizar el tiempo y los recursos, que van más allá de las configuraciones y los scripts.

Las dos automatizaciones están implementadas en la clase `Notifier` del paquete `src.utils`. Consiste en un notificador cuya finalidad es automatizar la comunicación de eventos clave del entrenamiento a través de Telegram, permitiendo un seguimiento remoto, y centralizar el registro de resultados en Google Sheets, facilitando la trazabilidad y análisis de experimentos. Esto se ha hecho principalmente para la etapa de implementaciones propias, donde podía haber problemas en las implementaciones y no se hacía uso de herramientas como TensorBoard⁶ para seguir y visualizar métricas.

Bot de Telegram

Se ha creado un bot de Telegram y añadido a un Grupo de Telegram con la opción de Temas activada. La Telegram Bot API⁷ es una interfaz basada en HTTP, por lo que haciendo uso de la librería `requests` de Python para enviar peticiones HTTP simples se puede automatizar el envío de mensajes a distintos Temas del grupo por parte del bot. Los mensajes contemplados durante el entrenamiento son: comienzo, error, actualización (cada 10 % de las épocas) y final, y se envían al Tema con el nombre del experimento, lo que permite un seguimiento más claro. Cada uno de estos mensajes contiene información relevante (causa del error, métricas obtenidas, etc.).

⁶<https://www.tensorflow.org/>

⁷<https://core.telegram.org/bots/api>



(a) Notificación de error

(b) Notificación de final

Figura 6.1: Mensajes del notificador para el experimento “deep_aug”: (a) comienzo del *fold* 5 del entrenamiento y error y (b) actualización y aviso de entrenamiento completado. *Elaboración propia*

API de Google Sheets

Una vez que se completaba un entrenamiento, el notificador añade una entrada en una hoja de cálculo de Google con los resultados. Esto es posible gracias a la integración entre la API de Google Sheets⁸, ofrecida por Google Cloud⁹, con las bibliotecas `gspread`¹⁰ y `google-auth`¹¹ (`oauth2client` está deprecada en su favor), que permiten la autenticación y control de las hojas de cálculo.

Se puede ver una imagen de la hoja de cálculo en el Apéndice A.

⁸<https://developers.google.com/workspace/sheets/api/guides/concepts>

⁹<https://console.cloud.google.com/marketplace/product/google/sheets.googleapis.com>

¹⁰<https://docs.gspread.org/en/v6.1.4/>

¹¹<https://google-auth.readthedocs.io/en/latest/>

Capítulo 7

Análisis experimental

En este capítulo se aborda el análisis de los experimentos realizados. Para ello, se presentan las métricas utilizadas y los métodos de validación empleados. Se detallarán los experimentos realizados junto con sus configuraciones, así como los resultados obtenidos en validación y test. Estos últimos serán analizados tanto de manera cuantitativa, atendiendo a las métricas, como cualitativa, visualizando algunas segmentaciones.

7.1. Métricas y métodos de validación

Métricas

Antes de definir las métricas específicas utilizadas para evaluar cada modelo, es importante introducir los conceptos básicos que sustentan su cálculo. Dado que se trata de un problema de segmentación semántica a nivel de voxel, cada predicción se puede comparar con la ground truth y clasificarse como uno de los siguientes:

- *True Positive* (TP): véxeles correctamente clasificados como pertenecientes a la clase de interés.
- *True Negative* (TN): véxeles correctamente clasificados como no pertenecientes a la clase de interés.
- *False Positive* (FP): véxeles incorrectamente clasificados como pertenecientes a la clase de interés.
- *False Negative* (FN): véxeles incorrectamente clasificados como no pertenecientes a la clase de interés.

A partir de estas cantidades, se pueden derivar diferentes métricas que permiten evaluar el desempeño de los modelos desde distintas perspectivas. Las principales métricas consideradas en este trabajo son:

- **Dice Similarity Coefficient (DSC).** Mide la superposición entre la segmentación predicha y la segmentación real. Se calcula como:

$$DSC = \frac{2 \cdot TP}{2 \cdot TP + FP + FN} \quad (7.1)$$

Este coeficiente toma valores entre 0 y 1, siendo 1 una coincidencia perfecta en la que la predicción es igual a la ground truth y no hay clasificaciones incorrectas ($FP = FN = 0$), y 0 indica que no hay superposición.

- **Intersection over Union (IoU).** También conocida como *Jaccard Index*, la intersección sobre la unión mide la proporción de intersección sobre la unión de las segmentaciones:

$$IoU = \frac{TP}{TP + FP + FN} \quad (7.2)$$

Al igual que el DSC, toma valores entre 0 y 1, siendo 1 una predicción perfecta.

- **Recall (Sensibilidad).** Indica la proporción de verdaderos positivos detectados respecto al total de elementos positivos reales:

$$Recall = \frac{TP}{TP + FN} \quad (7.3)$$

- **Precision (Precisión positiva).** Mide la proporción de verdaderos positivos entre todos los elementos clasificados como positivos:

$$Precision = \frac{TP}{TP + FP} \quad (7.4)$$

Además de estas métricas indicadas por el equipo clínico del Hospital Universitario Clínico San Cecilio, se ha considerado la inclusión de otra basadas en distancia de la superficie, relevante respecto a la precisión de los límites:

- **Hausdorff Distance (percentil 95, HD95).** Evalúa la máxima discrepancia entre los bordes de la predicción y la ground truth, atenuando el impacto de los valores atípicos al considerar el percentil 95 de las distancias.

Estas métricas se calculan por clase y por paciente, con el objetivo de capturar tanto el rendimiento global como su variabilidad individual. En los resultados se reportan tanto los valores promedio por paciente como los valores agregados sobre todo el conjunto de evaluación.

Métodos de validación

Durante los primeros experimentos, el método de validación que se utilizó fue *Hold-Out*, consistente en una división fija del conjunto de datos en subconjuntos de entrenamiento y validación. Esta división se realizó una única vez de manera aleatoria, con una proporción del 80 % para entrenamiento y el 20 % restante para validación, y se almacenó en un archivo JSON de modo que todos los experimentos usaran la misma división.

Al poco tiempo, para que el entrenamiento no dependiera de una partición fija y en línea con las prácticas comunes en la investigación en Deep Learning aplicado a imágenes médicas, se consideró la validación cruzada (*k-fold cross-validation*). Este método, que implica dividir el conjunto de datos en múltiples subconjuntos (5 en este caso) para entrenar y validar el modelo de manera iterativa, es frecuentemente empleado, especialmente cuando se trabaja con conjuntos de datos limitados, con el fin de obtener una evaluación más robusta del rendimiento del modelo y mejorar la fiabilidad de los resultados al reducir la varianza de la estimación del error, como se reporta en numerosos estudios [33, 36, 69, 70].

No obstante, durante las primeras fases de experimentación y entrenamiento con validación cruzada, se observó que su aplicación con 5 folds no mostraba diferencias significativas en los resultados entre los distintos folds. Esta consistencia empírica sugiere que el beneficio de la validación cruzada en términos de fiabilidad de la estimación, para este problema y conjunto de datos específicos, no justifica el incremento en los recursos computacionales y el tiempo de entrenamiento que implica ejecutar el modelo 5 veces.

Siguiendo con la línea de trabajo del proyecto, con el objetivo de optimizar los recursos disponibles y el tiempo de experimentación, se tomó la decisión de volver al enfoque Hold-Out. La división se hizo con la misma proporción (80 % entrenamiento y 20 % validación) pero **estratificada** en lugar de aleatoria, que significa dividir el conjunto de datos de manera que la proporción de clases o valores de la variable objetivo se mantenga igual en los subconjuntos resultantes respecto al dataset original. De las opciones que hay para condicionar la división de los datos (véase la Sección 4.1.1), se decidió por hacerlo en función del tipo de tumor:

Cuadro 7.1: Distribución de tipos de tumor en split de entrenamiento y validación

Tipo de tumor	Entrenamiento (%)	Validación (%)
PDAC	50,00 %	50,00 %
PDAC borderline	35,71 %	33,33 %
TNE	10,00 %	11,11 %
Otro	4,29 %	5,56 %

Esta estratificación ayuda a mitigar el riesgo de sesgo en la validación y a asegurar que el modelo pueda generalizar bien a diferentes tipos de tumores, a pesar del desequilibrio de clases inherente al problema. Numerosos estudios como [28, 29, 43, 59] también emplean Hold-Out en lugar de k-fold cross-validation en un contexto de segmentación semántica en imágenes médicas.

En conclusión, la elección del Hold-Out con una división estratificada se justifica no solo por las observaciones empíricas de ausencia de diferencia significativa en los resultados de 5-fold CV, sino también por su eficiencia computacional y su consistencia con metodologías empleadas en otros estudios de segmentación de imágenes médicas que enfrentan el desafío de datasets limitados, complementándose con técnicas de aumento de datos para asegurar la robustez y generalización del modelo.

7.2. Descripción de los experimentos

Implementaciones propias

Las implementaciones propias se centraron en experimentos con una modificación de DeepLabV3 en 2D y U-Net en 2D y 3D. Se han realizado multitud de estos experimentos, variando los parámetros y las funciones de pérdida utilizadas, todo ello disponible tanto en las configuraciones de los entrenamientos del repositorio del proyecto como en el Apéndice A.

La inferencia se ha realizado slice a slice en los entrenamientos 2D y subvolumen a subvolumen en los 3D. A continuación se describen los ajustes de un experimento para cada caso.

U-Net 2D

Este experimento toma los datos originales construyendo un Dataset 2D, que almacena los volúmenes slice a slice, y le aplica el siguiente preprocesamiento: aplicación de ventana (nivel 50 y ancho 400), normalización de intensidad, recorte desde los bordes de 120 píxeles (60 por cada lado) y redimensiona a 512×512 . Como aumento de datos incluye: recorte aleatorio de 256×256 , transformaciones afines (escalado de imagen aleatorio entre 0,95 y 1,05 veces su tamaño, desplazamiento de la imagen hasta un 2% en cada eje, y rotación entre -10 y 10 grados) con una probabilidad 0,2; ajuste aleatorio del brillo y contraste (ambos hasta un 20%) con probabilidad 0,3 de aplicarse; desenfoque gaussiano (tamaño de kernel gaussiano entre 3 y 7) con probabilidad 0,3; transformación elástica (deforma la imagen de manera elástica con una intensidad de 1 y una suavidad de 50) con probabilidad 0,3; y por último distorsión de la imagen usando una cuadrícula (5 divisiones y límite de distorsión de 0,3) con una probabilidad de 0,3.

El modelo se construye sobre la implementación de MONAI en 2D, con canales [16, 32, 64, 128, 256], strides [2, 2, 2, 2], 2 unidades residuales, dropout de 0,1 y normalización de batch.

El tamaño del batch es 4 (4 slices aleatorias entre el conjunto formado por las slices de todos los pacientes, pudiendo mezclarse entre sí), y se entrena durante 100 épocas como máximo, con función de pérdida Dice Loss (`MulticlassDiceLoss`), con pesos 0,3 para fondo, 1 para páncreas, arterias y venas, y 2 para tumor. La tasa de aprendizaje inicial es 0,0001 y el decaimiento de pesos (*weight decay*) 0,00001. Se aplica early stopping con paciencia 15.

DeepLabV3

Este experimento se basa en la implementación modificada de DeepLabV3. El Dataset 2D se construye a partir de los datos procesados estáticamente (almacenados slice a slice en formato NumPy para los volúmenes y PNG para las segmentaciones). Las transformaciones que se aplican a las imágenes son: recorte a la región de interés (encontrada mediante análisis de datos), dejándola con dimensión 256×256 en lugar de 512×512 ; aplicación de ventana (nivel 50 y ancho 400) y normalización de intensidades. Como medidas de data augmentation se usan las mismas que en el experimento anterior.

El modelo se construye con pesos preentrenados y con dropout 0,2. Se entrena durante 100 épocas como máximo. La función de pérdida es una combinación de Dice Loss y Focal Loss, con $\gamma = 2$ para la Focal Loss. La tasa de aprendizaje inicial es 0,0001 y el weight decay es 0,00001. Se aplica early stopping con paciencia 15.

U-Net 3D

Análogo al anterior pero en 3D, el Dataset 3D se construye a partir de los subvolúmenes generados mediante el preprocessamiento estático, de dimensiones $256 \times 256 \times 64$ por el recorte en alto y ancho a la región de interés y división en subvolúmenes de profundidad 64 con desplazamiento de 32 véxeles. Por tanto, el tamaño del batch se reduce a 2. Las transformaciones determinísticas que se aplican a los subvolúmenes son únicamente la ventana de HU (nivel 50 y ancho 400) y la normalización de intensidades, puesto que el resto de transformaciones mencionadas (orientación común, rotaciones, recorte al ROI, etc) ya se ha realizado durante el preprocessamiento estático.

El modelo se crea empleando la implementación propia de la arquitectura, utilizando 32 canales base, lo que genera una U-Net con canales [32, 64, 128, 256, 512], con kernels de tamaño 3, normalización de batch y función de activación ReLU.

La función de pérdida de este experimento es la combinación de Dice Loss con Focal Loss (con $\gamma = 2$). La tasa de aprendizaje inicial se establece a 0,00005 y el weight decay a 0,000005.

La duración del entrenamiento es, como máximo, 200 épocas, e incorpora early stopping con paciencia 20. Se valida mediante 5-fold cross-validation.

Implementados con MONAI

Todos los experimentos de este bloque tienen el preprocesamiento desarrollado en la Sección 4.2.2 y son en 3D. A continuación se detallan los parámetros de las funciones de aumento de datos:

- Deformaciones elásticas 3D: σ en el rango [3, 5], rango de magnitud [50, 100] y probabilidad 0,15.
- Rotaciones aleatorias en torno a uno o más ejes: rango en x , y y z de 0,3 y probabilidad 0,2. Se emplea modo bilineal para el volumen y vecino más cercano para la máscara.
- Zoom aleatorio: mínimo 0,8 y máximo 1,2, modos trilineal para volumen y vecino más cercano para la máscara, alineando esquinas y con probabilidad 0,16.
- Suavizado gaussiano aleatorio: $\sigma_x = \sigma_y = \sigma_z = [0,5; 1,15]$ con probabilidad 0,15.
- Campos de bias aleatorios: grado 3, rango de coeficiente [0,0; 0,1] y probabilidad 0,15.
- Escalados aleatorios de las intensidades del volumen: factor de escala 0,3 y probabilidad 0,5.
- Desplazamientos aleatorios de las intensidades: factor de desplazamiento 0,3 y probabilidad 0,5.
- Ruido gaussiano aleatorio: desviación estándar 0,1 y probabilidad 0,15.
- Volteos aleatorios: probabilidad 0,5 para flip en cada eje espacial.

En algunas transformaciones, como el escalado o desplazamiento, cuando se especifica un único factor x , se toma aleatoriamente del rango $(-x, x)$.

El tamaño de los subvolúmenes es $96 \times 96 \times 96$ y el tamaño de batch es 2. La duración de los entrenamientos va desde las 400 hasta las 800 épocas (salvo alguna excepción), con un intervalo de validación de 10 épocas. Los entrenamientos que incorporan early stopping lo hacen con una paciencia de

10, es decir, 100 épocas. Hacen uso de la división de datos estratificada por tipo de tumor.

Para la inferencia en validación, se realiza mediante ventana deslizante de tamaño $96 \times 96 \times 96$ con solapamiento de 0,65.

Con el objetivo de acelerar el proceso, se emplea `CacheDataset` de MO-NAI con una tasa de caché de 1 para cargar todos los datos con las transformaciones determinísticas en memoria. Se usa `ThreadDataLoader` como `dataloader` con `shuffle=True` para el de entrenamiento. El resto de configuraciones se especifican en cada experimento.

U-Net

Se proponen tres experimentos usando U-Net:

1. UNet 4M: canales [16, 32, 64, 128, 256], strides [2, 2, 2, 2], 2 unidades residuales y batch normalization. Como función de pérdida se usa una combinación de Dice Loss con Cross-Entropy.
 - Optimizador: SGD con *momentum* 0,9 y decaimiento de pesos 0,00004. Tasa de aprendizaje inicial de 0,025.
 - LR scheduler: StepLR de PyTorch con tamaño de paso 80 y $\gamma = 0,5$. Significa que cada 80 épocas reduce la tasa de aprendizaje multiplicándola por 0,5.
2. UNet 19M: variante de la propuesta anterior (1), con la diferencia de que los canales son [32, 64, 128, 256, 512] y se añade dropout de 0,2.
3. UNet 70M: variante de la anterior (2) en la que sólo se amplían los canales a [64, 128, 256, 512, 1024] y se usa normalización de instancia.

El nombre de estos experimentos procede del número de parámetros entrenables que tiene cada modelo, que al ampliar el número de canales aumenta. La duración de los experimentos es 400, 600 y 600 épocas respectivamente. Solo se emplea early stopping en las variantes (2) y (3). Para la (1) y (2) no se monitoriza la pérdida en validación.

DynUNet

Se propone DynUNet con kernels de tamaño 3, strides [1, 1, 1] para la primera capa, [2, 2, 2] para las sucesivas y [2, 2, 1] para la última. El tamaño de los kernels de upsampling es [2, 2, 2] para todas las capas excepto la última, que será [2, 2, 1] para concordar con los strides. Se establece normalización

de instancia, uso de bloques residuales y dropout 0,2. La función de pérdida es una combinación de Dice Loss con Cross-Entropy.

Como optimizador se opta por AdamW con una tasa de aprendizaje inicial de 0,00005 y un decaimiento de pesos de 0,00001. El LR scheduler en este caso es una combinación secuencial de dos planificadores de PyTorch:

- LinearLR: `end_factor=1.0` y 20 iteraciones totales. Usa un factor multiplicativo que crece linealmente desde `start_factor` (por defecto 0,33) hasta `end_factor=1.0` durante las iteraciones establecidas. Por tanto, la primera época tendrá un learning rate de $\eta = 5,0e^{-5} \cdot 0,33$ y en la época 20 $\eta = 5,0e^{-5} \cdot 1,0$.
- CosineAnnealingWarmRestarts: $T_0 = 100$, $T_{mult} = 1$ y $\eta_{min} = 1e-8$. Tras las primeras 20 épocas en las que actúa el planificador anterior, este scheduler ajusta el learning rate según una función coseno de forma cíclica, reiniciándose tras cada ciclo. La duración del primer ciclo es T_0 épocas, mientras que los siguientes ciclos duran $T_0 \cdot T_{mult}$ épocas. La tasa mínima que se puede alcanzar está definida por η_{min} .

La duración máxima de este experimento son 800 épocas.

Attention U-Net

Arquitectura Attention U-Net con canales [64, 128, 256, 512, 1024], strides [2, 2, 2, 2], tamaño de kernels [3, 3, 3] y dropout de 0,1. Se usa una combinación de Dice Loss y Cross-Entropy Loss como función de pérdida.

El optimizador escogido es AdamW con un learning rate inicial de 0,0001 y weight decay 0,00001. Se usa CosineAnnealingLR como LR scheduler con $T_{max} = 600$ y $\eta_{min} = 1,0e^{-6}$, que ajusta la tasa de aprendizaje siguiendo una curva coseno desde un valor inicial hasta un η_{min} , sin reinicios ciclados como hacía el del experimento anterior.

Duración máxima del entrenamiento: 600 épocas.

SegResNet

Se proponen dos experimentos con esta arquitectura:

1. SegResNet: 32 filtros iniciales, número de bloques de downsampling en cada capa [1, 2, 3, 4] y número de bloques de upsampling por capa [1, 1, 1]. Dropout de 0,2. La función de pérdida es una combinación de Dice Loss con Focal Loss. Se usa AdamW como optimizador con learning rate inicial de 0,0002 y weight decay $1,0e^{-5}$. El scheduler escogido es CosineAnnealingLR con $T_{max} = 600$ y $\eta_{min} = 1,0e^{-6}$.

2. SegResNet NoLRS: igual que el anterior pero sin LR scheduler (de ahí el nombre) y con una tasa de aprendizaje inicial de 0,0001.

La duración máxima de estos experimentos es 600 épocas.

V-Net

Se propone experimentar con la arquitectura V-Net sin modificar ningún parámetro de su configuración (no tiene configuraciones de kernels, bloques, etc), con una combinación de Dice Loss con Cross-Entropy Loss como función de pérdida.

El optimizador es AdamW con $lr = 1,0e^{-3}$ y weight decay $1,0e^{-5}$. El LR scheduler es CosineAnnealingLR con $T_{max} = 600$ y $\eta_{min} = 1,0e^{-6}$.

Duración máxima del entrenamiento: 600 épocas.

UNETR

El experimento con UNETR construye el modelo con los siguientes parámetros: tamaño de imagen $96 \times 96 \times 96$ (debe concordar con el tamaño de los subvolúmenes), número de características 16, dimensión de la capa oculta 768, 12 cabezas para el transformador y bloques tanto convolucionales como residuales, y normalización de instancia. La función de pérdida es una combinación de Dice Loss con Cross-Entropy Loss.

De nuevo el optimizador es AdamW con $lr = 1,0e^{-4}$ y weight decay $1,0e^{-5}$. Se usa un planificador polinomial (PolynomialLR), con $total_iters = 600$ y $power = 0,9$. Este scheduler justa la tasa de aprendizaje según una decadencia polinómica suave a lo largo de las primeras $total_iters$ iteraciones o épocas. Al tener $power < 1$, la caída es más suave al principio y se acelera al final.

La duración máxima establecida para este experimento es 600 épocas.

Swin UNETR

Se proponen dos experimentos con esta arquitectura:

1. Swin UNETR 100: tamaño de características 48, normalización de instancia y activación de checkpoint para acelerar el entrenamiento. Es necesario especificar las dimensiones de la imagen de entrada, $96 \times 96 \times 96$. Se usa una combinación de Dice Loss con Cross-Entropy Loss como función de pérdida. El optimizador es AdamW con $lr = 1,0e^{-4}$ y weight decay $1,0e^{-5}$. Se opta por CosineAnnealingLR como planificador de

learning rate, con $T_{max} = 100$ y $\eta_{min} = 1,0e^{-6}$. Se usa este T_{max} puesto que, de manera excepcional, se ha limitado el número de épocas a 100.

2. Swin UNETR 300: exactamente igual que el anterior pero con $T_{max} = 300$ en el scheduler en lugar de 100. La duración establecida para este entrenamiento es de 300 épocas.

7.3. Resultados

En esta sección se presentan los resultados obtenidos en los experimentos descritos en el apartado anterior. Para cada uno de ellos, los resultados expuestos son en el subconjunto de validación. El modelo con mejor rendimiento en validación será probado con el conjunto de test.

Resultados en validación

Implementación propia

No se ha tenido en cuenta la métrica Hausdorff Distance.

U-Net 2D. Activación de early stopping en la época 69/100.

Cuadro 7.2: Resultados de U-Net 2D

Clase	Dice	IoU	Precision	Recall
Páncreas	0,6637	0,5179	0,7262	0,6447
Tumor	0,1296	0,0883	0,2289	0,4838
Arterias	0,8336	0,7216	0,8497	0,8321
Venas	0,7494	0,6164	0,7985	0,7295
Media	0,5941	0,4861	0,6508	0,6725

DeepLabV3. Activación de early stopping en la época 75/100.

Cuadro 7.3: Resultados de DeepLabV3

Clase	Dice	IoU	Precision	Recall
Páncreas	0,6110	0,5014	0,7022	0,6805
Tumor	0,2150	0,1673	0,4880	0,2681
Arterias	0,7665	0,6726	0,7914	0,8316
Venas	0,6523	0,5495	0,7242	0,7283
Continúa en la siguiente página				

Cuadro 7.3 – continuación

Clase	Dice	IoU	Precision	Recall
Media	0,5612	0,4727	0,6765	0,6271

U-Net 3D. Resultado medio de 5-fold cross-validation.

Cuadro 7.4: Resultados de U-Net 3D

Clase	Dice	IoU	Precision	Recall
Páncreas	0,7086	0,5795	0,7774	0,7160
Tumor	0,2314	0,1634	0,3640	0,2309
Arterias	0,8702	0,7757	0,8980	0,8540
Venas	0,8232	0,7129	0,8690	0,8015
Media	0,6584	0,5579	0,7271	0,6506

Experimentos implementados con MONAI

Para estos experimentos, se añade la métrica HD95. Además, se acompañan las tablas de resultados con las gráficas de las curvas de los valores de la función de pérdida durante el entrenamiento, tanto en el subconjunto de entrenamiento como en el de validación (exceptuando algún modelo, reflejando la evolución en el trabajo), y el valor medio de Dice en validación.

UNet 4M

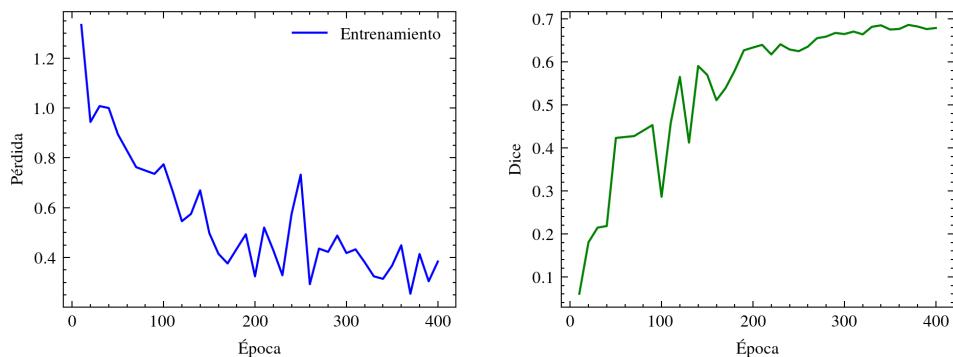


Figura 7.1: UNet 4M. Monitorización del entrenamiento

Cuadro 7.5: Resultados de UNet 4M

Clase	Dice	IoU	Precision	Recall	HD95
Páncreas	0,7816	0,6487	0,8055	0,7773	15,25
Tumor	0,3931	0,2955	0,5647	0,4152	40,40
Arterias	0,8395	0,7252	0,8340	0,8528	14,11
Venas	0,7863	0,6531	0,8191	0,7709	11,68
<i>Media</i>	0,7001	0,5806	0,7641	0,7040	19,41

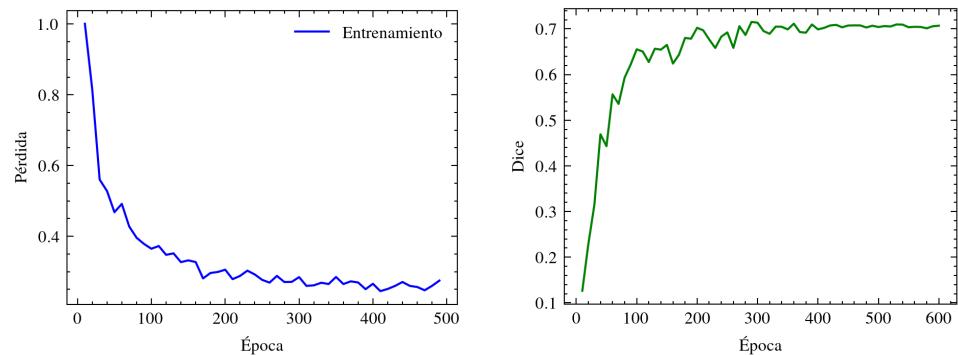
UNet 19M

Figura 7.2: UNet 19M. Monitorización del entrenamiento

Cuadro 7.6: Resultados de UNet 19M

Clase	Dice	IoU	Precision	Recall	HD95
Páncreas	0,7581	0,6174	0,7660	0,7606	13,41
Tumor	0,4535	0,3469	0,6429	0,4451	41,36
Arterias	0,8437	0,7311	0,8607	0,8333	12,39
Venas	0,8061	0,6794	0,8784	0,7535	10,61
<i>Media</i>	0,7153	0,5937	0,7911	0,6981	18,76

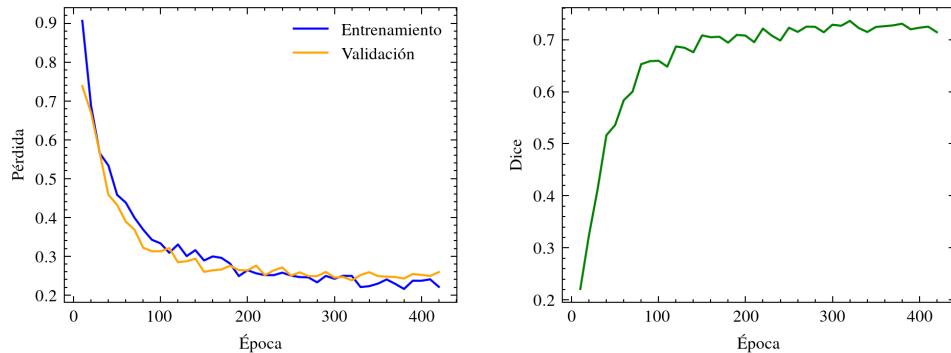
UNet 70M

Figura 7.3: UNet 70M. Monitorización del entrenamiento

Cuadro 7.7: Resultados de UNet 70M

Clase	Dice	IoU	Precision	Recall	HD95
Páncreas	0,7651	0,6257	0,7916	0,7517	13,28
Tumor	0,5130	0,3856	0,6272	0,5254	51,83
Arterias	0,8381	0,7233	0,8185	0,8679	14,72
Venas	0,8316	0,7146	0,8452	0,8246	15,54
Media	0,7370	0,6123	0,7727	0,7424	23,25

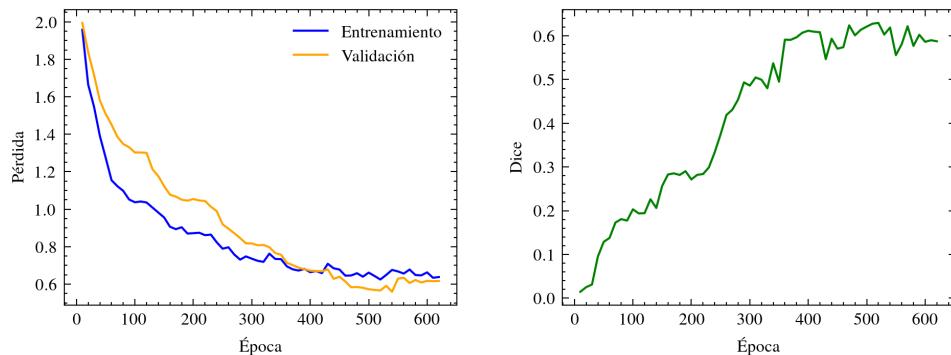
DynUNet

Figura 7.4: DynUNet. Monitorización del entrenamiento

Cuadro 7.8: Resultados de DynUNet

Clase	Dice	IoU	Precision	Recall	HD95
Páncreas	0,6782	0,5224	0,6366	0,7689	56,57
Tumor	0,3721	0,2534	0,3137	0,4920	154,9
Arterias	0,7847	0,6474	0,7132	0,8780	71,28
Venas	0,7030	0,5451	0,6080	0,8440	183,4
Media	0,6345	0,4921	0,5679	0,7457	116,5

Attention U-Net

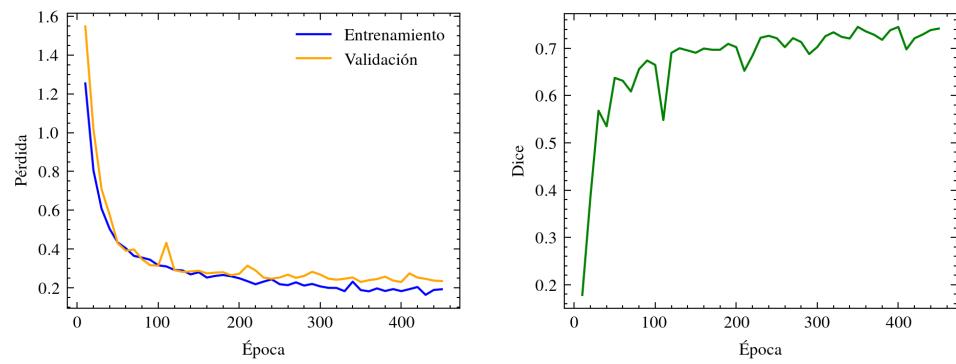


Figura 7.5: Attention U-Net. Monitorización del entrenamiento

Cuadro 7.9: Resultados de Attention U-Net

Clase	Dice	IoU	Precision	Recall	HD95
Páncreas	0,8018	0,6758	0,8295	0,7958	10,70
Tumor	0,5007	0,3778	0,6566	0,4958	29,08
Arterias	0,8504	0,7422	0,8586	0,8518	12,07
Venas	0,8320	0,7173	0,8919	0,7871	10,41
Media	0,7462	0,6283	0,8113	0,7326	15,25

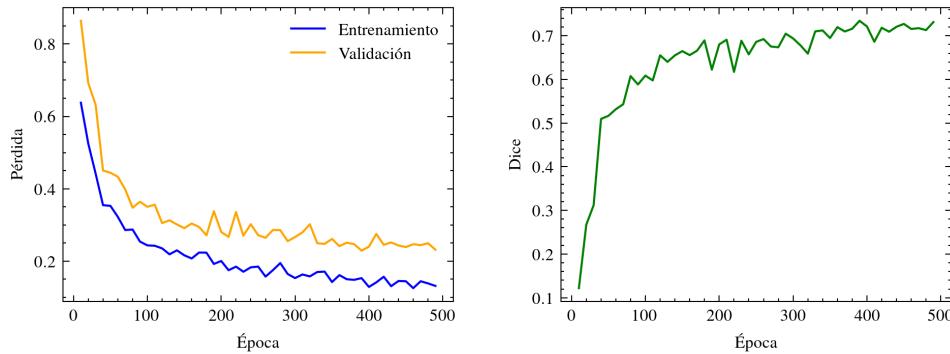
SegResNet

Figura 7.6: SegResNet. Monitorización del entrenamiento

Cuadro 7.10: Resultados de SegResNet

Clase	Dice	IoU	Precision	Recall	HD95
Páncreas	0,7780	0,6501	0,7988	0,7821	14,44
Tumor	0,4868	0,3655	0,5681	0,4874	45,25
Arterias	0,8471	0,7366	0,8344	0,8667	10,25
Venas	0,8302	0,7123	0,8676	0,8012	11,17
Media	0,7356	0,6161	0,7672	0,7344	20,28

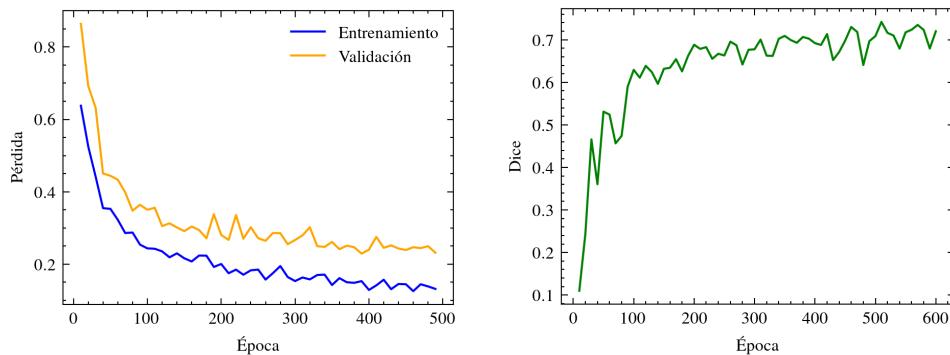
SegResNet NoLRS

Figura 7.7: SegResNet NoLRS. Monitorización del entrenamiento

Cuadro 7.11: Resultados de SegResNet NoLRS

Clase	Dice	IoU	Precision	Recall	HD95
Páncreas	0,7828	0,6497	0,8083	0,7699	14,71
Tumor	0,5155	0,3880	0,5891	0,5358	59,49
Arterias	0,8430	0,7304	0,8499	0,8429	11,04
Venas	0,8286	0,7115	0,8767	0,7922	10,11
Media	0,7425	0,6199	0,7810	0,7352	23,84

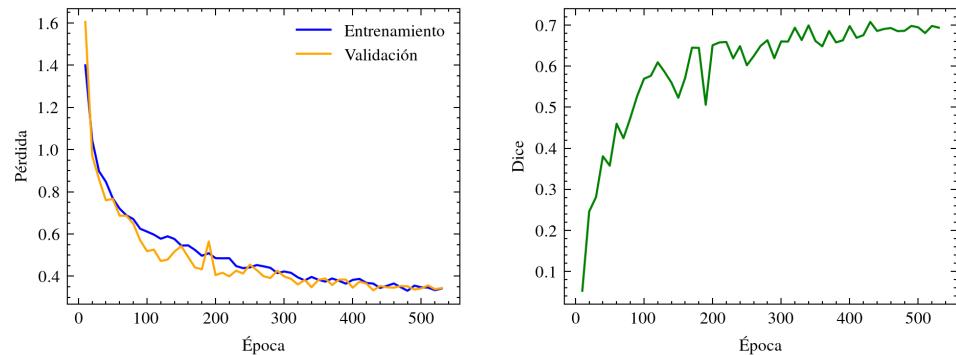
V-Net

Figura 7.8: V-Net. Monitorización del entrenamiento

Cuadro 7.12: Resultados de V-Net

Clase	Dice	IoU	Precision	Recall	HD95
Páncreas	0,7368	0,5947	0,7590	0,7484	20,93
Tumor	0,4584	0,3449	0,6394	0,4536	45,82
Arterias	0,8305	0,7123	0,8382	0,8315	12,00
Venas	0,8087	0,6845	0,8523	0,7824	10,53
Media	0,7086	0,5841	0,7760	0,7040	21,45

UNETR

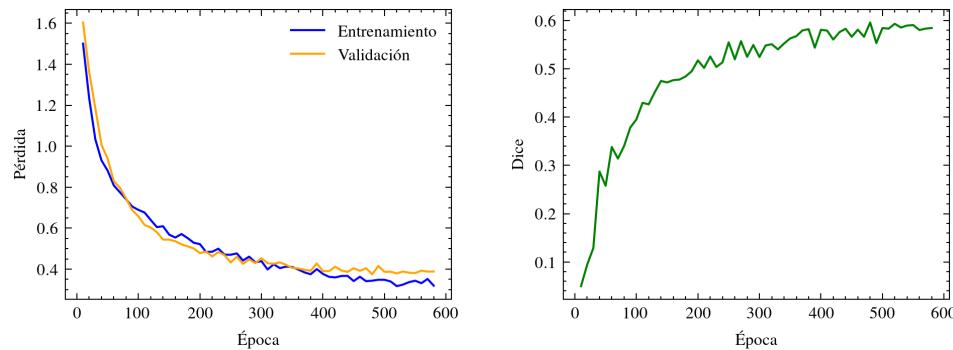


Figura 7.9: UNETR. Monitorización del entrenamiento

Cuadro 7.13: Resultados de UNETR

Clase	Dice	IoU	Precision	Recall	HD95
Páncreas	0,6129	0,4602	0,6842	0,5888	21,53
Tumor	0,2253	0,1580	0,4643	0,1743	55,73
Arterias	0,7974	0,6656	0,8234	0,7818	15,54
Venas	0,7478	0,6028	0,8293	0,6890	18,60
Media	0,5958	0,4717	0,7003	0,5585	27,85

Swin UNETR 100

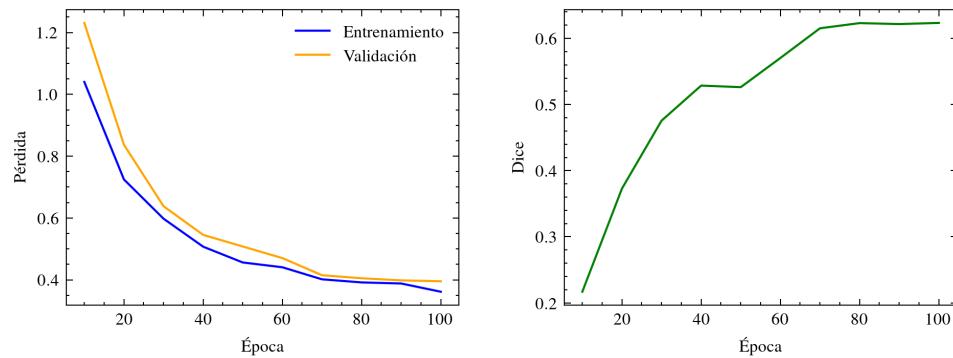


Figura 7.10: Swin UNETR 100. Monitorización del entrenamiento

Cuadro 7.14: Resultados de Swin UNETR 100

Clase	Dice	IoU	Precision	Recall	HD95
Páncreas	0,6244	0,4710	0,7271	0,5829	30,95
Tumor	0,3274	0,2296	0,5966	0,2637	72,21
Arterias	0,8004	0,6695	0,7963	0,8116	49,45
Venas	0,7464	0,6023	0,8016	0,7107	53,57
Media	0,6246	0,4931	0,7323	0,5922	50,87

Swin UNETR 300

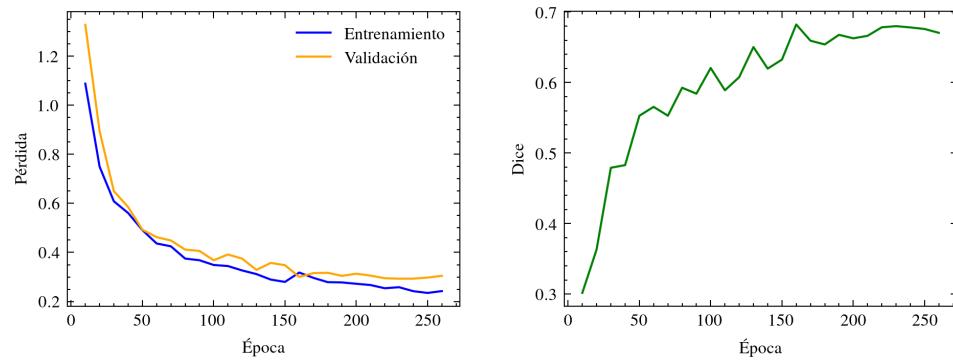


Figura 7.11: Swin UNETR 300. Monitorización del entrenamiento

Cuadro 7.15: Resultados de Swin UNETR 300

Clase	Dice	IoU	Precision	Recall	HD95
Páncreas	0,6581	0,5101	0,8156	0,5831	15,40
Tumor	0,4734	0,3507	0,5848	0,4386	52,76
Arterias	0,8206	0,6988	0,8119	0,8365	15,72
Venas	0,7818	0,6458	0,8060	0,7699	30,76
Media	0,6835	0,5513	0,7570	0,6570	28,11

Resumen y comparativa

Dado que la métrica más importante es el Dice, principalmente porque es la más comprensible y utilizada en el ámbito clínico, a continuación se muestra un cuadro resumen comparando los resultados de todos los entrenamientos con esta métrica.

Cuadro 7.16: Comparativa de Dice (resultados destacables resaltados)

Experimento	Páncreas	Tumor	Arterias	Venas	Media
UNet 4M	0,7816	0,3931	0,8395	0,7863	0,7001
UNet 19M	0,7581	0,4535	0,8437	0,8061	0,7153
UNet 70M	0,7651	0,5130	0,8381	0,8316	0,7370
DynUNet	0,6782	0,3721	0,7874	0,7030	0,6345
Attention U-Net	0,8018	0,5007	0,8504	0,8320	0,7462
SegResNet	0,7780	0,4868	0,8471	0,8302	0,7356
SegResNet No LRS	0,7828	0,5155	0,8430	0,8286	0,7425
V-Net	0,7368	0,4584	0,8305	0,8087	0,7086
UNETR	0,6129	0,2253	0,7974	0,7478	0,5958
Swin UNETR 100	0,6244	0,3274	0,8004	0,7464	0,6246
Swin UNETR 300	0,6581	0,4734	0,8206	0,7818	0,6835

Resultados en test

A continuación, se presentan los resultados del modelo del experimento “SegResNet No LRS” en el conjunto de test (último lote de datos entregado por el hospital, compuesto por 28 pacientes).

Cuadro 7.17: Resultados de “SegResNet No LRS” en test

Clase	Dice	IoU	Precision	Recall	HD95
Páncreas	0,7932	0,6597	0,8536	0,7499	13,46
Tumor	0,4145	0,3106	0,5551	0,4031	54,87
Arterias	0,8133	0,6899	0,9156	0,7442	22,65
Venas	0,7955	0,6632	0,8877	0,7300	18,32
Media	0,7041	0,5809	0,8075	0,6568	26,10

7.4. Análisis de resultados

El análisis de los resultados se estructura en dos fases complementarias: una evaluación cuantitativa basada en las métricas definidas, y una evaluación cualitativa que profundiza en el rendimiento visual de los modelos. Es importante destacar que, tal como se mencionó en la Sección 3.7 y la Sección 6.1, el desarrollo de este proyecto se ha llevado a cabo en dos etapas: una fase inicial de implementación propia y una segunda fase donde se incorporó el framework MONAI. Los experimentos realizados durante la etapa de implementación propia, que incluyeron modelos 2D y un modelo 3D inicial, fueron cruciales para comprender a fondo la naturaleza del problema y

las limitaciones de un enfoque base. Aunque estos experimentos proporcionaron valiosos aprendizajes sobre el flujo de datos y las transformaciones, sus resultados fueron limitados en comparación con el estado del arte y la complejidad del problema, como puede observarse en los Cuadros 7.2, 7.3 y 7.4. Con base en estas observaciones y la necesidad de emplear técnicas más avanzadas para mejorar el rendimiento de los modelos, el análisis detallado de resultados se centrará en los modelos 3D implementados con MONAI, los cuales han demostrado una capacidad superior para capturar las dependencias espaciales y contextuales intercorte esenciales en la segmentación de volúmenes médicos.

Análisis cuantitativo

Rendimiento general de los modelos en validación (Cuadro 7.16)

El Coeficiente de Similitud Dice (DSC) es la métrica más importante y comprensible en el ámbito clínico. En base a ella:

- Los modelos “Attention U-Net” y “SegResNet No LRS” son los que obtienen los mejores resultados generales, con un Dice medio de 0,7462 y 0,7425 respectivamente. Esto sugiere que arquitecturas que incorporan mecanismos de atención (Attention U-Net) o residuales (SegResNet sin planificador de learning rate en este caso) son particularmente efectivas para este problema.
- “UNet 70M” también muestra un rendimiento muy sólido (0,7370), indicando que un mayor número de canales, y por ende de parámetros entrenables, en una arquitectura U-Net puede mejorar el rendimiento.
- Modelos como DynUNet y UNETR muestran un Dice medio inferior (0,6345 y 0,5958 respectivamente). Destacar que UNETR presenta el Dice medio más bajo, lo que podría indicar que su configuración o el enfoque basado en transformadores puro en el codificador no se adapta tan bien al conjunto de datos o requiere un ajuste de hiperparámetros más exhaustivo.

La segmentación del **tumor** es, como se anticipaba en la introducción y el estado del arte, el desafío más significativo debido a sus características. Los valores de Dice para esta clase son notablemente más bajos en comparación con el páncreas, las arterias o las venas. El mejor rendimiento lo obtiene “SegResNet No LRS” con 0,5155, seguido muy de cerca por “UNet 70M” con 0,5130 y “Attention U-Net” con 0,5007. Esto refuerza la dificultad inherente a la detección de lesiones pequeñas y minoritarias en el dataset.

En cuanto al **páncreas, arterias y venas**, los resultados son considerablemente más altos. “Attention U-Net” obtiene el mejor Dice en todas ellas, algo esperado ya que estas estructuras son de mayor tamaño y su contraste con el fondo es más definido o su variabilidad es menor, como ocurre con las arterias y las venas.

La métrica Hausdorff Distance con percentil 95 (HD95), definida como métrica de evaluación de precisión de los límites, también muestra valores más altos para el tumor, lo que indica una mayor discrepancia en la delimitación de sus bordes en comparación con las otras estructuras. Los valores de HD95 son generalmente más bajos, esto es, mejores pues reflejan menor distancia en véxoles entre las predicciones y la ground truth, para páncreas, arterias y venas, lo que concuerda con los mejores resultados de Dice e IoU para estas clases.

Rendimiento del modelo seleccionado en el conjunto de test (Cuadro 7.17)

De los experimentos realizados, se ha seleccionado el mejor modelo en base a las siguientes condiciones, ordenadas de mayor a menor prioridad:

1. Dice e IoU en tumor.
2. Recall en tumor, pues mide cuántos véxoles realmente tumorales han sido segmentados.
3. Precisión en tumor, pues mide cuántos véxoles predichos como tumor son realmente tumor.
4. HD95 en tumor.
5. Rendimiento general de las métricas en el mismo orden.

Aunque la arquitectura Attention U-Net del experimento “Attention U-Net” destaca en páncreas, arterias y venas, la prioridad es el tumor ya que en estas tres estructuras no hay mucha diferencia. Es por ello que, se seleccionará SegResNet (experimento “SegResNet No LRS”) al cumplir los requisitos 1 y 2, y no distar demasiado en el resto de condiciones con las demás propuestas evaluadas.

Por tanto, SegResNet (experimento “SegResNet No LRS”) es el modelo que se evaluará en el conjunto de test (28 pacientes adicionales nunca antes vistos). Esta decisión implica que no se incurre en *data snooping*, que se refiere a la práctica de seleccionar o ajustar modelos e hiperparámetros basándose en el rendimiento observado en un subconjunto de prueba, ya que

estos 28 pacientes adicionales simulan los datos que no se tienen pero se tendrían en un futuro para que los procesara un modelo.

Los resultados en el conjunto de test muestran una ligera disminución en el Dice medio general (0,7041) en comparación con el resultado de validación (0,7425). Esta diferencia es común y podría deberse a la heterogeneidad de los datos del test o a una generalización no perfecta del modelo. La validación cruzada podría haber ofrecido una estimación de rendimiento más robusta en este sentido, aunque la decisión de volver a Hold-Out se justificó por eficiencia computacional y consistencia con otros estudios en el contexto de datasets limitados (véase la Sección 7.1).

Para el **tumor**, el Dice disminuye a 0,4145 en el test, lo que reafirma la dificultad de esta segmentación y la necesidad de continuar investigando en técnicas que aborden el desequilibrio de clases y el pequeño tamaño de la lesión de manera más robusta.

Las métricas de Precision y Recall para el **tumor** (0,5551 y 0,4031 respectivamente) sugieren que, aunque el modelo tiene una tasa aceptable de positivos verdaderos identificados, también clasifica incorrectamente una cantidad significativa de vóxeles como tumor (Falsos Positivos) o no logra detectar todos los tumores (Falsos Negativos), lo cual es coherente con un Dice más bajo.

Para **páncreas**, **arterias** y **venas**, el rendimiento es muy similar en ambos casos, reflejando una buena capacidad de generalización del modelo para estas estructuras.

El análisis cuantitativo demuestra que los modelos de Deep Learning (especialmente Attention U-Net y SegResNet No LRS) son prometedores para la segmentación de estructuras pancreáticas en imágenes de TC, logrando altos coeficientes Dice para el páncreas y los vasos. Sin embargo, la segmentación del tumor sigue siendo un desafío considerable, lo que subraya la importancia de este proyecto.

Análisis cualitativo

Además del análisis cuantitativo presentado en la sección anterior, se ha llevado a cabo una evaluación cualitativa exhaustiva de las segmentaciones obtenidas por los modelos. Este análisis visual es crucial para complementar las métricas numéricas, ya que permite identificar patrones de error, comprender las dificultades inherentes a la tarea de segmentación en imágenes médicas y evaluar la relevancia clínica de los resultados obtenidos. La interpretación visual de las predicciones ofrece una perspectiva sobre el comportamiento de los modelos en diferentes escenarios anatómicos y patológicos, y cómo se alinean con la precisión requerida por el equipo clínico.

Mejor caso

En el mejor caso de prueba, representado en las Figuras 7.12 a 7.15, la predicción del modelo SegResNet, entrenado con el experimento “SegResNet No LRS”, demuestra una calidad de segmentación excepcionalmente alta para el páncreas, las arterias y las venas, con coeficientes Dice de 0,82, 0,87, y 0,80 respectivamente. Incluso para el tumor, que es inherentemente más difícil de segmentar, se obtiene un Dice de 0,81, lo cual es un resultado muy notable dada la complejidad del problema.

Visualmente, la Figura 7.12 muestra una segmentación completa y coherente de todas las estructuras de interés en el volumen TC, muy similar a la ground truth.

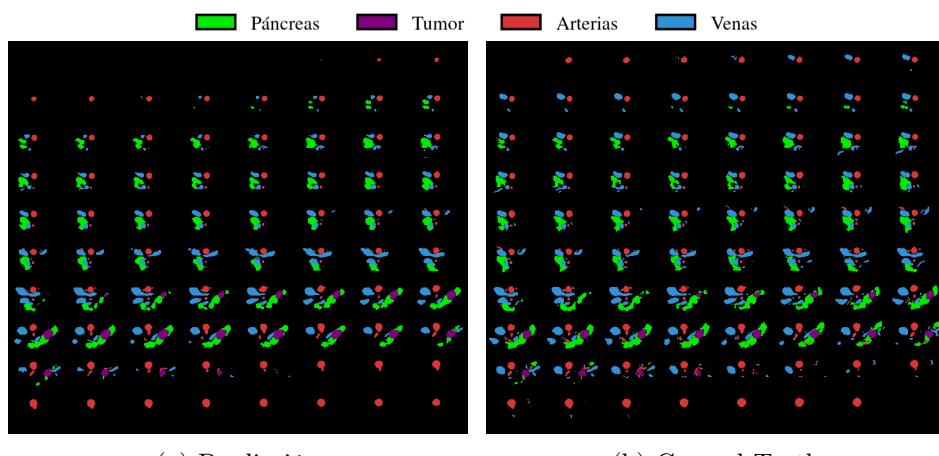


Figura 7.12: Predicción completa del mejor caso en test. *Elaboración propia*

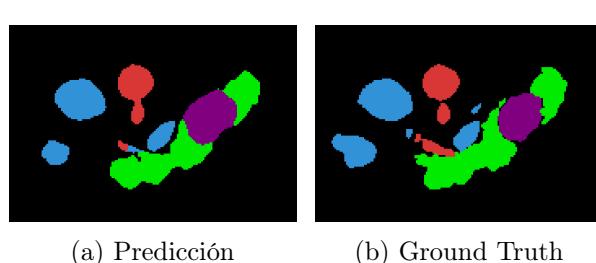


Figura 7.13: Predicción ampliada del mejor caso en test. *Elaboración propia*

Al acercar la imagen en la Figura 7.13, se puede apreciar con mayor detalle la precisión del modelo en la delimitación de los contornos, reflejando una buena capacidad para distinguir el tejido sano del patológico, así como las estructuras vasculares adyacentes. La concordancia es casi perfecta en la

mayoría de las regiones. Sin embargo, la Figura 7.13 también revela ligeras confusiones en los bordes del tumor y el páncreas, y entre venas y arterias. Estas dificultades son esperables de acuerdo con los desafíos mencionados a lo largo del proyecto, tales como límites entre tejidos difusos e intensidades de vértex entre el páncreas sano y el tumoral, así como entre arterias y venas, solapadas.

La Figura 7.14 complementa este análisis mostrando la superposición de las predicciones y las máscaras reales en cortes individuales.

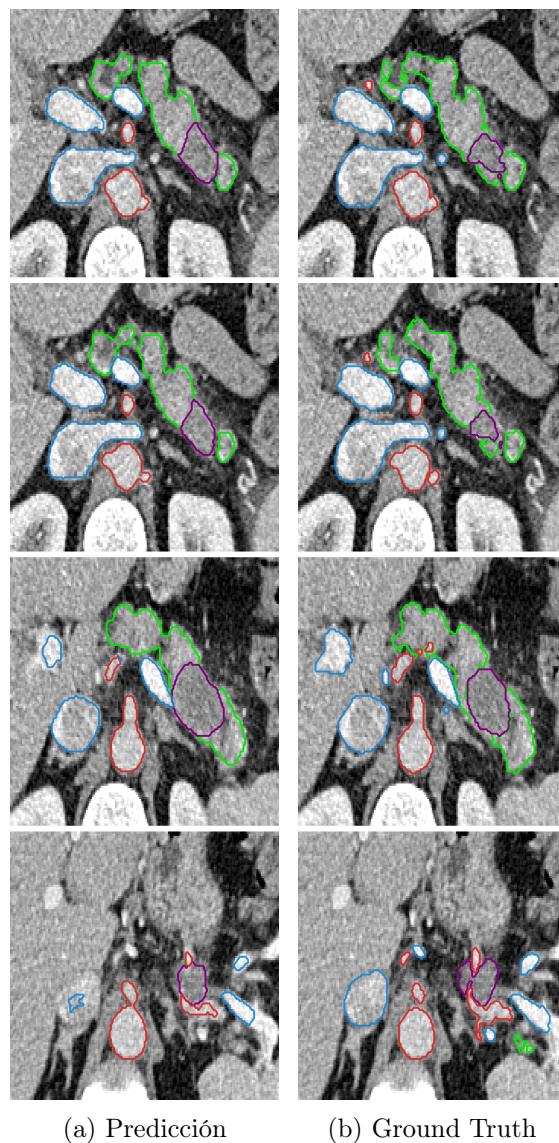


Figura 7.14: Cortes con contornos de predicción y máscara en el mejor caso de test. *Elaboración propia*

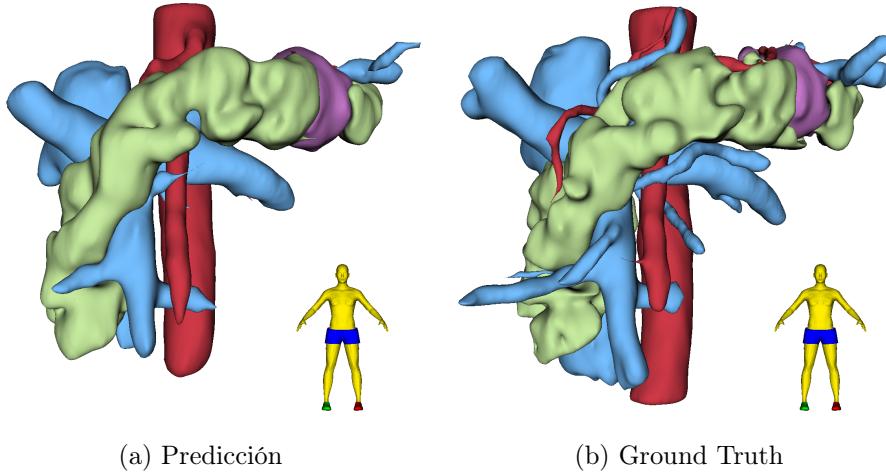


Figura 7.15: Reconstrucción 3D del mejor caso de test. La figura humana indica la orientación. *Elaboración propia* mediante Slicer [6]

La reconstrucción tridimensional en la Figura 7.15 ofrece una visión global de la morfología del páncreas, el tumor y los vasos, y su relación espacial, constituyendo la visualización de mayor importancia clínica. Se puede ver una elevada coincidencia que, exceptuando algunas arterias o venas, convierte la predicción en una muy aceptable en el contexto médico.

Peor caso

El análisis del peor caso, ilustrado en las Figuras 7.16 a 7.19, presenta un contraste significativo, especialmente en la segmentación del tumor, donde el modelo obtiene un coeficiente Dice de 0,00. Esto indica que el modelo ha sido incapaz de detectar o segmentar la lesión tumoral en este paciente en particular.

Las Figuras 7.16 y 7.17 muestran claramente cómo el modelo falla en predecir el tumor, y además, no segmenta completamente el páncreas ni algunas bifurcaciones de los vasos sanguíneos. Las razones detrás de este rendimiento deficiente en el tumor pueden atribuirse a varios desafíos ya identificados en la literatura y en la propia memoria de este proyecto, principalmente el pequeño tamaño de los tumores, lo que produce un fuerte desequilibrio de clases que puede llevar a los modelos a ignorar o segmentar deficientemente estas regiones. Aunque se han aplicado técnicas de data augmentation y funciones de pérdida como Dice Loss para mitigar este desequilibrio, este caso extremo subraya que el problema aún no está resuelto.

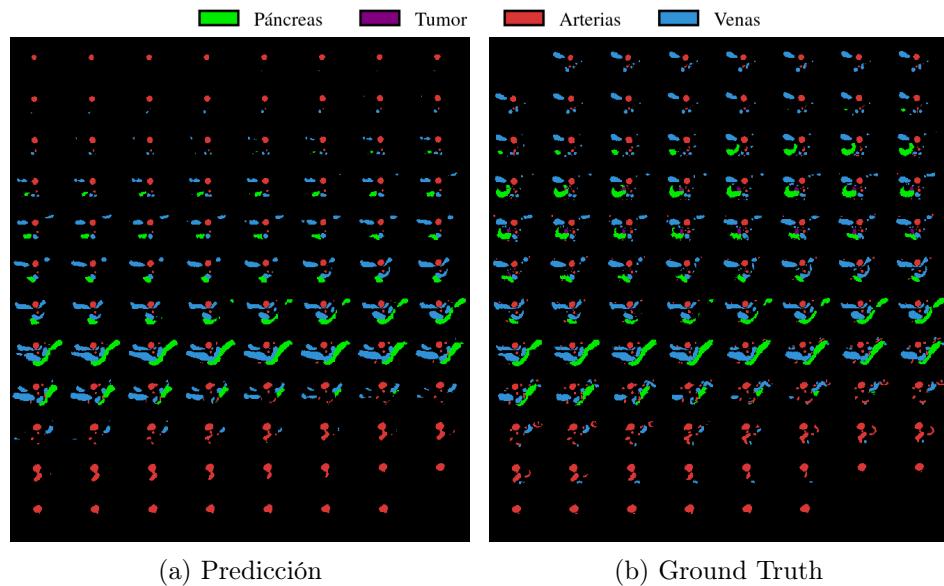


Figura 7.16: Predicción completa del peor caso en test. *Elaboración propia*

Se ha aumentado el tamaño de la Figura 7.17 en comparación con el caso anterior con el objetivo de facilitar la visualización de este ejemplo más complejo.

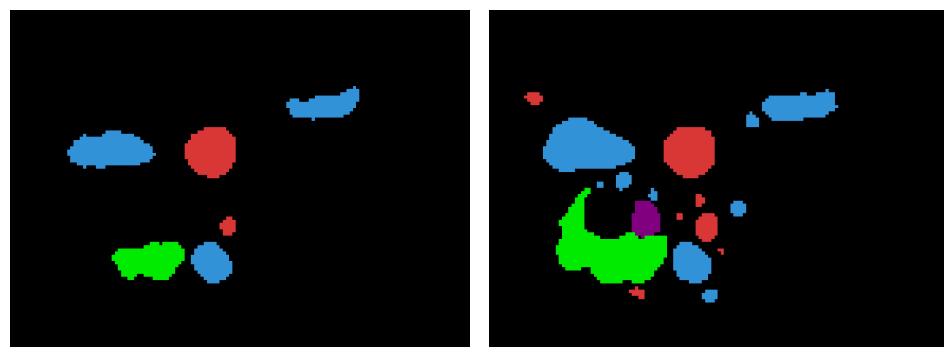


Figura 7.17: Predicción ampliada del peor caso en test. *Elaboración propia*

A pesar de la deficiencia en la segmentación tumoral, el rendimiento del modelo para el páncreas (Dice 0,74), las arterias (Dice 0,84) y las venas (Dice 0,74) sigue siendo decente, lo que sugiere que el modelo aún puede ser útil para la segmentación de estas estructuras más grandes y con mejor contraste. Esto es visualmente corroborado por siguiente figura (Figura 7.18, de nuevo ampliada), donde los contornos de las predicciones para páncreas y vasos se acercan razonablemente a la ground truth.

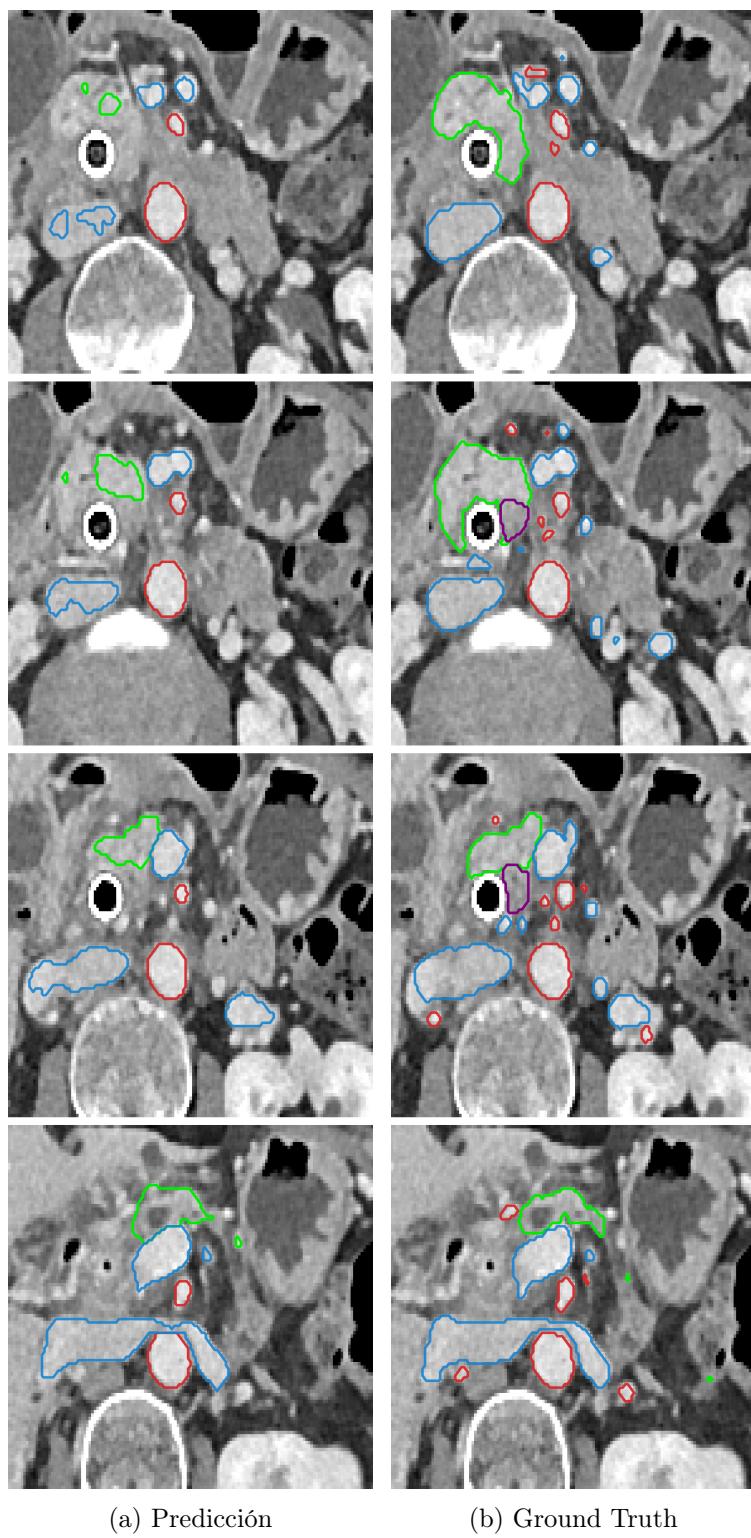


Figura 7.18: Cortes con contornos de predicción y máscara en el peor caso de test. *Elaboración propia*

La reconstrucción 3D del peor caso en la Figura 7.19 visibiliza las imperfecciones en la segmentación del páncreas y el reducido tamaño del tumor, pues se encuentra dentro del páncreas sin ser visible desde el exterior, a diferencia del caso anterior, aunque mantiene una estructura global reconocible para los vasos.

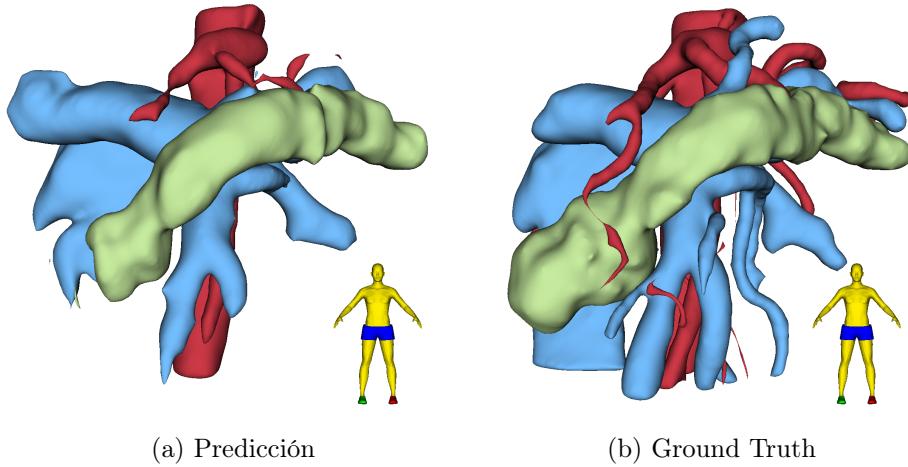


Figura 7.19: Reconstrucción 3D del peor caso de test. La figura humana indica la orientación. *Elaboración propia* mediante Slicer [6]

En el contexto clínico, un modelo como este requeriría la supervisión de un especialista para la interpretación de las segmentaciones tumorales, aunque podría agilizar la visualización de otras estructuras.

Este peor caso refuerza la importancia de la transparencia en la investigación, pues no se pueden presentar simplemente resultados altos en métricas sin atender a la variabilidad ni a las visualizaciones, y la necesidad de abordar las limitaciones del estudio.

Capítulo 8

Conclusiones y trabajos futuros

8.1. Conclusiones

En resumen, este proyecto ha logrado integrar diversas técnicas de Deep Learning, con un enfoque en modelos 3D y el framework MONAI, para abordar el desafiante problema de la segmentación de estructuras pancreáticas. Los resultados, aunque con margen de mejora en la detección de tumores más complejos, validan la viabilidad del enfoque y su relevancia clínica. La experiencia en este TFG ha sido fundamental para profundizar en el manejo de datos médicos, comprender las particularidades de la segmentación 3D y aplicar metodologías de desarrollo rigurosas.

A continuación, se detalla el grado de consecución de cada objetivo planteado:

- **Objetivo General (OG).** Cumplido satisfactoriamente. Se han desarrollado y evaluado modelos de Deep Learning 3D capaces de segmentar con alta precisión el páncreas y los vasos adyacentes. La detección de tumores ha mostrado un rendimiento prometedor, aunque con margen de mejora. La colaboración con el hospital ha sido esencial tanto para el acceso a datos reales como para la validación clínica de los resultados.
- **OE1: 100 %.** El Capítulo 2 de esta memoria recoge una revisión exhaustiva del estado del arte, destacando la falta de estudios que combinen la segmentación simultánea de páncreas, tumores, arterias y venas en modelos 3D aplicables en contextos clínicos.
- **OE2: 100 %.** El Capítulo 4 detalla el análisis del dataset, abordando su variabilidad anatómica y el fuerte desequilibrio de clases. Se han aplicado técnicas avanzadas de preprocesamiento, tanto manuales como automatizadas mediante MONAI, optimizando la preparación de los datos y mejorando el rendimiento de los modelos.

- **OE3:** 90 %. Como se describe en el Capítulo 6, se ha diseñado un marco experimental en dos fases: una inicial, de desarrollo propio, y otra basada en MONAI. Se han explorado múltiples arquitecturas 2D y 3D, aunque hay aspectos a mejorar en robustez y eficiencia.
- **OE4:** 100 %. El Capítulo 7 valida el rendimiento de los modelos mediante métricas estándar (DSC, IoU, Precision, Recall, HD95). Se ha usado validación Hold-Out estratificada, y se ha seleccionado “SegResNet No LRS” como modelo final tras mostrar un resultado ligeramente superior en el tumor. La segmentación de esta patología sigue representando un reto, confirmado tanto por métricas como por análisis visuales, de acuerdo con las observaciones del equipo clínico.

En cuanto a las metas personales, este proyecto me ha permitido adquirir conocimientos fundamentales sobre el cáncer de páncreas y participar en un entorno multidisciplinar. La colaboración con el Hospital Universitario Clínico San Cecilio me ha brindado la oportunidad de contribuir a un problema clínico real, enriqueciendo mi formación más allá del ámbito puramente técnico.

Personalmente, me siento orgulloso de la capacidad de análisis y resolución de problemas que he desarrollado, así como de la mejora en mis habilidades de comunicación técnica al plasmar los resultados en esta memoria. Aunque los resultados obtenidos no son realmente disruptivos y han surgido desafíos en la generalización de los modelos a casos extremos, esta situación ha fortalecido mi capacidad para identificar áreas de mejora, tanto en mi trabajo como en la literatura. Lo considero como una valiosa lección aprendida para futuros proyectos profesionales o académicos, especialmente en este ámbito, donde la informática y, en especial, la Inteligencia Artificial se combinan con la medicina con el único objetivo de mejorar la calidad de vida de las personas.

8.2. Trabajos futuros

A partir de las conclusiones y los desafíos identificados en este TFG, se plantean diversas líneas de trabajo futuras que permitirán refinar los modelos desarrollados, expandir su aplicabilidad y superar las limitaciones actuales. Estas propuestas buscan continuar el camino de innovación y utilidad clínica del proyecto:

- Desarrollo de una arquitectura de Deep Learning propia y optimizada para la segmentación de cáncer de páncreas. A pesar del buen rendimiento de modelos existentes como SegResNet o Attention U-Net, la segmentación de tumores pancreáticos sigue siendo el mayor desafío,

como lo demuestra el peor caso con un Dice de 0,00 para el tumor (Sección 7.4). Implicaría explorar nuevas estrategias para manejar el fuerte desequilibrio de clases y mejorar la diferenciación de límites difusos entre otras.

- Desarrollo de una aplicación con interfaz gráfica (GUI) para radiólogos y cirujanos que simplifique la creación y visualización de modelos 3D. Una vez se disponga de un modelo de segmentación robusto, el siguiente paso crítico es llevar esta tecnología a la práctica clínica. Se puede desarrollar una aplicación que permita:
 - Cargar imágenes TC de pacientes y aplicar automáticamente el modelo de segmentación para obtener las máscaras del páncreas, tumor, arterias y venas.
 - Visualizar de forma interactiva los modelos 3D generados.
 - Facilitar la exportación de los modelos para su uso en impresoras 3D, siguiendo con el objetivo del proyecto DIGIPanCa de generar modelos físicos específicos del paciente.
- Ampliación del conjunto de datos, no solo expandiendo el dataset del hospital, sino también empleando conjuntos públicos como el MSD [27] o NIH [26] para entrenar modelos con más muestras y mejorar el rendimiento y la generalización de los mismos.
- Implementación de modelos de Aprendizaje Federado para aprovechar la información de múltiples instituciones o centros sin necesidad de compartir los datos brutos, y, por tanto, mitigar la escasez de datos cumpliendo las normativas de protección de datos y reduciendo las preocupaciones por la privacidad.

En resumen, este TFG ha sido un paso fundamental en el camino hacia la segmentación automática precisa del cáncer de páncreas. Los futuros trabajos se orientarán no solo a la mejora técnica de los algoritmos de Deep Learning, sino también a su integración práctica en el entorno clínico, ofreciendo herramientas que transformen la planificación quirúrgica.

Bibliografía

- [1] Stanford Medicine Children's Health, "El páncreas," 2025, accessed: Jun. 1, 2025. [Online]. Available: <https://www.stanfordchildrens.org/es/topic/default?id=pancreas-85-P03537>
- [2] HPB & General Surgery, Health Sciences North, "The Pancreas," 2024, accessed: Jun. 1, 2025. [Online]. Available: <https://www.hpbsurgery.ca/the-pancreas>
- [3] César Ramírez, "Cáncer de páncreas borderline resectable: el concepto que marca la evolución de la cirugía oncológica pancreática," [Online]. Available: <http://drcesarramirez.com/cirugia-mbe/162-cáncer-de-páncreas-borderlineresectable-el-concepto-que-marca-la-evolución-de-la-cirugía-oncológicapancreática.html>, dec 2014, accessed: Jun. 1, 2025.
- [4] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan *et al.*, "Microsoft coco: Common objects in context," in *Computer Vision – ECCV 2014*, D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars, Eds. Cham: Springer International Publishing, 2014, pp. 740–755.
- [5] J. Jordan, "An overview of semantic image segmentation," may 2018, accessed: Jun. 2, 2025. [Online]. Available: <https://www.jeremyjordan.me/semantic-segmentation/>
- [6] Andriy Fedorov and Reinhard Beichel and Jayashree Kalpathy-Cramer and Julien Finet and Jean-Christophe Fillion-Robin and Sonia Pujol and Christian Bauer and Dominique Jennings and Fiona Fennessy and Milan Sonka and John Buatti and Stephen Aylward and James V. Miller and Steve Pieper and Ron Kikinis, "Slicer 3D," version 5.8.1. Accessed: Mar. 3, 2025. [Online]. Available: <https://www.slicer.org/>
- [7] S. Oiseth, L. Jones, and E. Maza, "Tomografía computarizada (tc)," Online, feb 2023, 2-6-2025. [Online]. Available: <https://www.lecturio.com/pt/concepts/tomografia-computarizada-tc/>

- [8] S. Wiki, “Coordinate systems — slicer wiki,” 2023, 3-6-2025. [Online]. Available: https://www.slicer.org/w/index.php?title=Coordinate_systems&oldid=64383
- [9] M.-J. Kim, J.-G. Kim, S.-K. Yoon, and S.-D. Kim, “Functionality-based processing-in-memory accelerator for deep convolutional neural networks,” *IEEE Access*, vol. PP, pp. 1–1, 10 2021.
- [10] M. Yani, S. Irawan, and C. Setianingsih, “Application of transfer learning using convolutional neural network method for early detection of terry’s nail,” *Journal of Physics: Conference Series*, vol. 1201, p. 012052, 05 2019.
- [11] Z. Rguibi, A. Hajami, D. Zitouni, A. Elqaraoui, and A. Bedraoui, “Cxai: Explaining convolutional neural networks for medical imaging diagnostic,” *Electronics*, vol. 11, no. 11, 2022. [Online]. Available: <https://www.mdpi.com/2079-9292/11/11/1775>
- [12] O. A. Montesinos López, A. Montesinos López, and J. Crossa, *Artificial Neural Networks and Deep Learning for Genomic Prediction of Continuous Outcomes*. Cham: Springer International Publishing, 2022, pp. 427–476. [Online]. Available: https://doi.org/10.1007/978-3-030-89010-0_11
- [13] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, “Dropout: A simple way to prevent neural networks from overfitting,” *Journal of Machine Learning Research*, vol. 15, no. 56, pp. 1929–1958, 2014. [Online]. Available: <http://jmlr.org/papers/v15/srivastava14a.html>
- [14] K. Akbar, “Instance vs. batch normalization,” <https://www.baeldung.com/cs/instance-vs-batch-normalization>, accessed: Jun. 10, 2025.
- [15] R. Gençay, “Pricing and hedging derivative securities with neural networks: Bayesian regularization, early stopping, and bagging - scientific figure on researchgate,” Disponible en https://www.researchgate.net/figure/Early-stopping-based-on-cross-validation_fig1_3302948, 2021, 12-6-2025.
- [16] O. Ronneberger, P. Fischer, and T. Brox, “U-net: Convolutional networks for biomedical image segmentation,” in *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, N. Navab, J. Hornegger, W. M. Wells, and A. F. Frangi, Eds. Cham: Springer International Publishing, 2015, pp. 234–241.
- [17] Ö. Çiçek, A. Abdulkadir, S. S. Lienkamp, T. Brox, and O. Ronneberger, “3d u-net: Learning dense volumetric segmentation from sparse annotation,” in *Medical Image Computing and Computer-Assisted Intervention*

- *MICCAI 2016*, S. Ourselin, L. Joskowicz, M. R. Sabuncu, G. Unal, and W. Wells, Eds. Cham: Springer International Publishing, 2016, pp. 424–432.
- [18] K. Bai, “A comprehensive introduction to different types of convolutions in deep learning,” Available: <https://medium.com/data-science/a-comprehensive-introduction-to-different-types-of-convolutions-in-deep-learning-669281e58215>, feb 2019, accessed: Jun. 3, 2025.
- [19] O. Oktay, J. Schlemper, L. L. Folgoc, M. Lee, M. Heinrich, K. Misawa *et al.*, “Attention u-net: Learning where to look for the pancreas,” 2018. [Online]. Available: <https://arxiv.org/abs/1804.03999>
- [20] F. Milletari, N. Navab, and S.-A. Ahmadi, “V-net: Fully convolutional neural networks for volumetric medical image segmentation,” 2016. [Online]. Available: <https://arxiv.org/abs/1606.04797>
- [21] A. Myronenko, “3d mri brain tumor segmentation using autoencoder regularization,” 2018. [Online]. Available: <https://arxiv.org/abs/1810.11654>
- [22] A. Hatamizadeh, D. Yang, H. R. Roth, and D. Xu, “Unetr: Transformers for 3d medical image segmentation,” *2022 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pp. 1748–1758, 2021. [Online]. Available: <https://api.semanticscholar.org/CorpusID:232290634>
- [23] A. Hatamizadeh, V. Nath, Y. Tang, D. Yang, H. Roth, and D. Xu, “Swin unetr: Swin transformers for semantic segmentation of brain tumors in mri images,” 2022. [Online]. Available: <https://arxiv.org/abs/2201.01266>
- [24] J. P. Neoptolemos, J. Kleeff, P. Michl, E. Costello, W. Greenhalf, and D. H. Palmer, “Therapeutic developments in pancreatic cancer: current and future perspectives,” *Nat Rev Gastroenterol Hepatol*, vol. 15, no. 6, pp. 333–348, Jun. 2018.
- [25] W. Park, A. Chawla, and E. M. O'Reilly, “Pancreatic cancer: A review,” *JAMA*, vol. 326, no. 9, pp. 851–862, Sep. 2021.
- [26] H. Roth, A. Farag, E. B. Turkbey, L. Lu, J. Liu, and R. M. Summers, “Data from pancreas-ct (version 2),” 2016. [Online]. Available: <https://doi.org/10.7937/K9/TCIA.2016.tNB1kqBU>
- [27] A. L. Simpson, M. Antonelli, S. Bakas, M. Bilello, K. Farahani, B. van Ginneken *et al.*, “A large annotated medical image dataset for the development and evaluation of segmentation algorithms,” 2019. [Online]. Available: <https://arxiv.org/abs/1902.09063>

- [28] M. Oda, N. Shimizu, H. Oda, Y. Hayashi, T. Kitasaka, M. Fujiwara *et al.*, “Towards dense volumetric pancreas segmentation in ct using 3d fully convolutional networks,” in *Medical Imaging 2018: Image Processing*, E. D. Angelini and B. A. Landman, Eds. SPIE, Mar. 2018. [Online]. Available: <http://dx.doi.org/10.1117/12.2293499>
- [29] C. Karri, J. Santinha, N. Papanikolaou, S. K. Gottapu, M. Vuppula, and P. M. K. Prasad, “Pancreatic cancer detection through semantic segmentation of ct images: a short review,” *Discover Artificial Intelligence*, vol. 4, no. 1, p. 101, Dec 2024. [Online]. Available: <https://doi.org/10.1007/s44163-024-00148-x>
- [30] T. Mahmoudi, Z. M. Kouzahkanan, A. R. Radmard, R. Kafieh, A. Salehnia, A. H. Davarpanah *et al.*, “Segmentation of pancreatic ductal adenocarcinoma (pdac) and surrounding vessels in ct images using deep convolutional neural networks and texture descriptors,” *Scientific Reports*, vol. 12, no. 1, p. 3092, Feb 2022. [Online]. Available: <https://doi.org/10.1038/s41598-022-07111-9>
- [31] A. Moglia, M. Cavicchioli, L. Mainardi, and P. Cerveri, “Deep learning for pancreas segmentation on computed tomography: a systematic review,” *Artificial Intelligence Review*, vol. 58, no. 8, p. 220, May 2025. [Online]. Available: <https://doi.org/10.1007/s10462-024-11050-4>
- [32] N. Zhao, N. Tong, D. Ruan, and K. Sheng, “Fully automated pancreas segmentation with two-stage 3d convolutional neural networks,” in *Medical Image Computing and Computer Assisted Intervention – MICCAI 2019*, D. Shen, T. Liu, T. M. Peters, L. H. Staib, C. Essert, S. Zhou *et al.*, Eds. Cham: Springer International Publishing, 2019, pp. 201–209.
- [33] J. Juwita, G. M. Hassan, and A. Datta, “Pancreas segmentation in ct scans: A novel momunet based workflow,” *Computers in Biology and Medicine*, vol. 193, p. 110346, 2025. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0010482525006973>
- [34] S.-H. Lim, Y. J. Kim, Y.-H. Park, D. Kim, K. G. Kim, and D.-H. Lee, “Automated pancreas segmentation and volumetry using deep neural network on computed tomography,” *Scientific Reports*, vol. 12, no. 1, p. 4075, Mar 2022. [Online]. Available: <https://doi.org/10.1038/s41598-022-07848-3>
- [35] J. Ma, Y. Zhang, S. Gu, C. Zhu, C. Ge, Y. Zhang *et al.*, “Abdomenct-1k: Is abdominal organ segmentation a solved problem?” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 10, pp. 6695–6714, 2022.

- [36] J. Becktepe, L. Hennig, S. Oeltze-Jafra, and M. Lindauer, “Auto-nnunet: Towards automated medical image segmentation,” 2025. [Online]. Available: <https://arxiv.org/abs/2505.16561>
- [37] Q. Li, X. Liu, Y. He, D. Li, and J. Xue, “Temperature guided network for 3d joint segmentation of the pancreas and tumors,” *Neural Networks*, vol. 157, pp. 387–403, 2023. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0893608022004221>
- [38] MONAI Team, “MONAI Model Zoo. Pancreas CT DiNTS Segmentation,” https://monai.io/model-zoo.html#/model/pancreas_ct_dints_segmentation, 2025, version 0.5.1. Accessed: Jun. 4, 2025.
- [39] Y. He, D. Yang, H. Roth, C. Zhao, and D. Xu, “DiNTS: Differentiable neural network topology search for 3d medical image segmentation,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021, pp. 5841–5850.
- [40] L. Zhao, T. Wang, Y. Chen, X. Zhang, H. Tang, F. Lin *et al.*, “A novel framework for segmentation of small targets in medical images,” *Scientific Reports*, vol. 15, no. 1, p. 9924, Mar 2025. [Online]. Available: <https://doi.org/10.1038/s41598-025-94437-9>
- [41] L. Grenacher, M. Thorn, H. P. Knaebel, M. Vetter, P. Hassenpflug, T. Kraus *et al.*, “[the role of 3-D imaging and computer-based postprocessing for surgery of the liver and pancreas],” *Rofo*, vol. 177, no. 9, pp. 1219–1226, Sep. 2005.
- [42] M. Klauss, M. Schöbinger, I. Wolf, J. Werner, H.-P. Meinzer, H.-U. Kauczor *et al.*, “Value of three-dimensional reconstructions in pancreatic carcinoma using multidetector CT: initial results,” *World J Gastroenterol*, vol. 15, no. 46, pp. 5827–5832, Dec. 2009.
- [43] M. Oda, H. R. Roth, T. Kitasaka, K. Misawa, M. Fujiwara, and K. Mori, “Abdominal artery segmentation method from ct volumes using fully convolutional neural network,” *International Journal of Computer Assisted Radiology and Surgery*, vol. 14, no. 12, pp. 2069–2081, Dec 2019. [Online]. Available: <https://doi.org/10.1007/s11548-019-02062-5>
- [44] S. Nikolov, S. Blackwell, A. Zverovitch, R. Mendes, M. Livne, J. De Fauw *et al.*, “Clinically applicable segmentation of head and neck anatomy for radiotherapy: Deep learning algorithm development and validation study,” *J Med Internet Res*, vol. 23, no. 7, p. e26151, Jul 2021. [Online]. Available: <https://www.jmir.org/2021/7/e26151>
- [45] J. T. Guibas, T. S. Virdi, and P. S. Li, “Synthetic medical images from dual generative adversarial networks,” 2018. [Online]. Available: <https://arxiv.org/abs/1709.01872>

- [46] O. Diaz, K. Kushibar, R. Osuala, A. Linardos, L. Garrucho, L. Igual *et al.*, “Data preparation for artificial intelligence in medical imaging: A comprehensive guide to open-access platforms and tools,” *Physica Medica*, vol. 83, pp. 25–37, 2021. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S1120179721000958>
- [47] T. B. Russell and S. Aroori, “The pancreas from a surgical perspective: an illustrated overview,” *Art of Surgery*, vol. 6, no. 0, 2022. [Online]. Available: <https://aos.amegroups.org/article/view/7309>
- [48] S. Sánchez Cabús and L. Fernández-Cruz, “Cirugía del cáncer de páncreas: estrategias quirúrgicas según los datos basados en la evidencia,” *Cirugía Española*, vol. 93, no. 7, pp. 423–435, 2015. [Online]. Available: <https://doi.org/10.1016/j.ciresp.2015.03.009>
- [49] S. Navarro Colás and E. C. Vaquero Raya, “Estrategia ante una neoplasia mucinosa papilar intraductal de páncreas,” *Gastroenterología y Hepatología*, vol. 35, no. 9, pp. 652–659, 2012. [Online]. Available: <https://doi.org/10.1016/j.gastrohep.2012.05.009>
- [50] S. Bakas, M. Reyes, A. Jakab, S. Bauer, M. Rempfler, A. Crimi *et al.*, “Identifying the best machine learning algorithms for brain tumor segmentation, progression assessment, and overall survival prediction in the brats challenge,” 2019. [Online]. Available: <https://arxiv.org/abs/1811.02629>
- [51] A. Calzado and J. Geleijns, “Tomografía computarizada. Evolución, principios técnicos y aplicaciones,” *Revista de Física Médica*, vol. 11, no. 3, Dec. 2010. [Online]. Available: <https://revistadefisicamedica.es/index.php/rfm/article/view/115>
- [52] A. Fedorov, R. Beichel, J. Kalpathy-Cramer, J. Finet, J.-C. Fillion-Robin, S. Pujol *et al.*, “3D Slicer as an image computing platform for the Quantitative Imaging Network,” *Magnetic Resonance Imaging*, vol. 30, no. 9, pp. 1323–1341, 2012, quantitative Imaging in Cancer. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0730725X12001816>
- [53] P. Team, “PyTorch documentation.” [Online]. Available: https://www.researchgate.net/figure/Early-stopping-based-on-cross-validation_fig1_3302948
- [54] M. J. Cardoso, W. Li, R. Brown, N. Ma, E. Kerfoot, Y. Wang *et al.*, “MONAI: An open-source framework for deep learning in healthcare,” *arXiv:2211.02701*, 2022.

- [55] Y. Huang, J. Wen, Y. Wang, J. Hu, Y. Wang, and W. Yang, “Subset selection strategy-based pancreas segmentation in ct,” *Quantitative Imaging in Medicine and Surgery*, vol. 12, no. 6, 2022. [Online]. Available: <https://qims.amegroups.org/article/view/92267>
- [56] J. Colman, L. Zhang, W. Duan, and X. Ye, “Dr-unet104 for multimodal mri brain tumor segmentation,” in *Brainlesion: Glioma, Multiple Sclerosis, Stroke and Traumatic Brain Injuries*, A. Crimi and S. Bakas, Eds. Cham: Springer International Publishing, 2021, pp. 410–419.
- [57] S. N. Llambias, M. Nielsen, and M. M. Ghazi, “Data augmentation-based unsupervised domain adaptation in medical imaging,” 2023. [Online]. Available: <https://arxiv.org/abs/2308.04395>
- [58] I. D. Mienye, T. G. Swart, G. Obaido, M. Jordan, and P. Ilono, “Deep convolutional neural networks: A comprehensive review,” *Preprints*, August 2024. [Online]. Available: <https://doi.org/10.20944/preprints202408.1288.v1>
- [59] M. Yeung, L. Rundo, Y. Nan, E. Sala, C.-B. Schönlieb, and G. Yang, “Calibrating the dice loss to handle neural network overconfidence for biomedical image segmentation,” *Journal of Digital Imaging*, vol. 36, no. 2, pp. 739–752, Apr 2023. [Online]. Available: <https://doi.org/10.1007/s10278-022-00735-3>
- [60] J. Terven, D.-M. Cordova-Esparza, J.-A. Romero-González, A. Ramírez-Pedraza, and E. A. Chávez-Urbiola, “A comprehensive survey of loss functions and metrics in deep learning,” *Artificial Intelligence Review*, vol. 58, no. 7, p. 195, Apr 2025. [Online]. Available: <https://doi.org/10.1007/s10462-025-11198-7>
- [61] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, “Focal loss for dense object detection,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 42, no. 2, pp. 318–327, 2020.
- [62] N. Ketkar and J. Moolayil, *Deep learning with Python : learn best practices of deep learning models with PyTorch*, 2nd ed. Place of publication not identified: Apress, 2021.
- [63] X. Liu, L. Song, S. Liu, and Y. Zhang, “A review of deep-learning-based medical image segmentation methods,” *Sustainability*, vol. 13, no. 3, 2021. [Online]. Available: <https://www.mdpi.com/2071-1050/13/3/1224>
- [64] L.-C. Chen, G. Papandreou, F. Schroff, and H. Adam, “Rethinking atrous convolution for semantic image segmentation,” 2017. [Online]. Available: <https://arxiv.org/abs/1706.05587>

- [65] U. Kamal, A. M. Rafi, R. Hoque, J. Wu, and M. K. Hasan, “Lung cancer tumor region segmentation using recurrent 3d-denseunet,” in *Thoracic Image Analysis*, J. Petersen, R. San José Estépar, A. Schmidt-Richberg, S. Gerard, B. Lassen-Schmidt, C. Jacobs *et al.*, Eds. Cham: Springer International Publishing, 2020, pp. 36–47.
- [66] F. Isensee, J. Petersen, A. Klein, D. Zimmerer, P. F. Jaeger, S. Kohl *et al.*, “nnu-net: Self-adapting framework for u-net-based medical image segmentation,” 2018. [Online]. Available: <https://arxiv.org/abs/1809.10486>
- [67] F. Isensee, P. F. Jaeger, S. A. A. Kohl, J. Petersen, and K. H. Maier-Hein, “nnu-net: a self-configuring method for deep learning-based biomedical image segmentation,” *Nature Methods*, vol. 18, no. 2, p. 203–211, Dec. 2020. [Online]. Available: <http://dx.doi.org/10.1038/s41592-020-01008-z>
- [68] M. Futrega, A. Milesi, M. Marcinkiewicz, and P. Ribalta, “Optimized u-net for brain tumor segmentation,” 2021. [Online]. Available: <https://arxiv.org/abs/2110.03352>
- [69] S. Tilborghs, J. Bertels, D. Robben, D. Vandermeulen, and F. Maes, “The dice loss in the context of missing or empty labels: Introducing ϕ and ϵ ,” in *Medical Image Computing and Computer Assisted Intervention – MICCAI 2022*, L. Wang, Q. Dou, P. T. Fletcher, S. Speidel, and S. Li, Eds. Cham: Springer Nature Switzerland, 2022, pp. 527–537.
- [70] E. Somasundaram, Z. Taylor, V. V. Alves, L. Qiu, B. L. Fortson, N. Mahalingam *et al.*, “Deep learning models for abdominal ct organ segmentation in children: Development and validation in internal and heterogeneous public datasets,” *American Journal of Roentgenology*, vol. 223, no. 1, p. e2430931, 2024, PMID: 38691411. [Online]. Available: <https://doi.org/10.2214/AJR.24.30931>

Apéndice A

Hoja de resultados de Google Sheets

En la siguiente página se puede ver una muestra de la estructura y contenido de la hoja de cálculo utilizada por el notificador para registrar los resultados de los experimentos.

experiment	configuration										epochs	final loss	final global dice	best	best dice	best dice coefficients per class	training_time	date	directory					
	config	model	bs	lr	optim	loss_fn	completed	max	best	train_loss														
deep_aug	deep_aug	deeplabv3	4	0.0001	AdamW	CombinedLoss	21	100	6	2.04803	0.84151	0.65550	0.737812	0.64495	0.59953	0.09850	0.74262	9:27:17.867	16-03-2025 21:51:48	17-03-2025 7:31:37				
deep_aug	deep_aug	deeplabv3	4	0.0001	AdamW	CombinedLoss	34	100	19	1.66079	2.43079	0.85608	0.67258	0.234847	0.67205	0.90398	0.65599	0.10728	0.83651	0.76648	10:24:39.307	16-03-2025 6:55:34	18-03-2025 7:51:07	
deep_aug_randrop	deep_aug_randrop	deeplabv3	4	0.0001	AdamW	CombinedLoss	39	100	24	2.19933	2.41968	0.76856	0.65899	0.23160	0.67058	0.90344	0.67227	0.11424	0.82780	0.74517	8:00:10.025	18-03-2025 20:45:05	19-03-2025 4:50:26	
experiments/unet_mona	experiment_unet_mona	unet	4	0.0001	AdamW	MulticlassDiceLoss	69	100	54	0.30699	0.41482	0.75220	0.66675	0.40599	0.67403	0.99382	0.66374	0.12955	0.83386	0.74938	1:33:01.515	19-03-2025 8:26:35	19-03-2025 10:04:28	
deep_aug_randrop	deep_aug_randrop	deeplabv3	4	0.0001	AdamW	CombinedLoss	36	100	21	2.21993	2.40901	0.75989	0.65300	0.22879	0.67188	0.90305	0.65370	0.13289	0.83046	0.74904	7:20:29.469	19-03-2025 7:44:30	19-03-2025 15:10:40	
ug_randrop_no_bord	deep_aug_randrop_no_bord	deeplabv3	4	0.0001	AdamW	CombinedLoss	62	100	47	2.01957	2.23264	0.78613	0.67835	0.67854	0.98925	0.67982	0.12112	0.82486	0.75564	9:07:58.567	19-03-2025 7:44:02	19-03-2025 16:59:06		
deep_aug	deep_aug	deeplabv3	4	0.0001	AdamW	CombinedLoss	24	100	9	1.69617	2.42048	0.84661	0.68132	0.23664	0.66992	0.90342	0.63562	0.10984	0.82449	0.79369	20:13:58.182	20-03-2025 1:52:14	21-03-2025 8:06:24	
deep_aug_no_bord	deep_aug_no_bord	deeplabv3	4	0.0001	AdamW	CombinedLoss	31	100	16	2.20145	2.29584	0.79141	0.70084	0.67144	0.98919	0.68189	0.12089	0.82790	0.79540	22:00:00.000	20-03-2025 11:59:59	21-03-2025 12:01:15		
ug_randrop_no_bord	deep_aug_randrop_no_bord	deeplabv3	4	0.0001	AdamW	FocalLoss	29	100	14	0.00426	0.02459	0.67362	0.50765	0.03065	0.55672	0.98841	0.47769	0.05992	0.63110	0.58848	5:51:04.935	22-03-2025 14:57:51	22-03-2025 21:45:38	
deep_aug_randrop_2	deep_aug_randrop_2	deeplabv3	4	0.0001	AdamW	FocalLoss	37	100	22	0.00166	0.01233	0.68105	0.59505	0.00112	0.51517	0.99701	0.50982	0.05570	0.60570	0.58761	8:20:45.261	22-03-2025 15:54:31	23-03-2025 20:29:39	
ug_randrop_no_borddeep	ug_randrop_no_bord	deeplabv3	4	0.0001	AdamW	FocalLoss	20	100	5	0.00114	0.00453	0.78191	0.61677	0.00404	0.52198	0.98607	0.46386	0.05522	0.57621	0.52859	11:14:58.330	22-03-2025 15:47:27	23-03-2025 3:09:33	
deep_aug	deep_aug_2	deeplabv3	4	0.0001	AdamW	WeightedDiceLoss	18	100	3	0.60466	0.60678	0.26862	0.24163	0.60386	0.19820	0.90095	0.00003	0.00002	0.00008	0.00008	3:37:39.969	23-03-2025 11:34:09	23-03-2025 15:21:38	
ug_randrop_no_borddeep	ug_randrop_no_bord	deeplabv3	4	0.0001	AdamW	WeightedDiceLoss	43	200	18	0.62928	0.61500	0.19749	0.19919	0.59571	0.20000	0.90162	0.00002	0.00003	0.00007	0.00007	0:57:00.647	23-03-2025 16:20:12	23-03-2025 16:20:12	
experiment/unet_mona	experiment_unet_mona_3	net(MONAI)	4	0.0001	AdamW	WeightedDiceLoss	31	100	19	0.62024	0.60565	0.19749	0.19919	0.59571	0.20000	0.90162	0.00002	0.00003	0.00007	0.00007	0:57:20.066	23-03-2025 11:39:45	23-03-2025 11:39:45	
deep_aug_unet_mona	deep_aug_unet_mona_4	net(MONAI)	4	0.0001	AdamW	DiceFocalLoss	138	200	113	0.39265	0.42934	0.79653	0.70443	0.42113	0.70984	0.98987	0.70922	0.22814	0.82790	0.73834	4:42:21.317	23-03-2025 16:39:59	23-03-2025 23:28:18	
ug_randrop_no_bord	ug_randrop_no_bord	deeplabv3	4	0.0001	AdamW	DiceFocalLoss	46	100	31	0.23671	0.23732	0.75016	0.02762	0.24734	0.70993	0.98837	0.64893	0.36582	0.77652	0.71582	11:16:14.812	23-03-2025 18:22:23	24-03-2025 5:47:02	
deep_aug	deep_aug_3	deeplabv3	4	0.0001	AdamW	DiceFocalLoss	57	100	42	0.10263	0.27188	0.89330	0.70985	0.25643	0.78183	0.99504	0.70308	0.32450	0.81405	0.75543	1 day, 15:41:39.940	23-03-2025 18:22:11	25-03-2025 10:12:04	
experiment/unet_mona_5	experiment_unet_mona_5	net(MONAI)	4	0.0001	AdamW	DiceFocalLoss	97	200	72	0.47627	0.44062	0.74946	0.69521	0.43736	0.90421	0.67926	0.24373	0.82171	0.76789	1:24:36.241	25-03-2025 20:44:16	25-03-2025 22:14:58		
deep_aug	deep_aug_5	deeplabv3	4	0.0001	AdamW	DiceFocalLoss	60	100	50	0.13256	0.13005	0.86005	0.70746	0.25605	0.79132	0.98954	0.70956	0.27000	0.82350	0.77211	1:26:40.246	25-03-2025 20:44:06	25-03-2025 23:38:18	
unet3d	unet3d_1	unet3d	2	0.00005	AdamW	DiceFocalLoss	88	200	50	0.00006	0.00000	0.22690	0.23309	0.00000	0.00000	0.9941	0.22990	0.00000	0.00000	0.00000	0.00000	0:57:00.717	26-03-2025 10:33:39	26-03-2025 23:38:18
deep_aug	deep_aug_5	deeplabv3	4	0.0001	AdamW	DiceFocalLoss	47	100	32	0.15159	0.26489	0.61293	0.53556	0.26277	0.53988	0.99390	0.41528	0.04354	0.69630	0.00016	9:17:20.312	30-03-2025 15:16:15	31-03-2025 0:42:25	
unet3d_mona	unet3d_2	unet3d(MONAI)	4	0.00005	AdamW	DiceFocalLoss	94	200	74	0.08535	0.31253	0.20691	0.11712	0.29862	0.73328	0.99497	0.74451	0.20542	0.83101	4:39:07.537	03-04-2025 16:39:43	03-04-2025 21:19:12		
unet3d	unet3d_1	unet3d	2	0.00005	AdamW	DiceFocalLoss	110	200	90	0.08530	0.27620	0.62132	0.70404	0.21618	0.75409	0.99550	0.76845	0.20240	0.86544	0.63147	0:43:41.427	03-04-2025 17:08:24	03-04-2025 23:52:18	
deep_aug	deep_aug_6	deeplabv3	16	0.0001	AdamW	DiceFocalLoss	85	100	70	0.12991	0.26402	0.60362	0.54781	0.23247	0.56277	0.99451	0.46196	0.06022	0.71367	0.57748	18:13:02.207	03-04-2025 16:09:39	04-04-2025 10:20:13	
unet3d_mona	unet3d_mona_3	net(MONAI)	4	0.0001	AdamW	DiceCELoss	68	200	48	0.00479	0.00433	0.45443	0.46705	0.73749	0.73749	0.99577	0.00007	0.00008	0.00005	0.00005	3:29:40.000	03-04-2025 15:31:26	03-04-2025 23:52:18	
deep_aug	deep_aug_5	deeplabv3	4	0.0001	AdamW	DiceFocalLoss	83	100	68	0.12442	0.20544	0.64016	0.15181	0.23262	0.55716	0.99405	0.46087	0.03782	0.71522	0.57756	8:20:38.132	07-04-2025 20:24:22	08-04-2025 4:45:35	
deep_aug	deep_aug_5	deeplabv3	4	0.0001	AdamW	DiceFocalLoss	75	100	60	0.12026	0.24925	0.61634	0.64063	0.24339	0.64784	0.99432	0.61098	0.21509	0.76652	0.62320	6:44:22.771	15-04-2025 21:58:56	24-04-2025 4:43:54	
unet3d	unet3d_4	unet3d	2	0.00005	AdamW	DiceFocalLoss	90	200	70,4	0.10384	0.37787	0.59295	0.36994	0.36058	0.68384	N/A	0.70898	0.23136	0.87022	0.68231	1 day, 2:10:08.107	23-04-2025 8:51:00	24-04-2025 6:40:27	
deep_aug	deep_aug_9	deeplabv3	8	0.0001	AdamW	DiceCELoss	36	100	2	0.21067	0.04748	0.74917	0.51296	0.44177	0.52896	N/A	0.57284	0.14867	0.73777	0.69295	2:33:52.633	29-04-2025 17:35:00	01-05-2025 21:28:27	
unet3d	unet3d_5	unet3d	2	0.00005	AdamW	DiceFocalLoss	54	200	39,2	0.19267	0.35780	0.77738	0.56306	0.34384	0.58089	N/A	0.62883	0.13417	0.84084	0.72007	12 days, 13:30:28.4	01-05-2025 10:56:43	09-05-2025 0:32:29	

Figura A.1: Resultados en Google Sheet de experimentos propios a fecha del 9 de mayo de 2025. *Elaboración propia*

