



ULPGC
Universidad de
Las Palmas de
Gran Canaria

eii

ESCUELA DE
INGENIERÍA INFORMÁTICA

Trabajo de Fin de Grado

Segmentación basada en U-Net y xLSTM

Grado en Ciencia e Ingeniería de Datos

AUTOR: Susana Suárez Mendoza

TUTORIZADO POR:

Francisco Mario Hernández Tejera

Miguel Ángel Medina Ramírez

Junio 2025

SOLICITUD DE DEFENSA DE TRABAJO DE FIN DE TÍTULO

D./D^a Susana Suárez Mendoza, autor del trabajo Segmentación basada en U-Net y xLSTM, correspondiente a la titulación Grado en Ciencia e Ingeniería de Datos.

SOLICITA

que se inicie el procedimiento de defensa del mismo, para lo que se adjunta la documentación requerida, haciendo constar que

[X] se autoriza / [] no se autoriza la grabación en audio de la exposición y turno de preguntas.

Asimismo, con respecto al registro de la propiedad intelectual/industrial del TFT, declara que:

[] Se ha iniciado o hay intención de iniciarlo (defensa no pública).

[X] No está previsto.

Y para que así conste firma la presente. (fecha en firma electrónica)

El/La estudiante

Fdo. _____

A rellenar y firmar **obligatoriamente** por el/la/los/las tutores

En relación con la presente solicitud, se informa: (firmar donde corresponda)

Positivamente (en caso de detección de copia, esta firma quedará invalidada)

Fdo. _____

Negativamente (justificación en TFT05)

Fdo. _____

DIRECTOR DE LA ESCUELA DE INGENIERÍA INFORMÁTICA

Agradecimientos

Tras estos meses de trabajo, quisiera expresar mi más sincero agradecimiento a todas las personas que han hecho posible la realización de este Trabajo Fin de Grado.

En primer lugar, al profesor Francisco Mario Hernández Tejera y tutor de este proyecto, por compartir conmigo su amplia visión fruto de la experiencia, por su constante disposición para orientarme en cada fase del proceso y, sobre todo, por brindarme la oportunidad de llevar a cabo un trabajo con aplicación directa en el ámbito clínico.

A Miguel Ángel Medina Ramírez, tutor colaborador, le agradezco sus valiosas aportaciones técnicas y su paciencia para resolver cada duda presentada.

Finalmente, a mis padres, por su apoyo académico, y personal, incondicional desde el inicio hasta el final de esta carrera; sin su confianza y aliento, este logro no habría sido posible.

A todos, muchas gracias.

Resumen

El proyecto tuvo como objetivo desarrollar y evaluar arquitecturas híbridas para segmentación médica combinando la arquitectura U-Net con bloques xLSTM. Se realizó una revisión del estado del arte, seguida del diseño e implementación de tres variantes del modelo: colocando xLSTM en el cuello de botella, en el codificador y en el codificador y decodificador. El desarrollo siguió un enfoque iterativo: en cada iteración se entrenó y evaluó el modelo, incorporando ajustes basados en los resultados. Los experimentos emplearon conjuntos de datos médicos: Abdomen MRI (2D/3D), Endovis (instrumentos quirúrgicos), NeurIPS (células) y BraTS2023 (tumores cerebrales). Los mejores resultados se obtuvieron en modalidades 3D; el mayor desafío fue diferenciar instrumentos quirúrgicos en imágenes de endoscopia.

Abstract

The project aimed to develop and evaluate hybrid architectures for medical segmentation by combining the U-Net architecture with xLSTM blocks. A state-of-the-art review was performed, followed by the design and implementation of three variants of the model: placing xLSTM at the bottleneck, at the encoder and at the encoder and decoder. The development followed an iterative approach: in each iteration, the model was trained and evaluated, incorporating adjustments based on the results. The experiments used medical datasets: Abdomen MRI (2D/3D), Endovis (surgical instruments), NeurIPS (cells) and BraTS2023 (brain tumours). The best results were obtained in 3D modalities; the biggest challenge was to differentiate surgical instruments in endoscopy images.

Índice general

1. Introducción	1
1.1. Presentación del ámbito de trabajo	1
1.2. Normativa y legislación	3
1.3. Recursos utilizados	5
1.4. Objetivos	7
2. Análisis	9
2.1. Estado del arte	9
2.1.1. Tipos de segmentación	9
2.1.2. Revisión histórica	11
2.1.3. Nuevas Técnicas (2020-Actualidad)	13
2.1.4. Segmentación basada en U-Net	18
2.1.5. Nueva arquitectura: xLSTM(<i>Extended LSTM</i>)	20
2.1.6. Segmentación basada en U-Net y xLSTM	20
3. Competencias y Sostenibilidad	23
3.1. Competencias específicas	23
3.2. Alineamiento con los objetivos de desarrollo sostenible	23
4. Modelos utilizados	25
4.1. Matrix Long Short-Term Memory (mLSTM)	25
4.2. Adaptación de la mLSTM	27
4.3. ViL Block (<i>Vision-xLSTM Block</i>)	29
4.4. Arquitecturas basadas en U-Net y xLSTM	31
4.4.1. xLSTM-UNet_Bot	31
4.4.2. xLSTM-UNet_Enc	33
4.4.3. xLSTM-UNet_EncDec	34
5. Desarrollo	35
5.1. Conjuntos de datos	35
5.1.1. Resonancias magnéticas del abdomen (Abdomen MRI)	35
5.1.2. Imágenes de instrumentos de endoscopia (Endoscopy Images)	36
5.1.3. Imágenes microscópicas (Microscopy Images)	36
5.1.4. Imágenes de tumores cerebrales (BraTS2023 - MEN)	37
5.2. Técnicas de preprocesamiento	39
5.3. Descripción de la metodología	40
5.4. Hiperparámetros del entrenamiento	40
5.4.1. Iteraciones por época y por validación	40
5.4.2. Número de épocas (<i>epochs</i>)	41
5.4.3. Tamaño del lote (<i>batch size</i>)	41

5.4.4.	Tasa de aprendizaje (<i>learning rate</i>)	42
5.4.5.	Decaimiento de pesos y optimizador AdamW	42
5.4.6.	Función de pérdida compuesta: Entropía Cruzada y Dice	43
5.4.7.	Promedio Exponencialmente Móvil (EMA)	43
5.5.	Fases de desarrollo	44
5.5.1.	Primera Iteración: Experimentos preliminares en Google Colaboratory	44
5.5.2.	Segunda Iteración: Desarrollo incremental de los experimentos	45
5.5.3.	Tercera Iteración: Ejecución de experimentos de larga duración en equipo físico	46
6.	Resultados	55
6.1.	Métricas de rendimiento	55
6.1.1.	Coefficiente de Similitud de Dice (DSC)	55
6.1.2.	Índice de Dice de Superficie Normalizado (NSD)	56
6.1.3.	Precision, Sensibilidad y F1 Score	56
6.2.	Resultados de la primera iteración	57
6.2.1.	Resonancias magnéticas del abdomen en 2D	57
6.2.2.	Resonancias magnéticas del abdomen en 3D	58
6.2.3.	Imágenes de instrumentos de endoscopia	58
6.2.4.	Imágenes microscópicas	59
6.3.	Resultados de la segunda iteración	60
6.3.1.	Resonancias magnéticas del abdomen en 2D	60
6.3.2.	Resonancias magnéticas del abdomen en 3D	61
6.3.3.	Imágenes de instrumentos de endoscopia	62
6.4.	Resultados de la tercera iteración	64
6.4.1.	Resonancias magnéticas del abdomen en 2D	64
6.4.2.	Resonancias magnéticas del abdomen en 3D	66
6.4.3.	Imágenes de instrumentos de endoscopia	67
6.4.4.	Imágenes microscópicas	68
6.4.5.	Segmentación de tumores cerebrales	70
7.	Conclusiones	73
7.1.	Conclusiones	73
7.2.	Principales aportaciones	74
7.3.	Trabajo futuro	75

Índice de figuras

2.1.	Imagen original utilizada para la segmentación. Adaptado de Kirillov et al. [25].	10
2.2.	Ejemplo de segmentación semántica en visión por computador. Adaptado de Kirillov et al. [25].	10
2.3.	Ejemplo de segmentación de instancias en visión por computador. Adaptado de Kirillov et al. [25].	10
2.4.	Ejemplo de segmentación panóptica en visión por computador. Adaptado de Kirillov et al. [25].	10
2.5.	Arquitectura de la red U-Net para segmentación de imágenes biomédicas. Ronneberger et al. [36]	19
2.6.	Arquitectura del modelo xLSTM-UNet (<i>xLSTM-UNet_Enc</i>). Imagen extraída de Chen et al. [10].	21
2.7.	Esquema general de la arquitectura propuesta ViLU-Net, presentada por Heidari et al. [16]. (a) Diagrama general del modelo, (b) estructura interna del bloque ViL, (c) módulo de convolución inicial (Conv. Stem), y (d) bloques de muestreo hacia arriba y abajo (Up/Down Sampler). IN representa una operación de normalización por instancia (Instance Normalization).	22
4.1.	Diagrama del funcionamiento interno del bloque mLSTM. Se muestra cómo se combinan la memoria matricial, las puertas y la recuperación mediante consulta.	26
4.2.	Arquitectura del bloque mLSTM con múltiples cabezas de atención y memoria matricial adaptada.	28
4.3.	Esquema del ViLBlock con bifurcación de caminos, subcapa mLSTM, y combinación residual escalada.	30
4.4.	Arquitectura del bloque <i>BasicResBlock</i> .	31
4.5.	Arquitectura del bloque <i>BasicBlockD</i> .	31
4.6.	Arquitectura de <i>xLSTM-UNet_Bot</i> .	32
4.7.	Arquitectura de <i>xLSTM-UNet_Enc</i> .	33
4.8.	Arquitectura de <i>xLSTM-UNet_EncDec</i> .	34
5.1.	Ejemplo de segmentación de órganos abdominales en una imagen de resonancia magnética (MRI) del conjunto de datos AMOS.	35
5.2.	Ejemplo de segmentación de instrumentos quirúrgicos en imágenes endoscópicas del desafío MICCAI 2017 EndoVis Challenge.	36
5.3.	Ejemplo de segmentación celular en imágenes de microscopía del desafío NeurIPS 2022 Cell Segmentation Challenge.	37
5.4.	Ejemplo de segmentación de un tumor cerebral en las distintas modalidades de resonancia magnética del desafío BraTS2023-MEN.	38
5.5.	Curvas de validación (EMA-DSC) por <i>fold</i> en los experimentos con 500 épocas para el conjunto Abdomen MRI 2D.	48
5.6.	Curvas de validación (EMA-DSC) por <i>fold</i> en los experimentos con 700 épocas para el conjunto Abdomen MRI 2D.	49

5.7. Curvas de validación (EMA-DSC) por <i>fold</i> en los experimentos con Abdomen MRI 3D.	50
5.8. Curvas de validación (EMA-DSC) por <i>fold</i> en los experimentos con el conjunto Endovis 2D.	51
5.9. Curvas de validación (EMA-DSC) por <i>fold</i> en los experimentos con el conjunto NeurIPS 2D.	52
5.10. Curvas de validación (EMA-DSC) por <i>fold</i> en los experimentos con 50 épocas para el conjunto BraTS2023 3D.	53
5.11. Curvas de validación (EMA-DSC) por <i>fold</i> en los experimentos con 100 épocas para el conjunto BraTS2023 3D.	54
6.1. Evolución de las métricas DSC y NSD del entrenamiento incremental del modelo en AbdomenMRI (2D), evaluado cada 20 épocas.	60
6.2. Ejemplo cualitativo de segmentación para la modalidad AbdomenMRI (2D) evaluando el progreso cada 20 épocas.	61
6.3. Evolución de las métricas DSC y NSD del entrenamiento incremental del modelo en AbdomenMRI (3D), evaluado cada 20 épocas.	61
6.4. Ejemplo cualitativo de segmentación para la modalidad AbdomenMRI (3D), evaluando el progreso cada 20 épocas.	62
6.5. Evolución de las métricas DSC y NSD del entrenamiento incremental del modelo en Endovis (2D), evaluado cada 20 épocas.	63
6.6. Ejemplo cualitativo del rendimiento del modelo en Endovis (2D), evaluando el progreso cada 20 épocas.	64
6.7. Comparación de las métricas DSC y NSD para los diferentes modelos en el conjunto de datos AbdomenMRI 2D en la tercera iteración.	65
6.8. Ejemplo cualitativo de segmentación en AbdomenMRI 3D, tercera iteración.	66
6.9. Ejemplo cualitativo de segmentación en Endovis 2D, tercera iteración.	67
6.10. DSC por clase en cada <i>fold</i> para ambas arquitecturas en imágenes de microscopía.	69
6.11. Ejemplo cualitativo de segmentación celular en NeurIPS 2D, tercera iteración.	69
6.12. Comparación del DSC por clase (ET, FLAIR, NETC) entre las arquitecturas xLSTMBot y xLSTMEnc en el conjunto de prueba de BraTS2023-MEN.	70
6.13. Ejemplo cualitativo de segmentación en un corte axial con las cuatro modalidades de entrada (T1, T1Gd, T2, FLAIR).	71
6.14. Ejemplo tridimensional de segmentación tumoral en modalidad T1 mediante nueve cortes seleccionados.	72

Índice de cuadros

3.1. Grado de relación del TFG con los objetivos de desarrollo sostenible [34]. . .	24
5.1. Configuraciones de entrenamiento durante la primera iteración experimental.	45
5.2. Planificación del experimento para el dataset de segmentación de tumores cerebrales.	46
5.3. Experimentos evaluados de forma incremental.	46
5.4. Configuración y tiempos de los experimentos con AbdomenMRI 2D en la tercera iteración.	47
5.5. Configuración y tiempos de los experimentos con AbdomenMRI 3D en la tercera iteración.	49
5.6. Configuración y tiempos de los experimentos con Endovis 2D en la tercera iteración.	51
5.7. Configuración y tiempos de los experimentos con NeurIPS 2D en la tercera iteración.	52
5.8. Configuración y tiempos de los experimentos con BraTS2023 3D en la tercera iteración.	53
6.1. Resultados de DSC y NSD para experimentos con AbdomenMRI (2D) en la iteración 1.	58
6.2. Resultados de DSC y NSD para experimentos con AbdomenMRI (3D) en la iteración 1.	58
6.3. Resultados de DSC y NSD para experimentos con Endovis (2D) en la iteración 1.	59
6.4. Resultados de DSC y F1 Score para experimentos con NeurIPS (2D) en la iteración 1.	59
6.5. Resultados comparativos de DSC y NSD para AbdomenMRI 2D.	65
6.6. Resultados comparativos de DSC y NSD para AbdomenMRI 3D.	66
6.7. Resultados comparativos de DSC y NSD para Endovis 2D.	67
6.8. Resultados comparativos de F1-Score para NeurIPS 2D.	68

Capítulo 1

Introducción

No se trata de reemplazar médicos,
sino de aumentar sus capacidades.

Eric Topol

1.1. Presentación del ámbito de trabajo

En el marco de la visión por computador, la segmentación de imágenes se ha consolidado como una de las tareas más relevantes, impulsada en gran medida por los avances del aprendizaje profundo. Esta técnica ha adquirido un papel fundamental en múltiples campos, desde la gestión del tráfico en ciudades inteligentes y la conducción autónoma, hasta el diagnóstico médico. En particular, en el ámbito de la medicina, la segmentación de imágenes permite dividir una imagen en regiones con significado clínico, como órganos, tejidos o lesiones, lo que facilita la identificación, localización y medición de posibles enfermedades. Esto no solo permite un análisis automatizado más preciso, sino que también apoya un diagnóstico más eficiente y un seguimiento más efectivo de las condiciones médicas.

La segmentación de imágenes consiste en dividir una imagen en regiones o segmentos con significado, permitiendo extraer información estructurada y relevante a partir de datos visuales. En el ámbito médico, esta técnica resulta esencial para tareas como la planificación quirúrgica, la navegación intraoperatoria, el estudio a largo plazo de pacientes o el entrenamiento de sistemas asistidos por inteligencia artificial. En los últimos años, el progreso del aprendizaje profundo ha hecho que los algoritmos de segmentación sean mucho más precisos y eficientes, lo que ha ayudado a resolver problemas cada vez más complejos con gran nivel de detalle. No obstante, este proceso aún enfrenta importantes desafíos, especialmente al tratar con imágenes que presentan alta variabilidad estructural, ruido, artefactos o baja resolución.

En sus primeras etapas de desarrollo, los algoritmos de segmentación se basaban en métodos clásicos como la detección de bordes, la umbralización o el crecimiento de regiones. Aunque estos métodos resultaron eficaces en condiciones controladas, presentaban limitaciones notables ante la diversidad e irregularidades propias de escenarios reales, especialmente en el ámbito clínico, donde pueden observarse variaciones anatómicas significativas entre pacientes. A medida que los conjuntos de datos crecieron en tamaño y complejidad, estos métodos tradicionales se volvieron insuficientes, lo que evidenció la necesidad de modelos más robustos y adaptativos, abriendo así paso a técnicas basadas en aprendizaje automático.

Con la llegada del aprendizaje profundo, la segmentación de imágenes experimentó un avance sin precedentes. En concreto, las redes neuronales convolucionales (CNN, *Convolutional Neuronal Network*) permitieron mejorar significativamente la precisión y la capacidad de generalización de los modelos, ya que pueden aprender patrones y características de forma jerárquica directamente a partir de los datos de entrada. Uno de los hitos más relevantes en este contexto fue la introducción de la arquitectura U-Net[36], diseñada originalmente para la segmentación de imágenes biomédicas, pero ampliamente adoptada en otros campos. Su estructura simétrica en forma de “U”, que combina codificadores y decodificadores mediante conexiones de salto (*skip connections*), permite capturar tanto el contexto global como los detalles espaciales finos, lo que mejora notablemente el rendimiento incluso en contextos complejos.

Paralelamente, el tratamiento de secuencias espaciales y temporales ha impulsado el desarrollo de arquitecturas capaces de modelar relaciones de largo alcance, principalmente en entornos donde el significado depende en gran medida del contexto. Un ejemplo destacado son las redes neuronales recurrentes, y en particular las LSTM (Long Short-Term Memory)[17], diseñadas para retener información a largo plazo y originalmente aplicadas en tareas de procesamiento secuencial como el lenguaje natural.

Recientemente, se han propuesto enfoques híbridos que combinan las fortalezas de las redes convolucionales y recurrentes para abordar tareas de segmentación en contextos complejos o dinámicos. Un ejemplo destacado es la arquitectura xLSTM (*Extended LSTM*), una variante mejorada de las LSTM que optimiza el manejo de la memoria y permite un procesamiento paralelo más eficiente, sin perder la capacidad de capturar dependencias a largo plazo. Estas mejoras han facilitado su integración en arquitecturas de segmentación como la U-Net, dando lugar a modelos híbridos que aprovechan tanto la extracción jerárquica de características de las CNN como la memoria extendida y la flexibilidad secuencial de las xLSTM, potenciando así el rendimiento en tareas con alta variabilidad espacial y temporal.

En este contexto, y en medio del auge de los modelos híbridos en inteligencia artificial, se enmarca el presente Trabajo de Fin de Grado. El objetivo principal es diseñar y evaluar un sistema de segmentación de imágenes médicas que combine la arquitectura U-Net con componentes xLSTM, aprovechando las fortalezas de ambas tecnologías para mejorar la precisión y robustez frente a la alta variabilidad estructural de los datos. En particular, se analizan tres variantes arquitectónicas: una que incorpora xLSTM en el cuello de botella de la U-Net (xLSTM-UNet_Bot), otra que lo integra a lo largo del codificador (xLSTM-UNet_Enc) y una tercera que lo implementa tanto en el codificador como en el decodificador (xLSTM-UNet_EncDec). El trabajo busca determinar cuál de estas configuraciones híbridas ofrece un mejor rendimiento en términos de precisión, generalización y capacidad para modelar dependencias espaciales complejas en tareas de segmentación médica.

En paralelo a los desafíos técnicos, la aplicación de modelos de segmentación en medicina también plantea importantes cuestiones legales y éticas. El uso de datos médicos está sujeto a un estricto marco normativo que incluye tanto el Reglamento General de Protección de Datos (RGPD) como la Ley Orgánica de Protección de Datos y Garantía de Derechos Digitales (LOPDGDD). A esto se suma la Ley de Autonomía del Paciente, que impone condiciones específicas sobre la confidencialidad y el uso de la información sanitaria. Estas normativas

obligan a que los datos sean anonimizados antes de su uso, lo que introduce limitaciones adicionales en el diseño de los modelos y en la gestión de los conjuntos de datos. Asimismo, el uso de servicios en la nube o APIs externas puede entrar en conflicto con las políticas de seguridad de muchas instituciones, especialmente en entornos hospitalarios o de investigación clínica. Por ello, las soluciones locales, capaces de operar sin conexión a servicios externos, resultan particularmente atractivas y necesarias en este ámbito.

Este proyecto se sitúa en la intersección entre la investigación en inteligencia artificial, las necesidades clínicas reales y las limitaciones legales y de infraestructura. Desarrollado desde una perspectiva aplicada, este trabajo aborda escenarios en los que los métodos tradicionales de segmentación resultan poco eficaces, y donde las arquitecturas híbridas, como las que se proponen aquí, pueden ofrecer mejoras sustanciales. A través del diseño y análisis comparativo de tres variantes de U-Net con integración de xLSTM, se busca no solo explorar nuevas soluciones técnicas, sino también ofrecer una propuesta realista, escalable y con capacidad de generalización, apta para su implementación en contextos clínicos o de investigación con recursos limitados.

En definitiva, este trabajo tiene como propósito ofrecer una solución innovadora y práctica al desafío de la segmentación de imágenes médicas, mediante la evaluación del rendimiento de arquitecturas híbridas basadas en U-Net y xLSTM. A través de esta investigación, se busca no solo aportar valor técnico y científico al ámbito académico, sino también desarrollar una propuesta aplicable en entornos reales donde la segmentación precisa, la eficiencia operativa y el respeto por la privacidad son factores críticos. Se espera que los resultados obtenidos puedan contribuir al avance de sistemas automatizados en contextos clínicos y de investigación, especialmente en escenarios con recursos limitados.

1.2. Normativa y legislación

En este estudio se desarrolló un modelo de segmentación de imágenes médicas utilizando diversos datasets públicos, como Abdomen MRI, Endoscopy images, Microscopy images y BraTS2023 MEN. El uso de estos datos está sujeto a la legislación vigente en materia de protección de datos, dado que se trata de información biomédica potencialmente sensible, aunque en este caso esté debidamente anonimizada.

La normativa principal en protección de datos personales en investigación médica aplicable incluye:

- El **Reglamento (UE) 2016/679, General de Protección de Datos (RGPD)**, que establece los principios y garantías para el tratamiento de datos personales en la Unión Europea[41].
- La **Ley Orgánica 3/2018, de Protección de Datos Personales y garantía de los derechos digitales (LOPDGDD)**, que adapta el RGPD al marco legal español[12].
- La **Ley 41/2002, básica reguladora de la autonomía del paciente y de derechos y obligaciones en materia de información y documentación clínica**[11].

Estas normas contemplan como categoría especial los datos relativos a la salud, esta-

bleciendo que su tratamiento está prohibido, salvo que se cumpla alguna de las condiciones recogidas en el artículo 9 del RGPD, como el consentimiento explícito del interesado o su uso con fines de investigación científica en el ámbito sanitario, siempre aplicando garantías adecuadas como la anonimización.

Anonimización y uso legítimo de los datos

Para el desarrollo y entrenamiento de los modelos se utilizaron varios conjuntos de datos de imágenes médicas, previamente procesados y liberados por instituciones científicas y por organizadores de desafíos internacionales de segmentación. Todos ellos están destinados al uso en contextos de investigación académica y se han puesto a disposición de la comunidad en repositorios públicos, con fines de evaluación y desarrollo de algoritmos de aprendizaje automático.

En particular:

- Los datasets utilizados (incluidos los de órganos abdominales, instrumentos quirúrgicos endoscópicos y células microscópicas) se encuentran disponibles en el repositorio [xLSTM-UNet\[9\]](#), el cual enlaza a su vez con conjuntos de datos empleados en los desafíos públicos AMOS (MICCAI 2022)[28], EndoVis (MICCAI 2017)[1] y NeurIPS Cell Segmentation Challenge (2022)[27].
- Estos conjuntos se encontraban previamente anonimizados, no contienen información personal identificable y utilizan identificadores abstractos (como `amos_id` o `cell_id`) no vinculados a ninguna identidad real. En consecuencia, no están sujetos a la aplicación directa de la LOPDGDD ni del RGPD, al no tratarse de datos personales según los términos legales definidos por dichas normativas.
- En el caso del dataset BraTS2023[30][23], su uso se encuentra regulado por las condiciones del propio reto, que incluyen un acuerdo explícito de uso para investigación. Estas condiciones garantizan:
 - La confidencialidad y anonimización de los datos.
 - El uso restringido al ámbito académico o del desafío.
 - La obligación de citar las publicaciones asociadas.
 - El cumplimiento de principios éticos y legales acordes al RGPD y la LOPDGDD.

Por tanto, el uso de todos los datos empleados en este proyecto se considera legítimo dentro del marco académico, respetando los principios de confidencialidad, anonimización y limitación del tratamiento.

Consideraciones éticas

De acuerdo con la *Ley de Autonomía del Paciente* (Ley 41/2002), los pacientes tienen derecho a la confidencialidad de su información clínica y a la protección de sus datos personales. En este contexto, el uso de imágenes médicas anonimizadas en investigaciones como esta respeta plenamente la normativa, dado que no es posible asociar las imágenes con individuos concretos, y su utilización tiene como finalidad el avance científico y tecnológico en el

campo del análisis de imágenes médicas mediante inteligencia artificial.

1.3. Recursos utilizados

Durante el desarrollo experimental del proyecto, se recurrió a diferentes entornos en función de las necesidades de procesamiento. En las etapas iniciales y para pruebas de menor escala, se utilizó Google Colaboratory, aprovechando una GPU NVIDIA Tesla T4. En este entorno, los recursos disponibles incluyen 12.7 GB de RAM del sistema (con aproximadamente 1.1 GB en uso), 15 GB de RAM de GPU y 112.6 GB de almacenamiento en disco. Este entorno se consideró adecuado para el desarrollo inicial y la ejecución de tareas menos exigentes en cuanto a cómputo y memoria.

Sin embargo, conforme aumentó la complejidad de los experimentos y la demanda computacional, se decidió trasladar su ejecución a un entorno físico de cómputo avanzado. Dicho entorno está conformado por un ordenador de escritorio equipado con un procesador Intel Core i7-13700K de 13^a generación, operando a 3.40 GHz. El sistema dispone de 64 GB de memoria RAM (63.7 GB utilizables) y funciona bajo el sistema operativo Windows 11 Pro, versión 24H2, con arquitectura de 64 bits. Además, cuenta con una tarjeta gráfica NVIDIA GeForce RTX 4090 con 24 GB de VRAM, lo que permite acelerar eficientemente las operaciones computacionales intensivas, como el entrenamiento de modelos de aprendizaje automático. El almacenamiento principal consiste en una unidad de 1.81 TB que ofrece una alta velocidad de acceso a los datos.

El desarrollo del proyecto, enfocado a la segmentación de imágenes médicas mediante las arquitecturas híbridas de U-Net y xLSTM, ha implicado la utilización de diversas herramientas de software, librerías especializadas y entornos de desarrollo para las tareas de preprocesamiento de imágenes, entrenamiento de modelos y visualización de resultados.

Python ha sido el lenguaje de programación principal empleado durante todo el estudio, por su uso generalizado en el ámbito de la inteligencia artificial y la visión por computador, así como la disponibilidad de librerías específicas para el procesamiento de imágenes médicas y el modelado. El entorno de desarrollo integrado (IDE) utilizado ha sido Visual Studio Code (VSCode) [31], facilitando la edición y organización del código fuente.

Para el entrenamiento de modelos y preprocesamiento de datos, se utilizaron Jupyter Notebooks [22] permitiendo así un flujo de trabajo interactivo y flexible. Su compatibilidad con herramientas de visualización, así como su integración con sistemas externos de control de versiones como Git, facilita tanto la experimentación como la documentación del proceso de desarrollo.

Se adaptó el modelo base a partir de la implementación oficial de xLSTM-UNet-PyTorch, disponible en [GitHub](#)[9], el cual implementa dos variantes de redes neuronales convolucionales orientadas a la segmentación de imágenes biomédicas. En particular, se utilizaron ambas arquitecturas descritas en el artículo original: una que integra un bloque xLSTM en el cuello de botella de la U-Net, y otra que incorpora los bloques de la xLSTM en el codificador. Además, se empleó la librería nnU-Net v2 [9], desarrollada por los propios autores del artículo original para gestionar el pipeline completo: preprocesamiento, planificación de experimentos,

entrenamiento, inferencia y evaluación. Esta librería se modificó ligeramente con el fin de permitir la inclusión, como parámetros en sus funciones, de ciertos hiperparámetros por defecto necesarios para la experimentación, los cuales no se encontraban especificados en el artículo original. Asimismo, se implementó una versión adicional adaptada a las capacidades de cómputo disponibles, basada en el enfoque propuesto por Heidari et al. [16], que extiende la integración de bloques xLSTM tanto al codificador como al decodificador.

Entre las librerías más relevantes empleadas en este proyecto se encuentran:

- **torch==2.0.0** [39]: Framework principal para la construcción, entrenamiento y evaluación de redes neuronales profundas.
- **monai==1.3.0** [33]: Librería especializada en imágenes médicas basada en PyTorch, utilizada para facilitar la gestión de volúmenes 3D y transformaciones.
- **nibabel** [7], **SimpleITK** [40], **dicom2nifti** [43]: Utilizadas para la lectura, manipulación y conversión de formatos estándar en imágenes médicas como NIfTI y DICOM.
- **matplotlib** [19], **seaborn** [47], **ipywidgets** [38]: Herramientas clave para la visualización de datos, incluyendo vistas en 2D y 3D, representación de máscaras segmentadas y creación de interfaces interactivas para la exploración de los resultados.
- **scikit-image** [42], **opencv-python** [6]: Para operaciones avanzadas de procesamiento de imágenes, como filtros, binarización y extracción de contornos.
- **numpy** [15], **pandas** [29], **scipy** [44]: Para el manejo y procesamiento eficiente de datos estructurados y matrices multidimensionales.
- **graphviz** [45], **imagecodecs** [14], **tiffle** [13]: Complementos para la visualización de arquitecturas, compresión y manipulación de formatos de imagen específicos utilizados en *datasets* biomédicos.
- **batchgenerators** [20], **dynamic-network-architectures** [21], **acvl-utils** [2]: Conjunto de utilidades desarrolladas por la comunidad de *nnU-Net* para el preprocesamiento automatizado de datos y la construcción de arquitecturas de red.
- **scikit-learn** [35]: Para la evaluación de modelos mediante métricas tradicionales como precisión, sensibilidad y puntuación F1.

Además, se utilizaron scripts en Bash para la ejecución automatizada de tareas en el entorno físico con mayor capacidad de cómputo, equipado con una GPU NVIDIA RTX 4090, destinada al entrenamiento de modelos con datasets de gran tamaño y resolución.

Finalmente, para el seguimiento y análisis de los entrenamientos, se utilizó la plataforma Weights & Biases (wandb [48]). Esta herramienta permite registrar métricas en tiempo real, visualizar curvas de aprendizaje, comparar diferentes ejecuciones y almacenar de forma centralizada los experimentos realizados. Su integración con frameworks como PyTorch facilita la monitorización del rendimiento de los modelos durante todas las fases del entrenamiento, incluyendo validación cruzada, pruebas con distintos hiperparámetros y selección de la mejor arquitectura.

1.4. Objetivos

El propósito principal fue desarrollar e implementar un procedimiento de segmentación de imágenes médicas basado en la arquitectura U-Net, integrando la variante xLSTM (*Extended LSTM*) con el fin de mejorar la captura de relaciones espaciales complejas de largo alcance. Para alcanzar esta meta, se establecieron diversos objetivos específicos, que abarcan desde la revisión del estado del arte hasta la evaluación final del modelo.

En primer lugar, se llevó a cabo una revisión bibliográfica, centrada en las técnicas de segmentación de imágenes, la arquitectura U-Net y los modelos LSTM, haciendo especial énfasis en la variante xLSTM. Durante este estudio se recopilieron trabajos relevantes del estado del arte que abordan tanto los enfoques clásicos de segmentación como los avances recientes en la incorporación de mecanismos de memoria en redes neuronales convolucionales aplicadas a imágenes médicas. Esta etapa de revisión permitió comprender las fortalezas y limitaciones de la U-Net en la segmentación biomédica, así como identificar los beneficios potenciales que ofrecen las variantes xLSTM para capturar dependencias contextuales de largo alcance. De este modo, se consolidó el marco conceptual necesario para el desarrollo posterior del modelo.

Con base en lo aprendido, la segunda fase consistió en la implementación práctica de los modelos propuestos. Se diseñó una red neuronal de tipo U-Net en un entorno de desarrollo de aprendizaje profundo a la cual se integraron bloques de xLSTM en posiciones estratégicas de la arquitectura. En esta etapa se definieron decisiones técnicas relevantes, tales como la estructura de las capas y el punto de inserción de los módulos xLSTM. A partir del artículo original, se consideró la colocación de los bloques xLSTM en el cuello de botella de la U-Net (xLSTM-UNet_Bot), en las capas del codificador (xLSTM-UNet_Enc) y/o en las capas del codificador y decodificador (xLSTM-UNet_EncDec), con el objetivo de evaluar su efecto sobre la captación de contexto a diferentes niveles de abstracción.

Luego, comenzaron los entrenamientos de los modelos implementados. Para ello se seleccionaron conjuntos de datos apropiados de imágenes médicas (por ejemplo, volúmenes de resonancia magnética cerebral con sus máscaras segmentadas). Se estableció un protocolo de entrenamiento cuidadoso, definiendo la función de pérdida apropiada (combinaciones de entropía cruzada y coeficiente de Dice), así como los hiperparámetros clave (tasa de aprendizaje, tamaño de lote, número de épocas, optimizador, etc.). Además, se empleó validación cruzada o un conjunto de validación para monitorizar el progreso del modelo durante el entrenamiento, ajustando los parámetros según el desempeño observado y evitando así el sobreajuste. Cabe destacar que no se abordó como un proceso estático, sino que adoptó un enfoque iterativo: a partir de los resultados obtenidos en entrenamientos preliminares, se introdujeron mejoras graduales en la arquitectura, los hiperparámetros y las técnicas de regularización, lo cual permitió perfeccionar progresivamente el comportamiento del modelo. El despliegue se llevó a cabo en un entorno físico con capacidad de cómputo adecuada para las fases de entrenamiento, inferencia y evaluación, lo que permitió validar el rendimiento del sistema en condiciones reales de uso. Dado que no se disponía del acceso a un entorno hospitalario para realizar estas pruebas, el desarrollo se llevó a cabo en un entorno controlado de investigación.

Al final de cada entrenamiento, se procedía a la evaluación de resultados. En esta etapa

se analiza la capacidad de la arquitectura. Se definieron métricas cuantitativas de evaluación (como el coeficiente de Dice, la precisión, la sensibilidad y el F1 Score) para medir el desempeño de la segmentación en los conjuntos de prueba. Además, los resultados de cada ronda sirvieron para identificar puntos críticos de mejora y guiar ajustes posteriores, tanto en la configuración del modelo como en el conjunto de datos o los procedimientos de pre-procesamiento. Gracias a este ciclo continuo de entrenamiento-evaluación-ajuste, fue posible alcanzar una configuración robusta y bien adaptada al dominio biomédico.

Finalmente, se llevó a cabo la documentación detallada de todo el proceso seguido y un análisis crítico de los resultados obtenidos. La memoria del proyecto recoge la metodología seguida en cada fase, las decisiones de diseño justificadas (arquitectura de red, parámetros de entrenamiento, criterios de evaluación, etc.) y los hallazgos más relevantes. A partir de este análisis se extrajeron conclusiones sobre el impacto del uso de xLSTM en la segmentación de imágenes médicas, discutiendo sus fortalezas, sus limitaciones y posibles líneas de trabajo futuro.

Capítulo 2

Análisis

2.1. Estado del arte

En la última década, la segmentación de imágenes ha experimentado avances significativos en el ámbito de la visión por computador, impulsados por la integración de modelos de aprendizaje profundo. Su aplicación se extendió ampliamente en diversas áreas, incluyendo el diagnóstico médico, la conducción autónoma, el análisis de imágenes satelitales, la gestión del tráfico en ciudades inteligentes, la detección de defectos en procesos de fabricación e incluso en la agricultura, donde se facilita la identificación de especies vegetales no deseadas para los agricultores.

La segmentación de imágenes es una técnica que consiste en dividir una imagen digital en múltiples regiones o segmentos con características similares, con el fin de facilitar la identificación, clasificación y análisis de los objetos presentes en la imagen. Su objetivo principal es agrupar píxeles en regiones significativas y no superpuestas, permitiendo una mejor interpretación de la escena y el razonamiento visual.

2.1.1. Tipos de segmentación

Con la llegada de los Modelos Fundacionales (*Foundation Models*, FMs), la segmentación ha evolucionado hacia métodos más flexibles y generalizables. En este contexto, se identificaron dos grandes categorías de segmentación: la segmentación genérica (*Generic Image Segmentation*, GIS) y la segmentación basada en prompts (*Promptable Image Segmentation*, PIS) [49].

Segmentación Genérica (GIS)

La segmentación genérica tiene como objetivo dividir una imagen sin la necesidad de información previa o intervención del usuario, basándose únicamente en características inherentes de la imagen. Esta se clasifica en tres categorías principales. Se presentan a continuación ejemplos que ilustran los distintos tipos de segmentaciones aplicadas a la imagen original, mostrada en la Figura 2.1.

- **Segmentación semántica.** La segmentación semántica consiste en clasificar cada píxel de una imagen según su significado semántico tratando todos los objetos de la misma categoría como una sola entidad. Específicamente, en la Figura 2.2, se observa cómo una imagen urbana ha sido segmentada en diferentes clases como vehículos, personas, edificios y señales de tráfico, destacando los distintos elementos con colores distintivos.



Figura 2.1: Imagen original utilizada para la segmentación. Adaptado de Kirillov et al. [25].

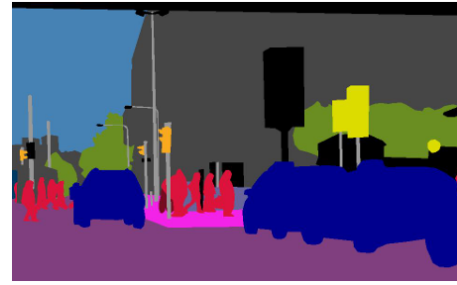


Figura 2.2: Ejemplo de segmentación semántica en visión por computador. Adaptado de Kirillov et al. [25].

- **Segmentación de instancias.** La segmentación de instancias, al contrario que la segmentación semántica, distingue entre distintos objetos de la misma clase, es decir, se centra en identificar los límites exactos de cada objeto individual.

Se observa claramente en la Figura 2.3, donde, a diferencia de la Figura 2.2, los objetos se identificaron de manera individual mediante máscaras de colores únicos y cuadros delimitadores. En este caso, se segmentaron de forma independiente los vehículos, peatones y otros elementos presentes en la escena, permitiendo no solo su clasificación, sino también su conteo y ubicación exacta en la imagen.



Figura 2.3: Ejemplo de segmentación de instancias en visión por computador. Adaptado de Kirillov et al. [25].

- **Segmentación panóptica.** Combina la segmentación semántica y de instancias, generando una predicción completa que distingue tanto los objetos individuales como las regiones de fondo de la imagen. La Figura 2.4 ilustra este concepto, donde se observa que cada objeto ha sido segmentado tanto a nivel de categoría como de instancia.



Figura 2.4: Ejemplo de segmentación panóptica en visión por computador. Adaptado de Kirillov et al. [25].

A diferencia de la segmentación semántica, que agrupa todos los objetos de una misma clase en una única máscara, y de la segmentación de instancias, que identifica cada objeto individualmente, la segmentación panóptica permite una representación más rica del entorno. Por ejemplo, se diferenciaron los peatones y automóviles tanto por su categoría como a nivel individual, mientras que elementos como edificios y carreteras se representaron como clases generales, sin instancias específicas.

Segmentación Basada en Prompts (PIS)

A diferencia de la segmentación genérica (GIS), la segmentación basada en prompts incorpora entradas adicionales, como texto o interacción del usuario, para definir la región de interés dentro de la imagen. Existen tres enfoques principales:

- **Segmentación interactiva:** Permite a los usuarios refinar la segmentación mediante clics, cajas delimitadoras o trazos. Modelos como SAM han revolucionado este tipo de segmentación, permitiendo ajustes en tiempo real.
- **Segmentación referencial:** Se basa en descripciones en lenguaje natural para localizar y segmentar objetos dentro de la imagen. Modelos como CLIP y LISA han demostrado un rendimiento sobresaliente en este ámbito.
- **Segmentación *few-shot*:** Aplica técnicas de aprendizaje con pocos ejemplos, permitiendo segmentar objetos en imágenes nuevas con solo unas pocas muestras de referencia. Modelos como SegGPT han explorado este paradigma con éxito.

2.1.2. Revisión histórica

Atendiendo a diversos estudios sobre las técnicas de segmentación de imágenes realizados entre 2014 y 2024, como los de Wasim [46] y Minaee et al. [32], se llevó a cabo una revisión de la evolución de estas técnicas, desde los enfoques clásicos hasta la irrupción del aprendizaje profundo y los modelos basados en *transformers*.

1. Métodos Clásicos de Segmentación (1970-2000)

Los primeros enfoques de segmentación de imágenes se fundamentaban en métodos matemáticos y técnicas de procesamiento de imágenes convencionales. Estos métodos, aunque eficaces en ciertas aplicaciones, presentan limitaciones en entornos complejos con variaciones significativas de iluminación y textura. Entre los enfoques más representativos se encuentran:

- **Umbralización** (Otsu, 1979): Técnica basada en la distribución de intensidad de los píxeles para determinar umbrales óptimos que permitan separar los objetos del fondo.
- **Detección de bordes** (Canny, 1986): Métodos basados en operadores como Sobel para identificar discontinuidades en la intensidad de la imagen, facilitando la detección de contornos.
- **Segmentación por crecimiento de regiones** (1990s): Algoritmos que agrupan píxeles contiguos con características similares, definiendo regiones homogéneas dentro de la imagen.
- **Métodos basados en clustering** (K-means, Expectación-Maximización, 1990s): Estrategias que segmentan la imagen en grupos según la distribución estadística de los colores o texturas.
- **Transformada de Watershed** (1990s): Método basado en la morfología matemática que modela la imagen como una topografía para segmentarla en regiones bien definidas.

2. Modelos Basados en Aprendizaje Automático (2000-2015)

Con el desarrollo del aprendizaje automático, la segmentación de imágenes experimentó una mejora significativa mediante la incorporación de técnicas estadísticas y basadas en teoría de grafos. Durante este período, se introdujeron los siguientes enfoques:

- **Graph Cut y Mean Shift** (2001-2010): Métodos de optimización basados en teoría de grafos para segmentar imágenes de manera eficiente.
- **Modelos de Contornos Activos** (Snakes, 1988-2010): Técnicas que emplean minimización de energía para ajustar los contornos de los objetos detectados, mejorando la segmentación en entornos ruidosos.
- **SVM y Random Forests** (2005-2015): Primeros modelos de aprendizaje supervisado aplicados a la segmentación de imágenes, utilizando características de textura, color y forma para mejorar la clasificación de regiones.

3. Aprendizaje Profundo y Redes Convolucionales (2015-2020)

La introducción de redes neuronales convolucionales (CNNs) marcó un cambio de paradigma en la segmentación de imágenes, permitiendo avances considerables en precisión y eficiencia. Entre las arquitecturas más destacadas de esta era se encuentran:

- **R-CNN (2014)**: Primer modelo en combinar regiones propuestas con CNNs, utilizando *Selective Search*, pero con alta demanda computacional.
- **Fast R-CNN (2015)**: Optimiza R-CNN con *RoI Pooling*, reduciendo significativamente el tiempo de inferencia y mejorando la eficiencia del entrenamiento.
- **Fully Convolutional Networks (FCN, 2015)**: Sustituye las capas densas por convoluciones, permitiendo segmentaciones de tamaño variable sin restricciones de entrada fija.
- **U-Net (2015)**: Arquitectura codificador-decodificador con conexiones de salto, diseñada para segmentación médica y extendida a otros dominios.
- **Faster R-CNN (2015)**: Introduce la *Red de Propuesta de Regiones (RPN)*, eliminando *Selective Search* y mejorando la eficiencia en detección de objetos.
- **DeepLab (2017-2018)**: Introduce convoluciones dilatadas y *Conditional Random Fields (CRF)* para mejorar la segmentación en términos de precisión de bordes y contexto.
- **Mask R-CNN (2017)**: Extiende Faster R-CNN con una rama adicional para la predicción de máscaras, permitiendo la segmentación de instancias.
- **G-RCNN (2018)**: Variante de Faster R-CNN con refinamiento geométrico, mejorando la precisión en la detección de objetos y la alineación de cuadros delimitadores.

2.1.3. Nuevas Técnicas (2020-Actualidad)

Mientras que hasta 2019 predominaban los métodos basados en CNN (como U-Net), desde 2020 el campo ha vivido una transformación radical gracias a arquitecturas como *transformers*, modelos fundacionales, métodos multimodales, técnicas de aprendizaje auto-supervisado y de pocos o ningún dato anotado. Este documento sintetiza estos avances, destacando su impacto, adaptaciones al ámbito clínico y herramientas abiertas para su implementación.

La información presentada en esta sección se recopiló y sintetizó a partir de diversos trabajos recientes que abordan los avances en segmentación médica mediante nuevas arquitecturas y enfoques. En particular, se consultaron los estudios de Azad et al. (2023), Awais et al. (2023), Zhou et al. (2024), Chen et al. (2023), Huang et al. (2023), Kazerouni et al. (2023) y otros trabajos relevantes del período 2020 en adelante [4, 3, 49, 8, 18, 24, 37].

1. Revolución con Transformers (2020-2023)

La introducción de *transformers*, originalmente diseñados para tareas de procesamiento del lenguaje natural (NLP), marcó un antes y un después en el campo de la segmentación de imágenes, al permitir modelar relaciones de largo alcance entre regiones de la imagen. A diferencia de las CNN tradicionales, que se limitan a contextos locales debido a su arquitectura basada en ventanas de convolución, los transformers permiten capturar dependencias globales, lo cual resulta fundamental en tareas donde el contexto general influye en la clasificación de píxeles o regiones.

- **Vision Transformer (ViT, 2020)**: fue el primer modelo en aplicar directamente la arquitectura de transformers al dominio visual. ViT dividía las imágenes en parches no superpuestos, tratándolos como tokens de una secuencia, al igual que las palabras en una oración. Aunque ViT demostró un gran potencial para clasificación, al no capturar con suficiente detalle las relaciones espaciales locales, como bordes, formas pequeñas o la continuidad entre regiones adyacentes, el modelo pierde información importante para segmentar con precisión los objetos dentro de una imagen.
- **SETR (Segmenter Transformer, 2021)**: propuso una solución específica para segmentación semántica, usando como base un ViT para extraer características globales, seguido de módulos de reconstrucción para recuperar la resolución espacial. Esta arquitectura demostró que los transformers podían superar a las CNN en tareas densas cuando se adaptaban correctamente.
- **TransUNet (2021)**: integró lo mejor de ambas aproximaciones al emplear una arquitectura tipo U-Net como base, reemplazando el codificador basado en CNN por un ViT. De esta manera, se combina la capacidad de capturar información contextual a gran escala con la eficiencia espacial característica del diseño U-Net.
- **Swin-UNet / Swin UNETR (2021–2022)**: incorporaron el Swin Transformer, que introduce un mecanismo de atención jerárquica mediante ventanas deslizantes. Esta estrategia consiste en dividir la imagen en pequeñas regiones locales (ventanas) sobre las cuales se aplica la autoatención, y luego desplazar estas ventanas para permitir la in-

teracción entre regiones adyacentes. Al organizar el procesamiento de forma jerárquica, el modelo puede capturar información contextual en distintos niveles de resolución, reduciendo significativamente el coste computacional sin perder la capacidad de modelar relaciones de largo alcance.

- **nnFormer (2022)**: avanzó un paso más al combinar convoluciones 3D con bloques de *transformers*, logrando una mejor extracción de características locales y globales en tareas de segmentación volumétrica, especialmente útil en contextos clínicos como resonancias magnéticas o tomografías computarizadas 3D.
- **DA-TransUNet y UNesT (2023)**: introdujeron mejoras estructurales relevantes. DA-TransUNet incorporó un mecanismo de atención dual, que combina atención espacial (centrada en dónde se encuentra la información relevante) y atención de canal (enfocada en qué tipo de información es importante), permitiendo capturar interacciones más complejas y representaciones más detalladas entre las características extraídas. Por su parte, UNesT utilizó una arquitectura anidada, donde varios bloques de *transformers* jerárquicos se organizan en diferentes niveles. Esto permitió combinar de forma progresiva la información contextual a distintas escalas.

2. Modelos Fundacionales y la Segmentación Universal (2023-2025)

La aparición de modelos cada vez más generales ha dado lugar al desarrollo de los llamados modelos fundacionales. Estos modelos se entrenan a gran escala en conjuntos de datos masivos y diversos, y están diseñados para ser muy flexibles, capaces de adaptarse a una amplia gama de tareas mediante ajustes mínimos (*fine-tuning*, adaptadores, etc.). En el contexto de la visión por computador, esta evolución ha impulsado el concepto de segmentación universal, refiriéndose a la capacidad de segmentar cualquier objeto o estructura en cualquier dominio, sin necesidad de entrenar modelos específicos para cada tarea o conjunto de datos.

- **Segment Anything Model (SAM, Meta AI, 2023)**: marca un punto de inflexión en esta tendencia. Entrenado con más de mil millones de máscaras sobre una gran variedad de imágenes, SAM es capaz de segmentar prácticamente cualquier objeto con la ayuda de prompts (señales de entrada como puntos, cajas o texto). Su arquitectura está basada en un decodificador ligero que actúa sobre un potente extractor de características (Image Encoder), permitiendo segmentación interactiva, en tiempo real y sin necesidad de entrenamiento adicional. Esto sienta las bases para una segmentación verdaderamente universal.
- **Adaptaciones al dominio médico** comenzaron a surgir poco después, aprovechando la generalidad de SAM pero ajustándolo a los desafíos propios de las imágenes médicas:
 - **MedSAM (2024)**: adaptado a más de 30 tipos de cáncer, MedSAM fue específicamente diseñado para su aplicación en imágenes médicas, con el objetivo de detectar estructuras complejas y sutiles. A pesar de su generalidad, logró resultados comparables a modelos altamente especializados entrenados desde cero para cada tipo de órgano o patología.

- **SAM-Med2D (2023)**: un *fine-tuning* sobre más de 4.6 millones de imágenes 2D médicas. Muestra una notable capacidad de generalización, funcionando de forma robusta en distintas modalidades (radiografías, ecografías, TACs), con una gran precisión incluso en tareas fuera del dominio de entrenamiento original.
- **SAMed, SAM-AutoMed, Medical SAM Adapter**: estas propuestas no re-entrenan completamente a SAM, sino que emplean técnicas eficientes como LoRA (*Low-Rank Adaptation*), módulos adaptadores o generación automática de *prompts* para facilitar su aplicación en entornos clínicos. Esta estrategia minimiza el coste computacional y permite una implementación más sencilla en contextos reales.
- **SAM-MedCls**: extiende la funcionalidad de SAM más allá de la segmentación, aplicándolo a tareas de clasificación médica. Esta capacidad multitarea demuestra el potencial de SAM como núcleo de modelos médicos generalistas, reduciendo la necesidad de múltiples redes independientes para tareas específicas.
- En paralelo, han aparecido nuevas propuestas que extienden esta idea:
 - **SAM3D**: lleva la arquitectura y filosofía de SAM al dominio volumétrico, permitiendo segmentación 3D (por ejemplo, en resonancias o tomografías), donde las relaciones espaciales entre cortes son cruciales.
 - **SegGPT**: inspirado en los modelos de lenguaje como GPT, SegGPT es un modelo multimodal que combina visión y lenguaje para realizar segmentación en escenarios *few-shot* (es decir, con muy pocos ejemplos de entrenamiento). Dado un par de imagen y segmentación de ejemplo, el modelo puede segmentar nuevos objetos en imágenes no vistas, lo que abre posibilidades para tareas con muy poca supervisión.

3. Segmentación Multimodal y Guiada por Texto

La segmentación multimodal se refiere al uso de múltiples tipos de entrada (modalidades), típicamente imagen y texto, para guiar el proceso de segmentación. A diferencia de los enfoques tradicionales basados exclusivamente en datos visuales, esta nueva línea de investigación permite que el modelo comprenda y actúe según instrucciones en lenguaje natural, descripciones clínicas o etiquetas semánticas. En el contexto médico, esto habilita una segmentación más semántica, flexible y accesible, especialmente útil en entornos clínicos donde la colaboración humano-máquina es clave.

- **CLIP (Contrastive Language–Image Pretraining, 2021)**: desarrollado por OpenAI, CLIP entrena conjuntamente representaciones de imagen y texto en un espacio embebido compartido. Esto permite buscar imágenes o regiones basadas en texto sin necesidad de entrenamiento supervisado.
- **MedCLIP**: es una versión adaptada de CLIP al dominio médico, entrenada con reportes clínicos, anotaciones especializadas e imágenes diagnósticas. Mejora la correspondencia entre texto médico e imagen, aumentando la precisión en escenarios clínicos reales, donde las descripciones son más técnicas y especializadas.
- **LISA y LLaVA-Med (2023)**: llevan esta capacidad aún más lejos al introducir la

comprensión de instrucciones detalladas en lenguaje natural. Estos modelos combinan visión con modelos grandes de lenguaje (LLMs), permitiendo al sistema interpretar comandos como “segmenta el contorno del ventrículo izquierdo”.

- **SaLIP (2024)**: representa un paso avanzado en la integración de modelos fundacionales. Combina la capacidad semántica de CLIP con el poder de segmentación precisa de SAM. SaLIP permite seleccionar regiones mediante texto y refinar las máscaras de manera precisa, sin necesidad de entrenamiento específico sobre datos médicos. Esta arquitectura híbrida habilita segmentación guiada por lenguaje con resultados de alta calidad incluso en dominios no vistos.

La integración entre visión y lenguaje está revolucionando la interacción de los profesionales de la salud con los sistemas de inteligencia artificial. En lugar de depender de interfaces rígidas o técnicas, ahora pueden emplearse instrucciones en lenguaje natural, como “resalta las zonas con hemorragia” o “detecta todas las lesiones hipodensas”.

4. Técnicas de Aprendizaje con Pocos Datos y Auto-Supervisión

Uno de los principales problemas en el campo de la segmentación médica es la escasez de datos anotados de calidad. Las anotaciones manuales requieren experiencia clínica especializada, lo que las hace costosas, lentas y, en muchos casos, inviables a gran escala. Para abordar esta limitación, han surgido enfoques de aprendizaje que reducen o eliminan la dependencia de etiquetas, agrupándose principalmente en dos categorías: auto-supervisión y *few-shot learning*.

- **Auto-supervisión (Self-Supervised Learning, SSL)**: estos métodos entrenan modelos utilizando señales inherentes a los datos, sin requerir anotaciones externas. Entre 2020 y 2023, modelos como **SimCLR**, **BYOL** y **MoCo** se adaptaron al dominio médico, lo que permitió preentrenar redes sobre grandes volúmenes de imágenes sin etiquetas.
- **SSL4MIS y CLADE (2022)**: son ejemplos de auto-supervisión específicamente diseñados para la segmentación médica. SSL4MIS emplea tareas auxiliares como reordenamiento espacial o reconstrucción para forzar al modelo a aprender detalles anatómicos. CLADE utiliza aprendizaje contrastivo guiado por información anatómica, dando prioridad a la preservación de estructuras clínicas importantes en las representaciones que aprende el modelo.
- **Few-shot learning y meta-learning**: estos enfoques buscan que un modelo generalice a nuevas tareas con muy pocos ejemplos (1 a 5 imágenes). Algoritmos como **MAML** (Model-Agnostic Meta-Learning) se combinaron con arquitecturas como **U-Net 3D** para lograr segmentación efectiva con datos mínimos, aprendiendo a adaptar rápidamente el modelo a nuevos órganos o patologías con escasa supervisión.
- **MisMatch y MedCoSS (2023)**: MisMatch utiliza diferentes vistas de una imagen con estructuras ligeramente alteradas para reforzar la robustez del modelo ante variaciones clínicas reales. MedCoSS introduce segmentación semi-supervisada sensible a la estructura, permitiendo entrenar con solo una fracción de datos anotados.

- **SSL-ALPNet (2022)**: representa una estrategia completamente auto-supervisada para segmentación, donde se generan *pseudo-máscaras* automáticamente como etiquetas suaves. El modelo aprende a segmentar sin intervención humana, utilizando estas máscaras generadas como guía inicial que luego es refinada progresivamente.

Estas técnicas no solo reducen el coste y la dependencia de anotaciones clínicas, sino que además promueven modelos más generales, adaptables y escalables.

5. Incorporación de Modelos Generativos y Difusos

A diferencia de los modelos discriminativos, que aprenden a clasificar o segmentar directamente, los modelos generativos son capaces de sintetizar nuevas muestras, rellenar regiones faltantes o refinar salidas de segmentación. Entre ellos destacan los modelos de difusión y las redes generativas antagónicas (GANs). Los modelos de difusión son una clase reciente de modelos generativos que aprenden a generar datos mediante la inversión de un proceso de adición gradual de ruido aplicado durante el entrenamiento. Inicialmente se agrega ruido a una imagen en múltiples pasos hasta que se convierte en puro ruido, y luego el modelo aprende a revertir ese proceso paso a paso.

- **DiffuseMorph, Diff-UNet, DiffMorph (2022–2023)**: representan una nueva generación de modelos para segmentación médica basados en procesos de difusión. Estas arquitecturas utilizan modelos de difusión para refinar contornos segmentados, suavizar bordes y garantizar que las regiones detectadas respeten la coherencia anatómica.
- **SEGGAD**: es un modelo híbrido que combina redes generativas adversariales (GANs) con modelos de difusión. Las GANs proporcionan una señal fuerte para mantener realismo estructural, mientras que la difusión añade robustez y refinamiento. SEGGAD ha sido aplicado exitosamente a la segmentación de anatomías con alta variabilidad interindividual, como órganos abdominales o estructuras vasculares finas.
- **RootNav 2.0**: aunque inicialmente diseñado para analizar sistemas radiculares en botánica, ha sido adaptado con éxito al análisis de estructuras biológicas complejas, como redes neuronales o sistemas vasculares en imágenes médicas. Su arquitectura generativa permite trazar trayectorias ramificadas complejas, ofreciendo una alternativa efectiva para segmentar estructuras alargadas y ramificadas que son difíciles de abordar con arquitecturas convencionales.

6. Aplicaciones de YOLO a Segmentación Médica

YOLO (You Only Look Once) es una familia de modelos de visión por computador originalmente diseñada para la tarea de detección de objetos en imágenes. Su principal ventaja es la capacidad de realizar predicciones en tiempo real, ya que trata la detección como un único problema, procesando toda la imagen en una sola pasada por la red.

En sus primeras versiones (YOLOv1–v3), el enfoque se centraba únicamente en la detección de cajas delimitadoras (*bounding boxes*). Sin embargo, versiones más recientes como YOLOv5, YOLOv7 y YOLOv8 han incorporado capacidades de segmentación de instancias, permitiendo identificar no solo la presencia y localización de objetos, sino también su contorno

exacto.

- **YOLOv5-seg, YOLOv7-mask y YOLOv8 (2020–2023)**: estas versiones se adaptaron para realizar segmentación en tiempo real. Al incorporar ramas de segmentación dentro de la misma arquitectura, pueden generar máscaras precisas de objetos o regiones de interés con alta eficiencia computacional.
- **Combinación con SAM** : recientes trabajos han explorado arquitecturas híbridas que combinan la rapidez de YOLO con la precisión morfológica de SAM. En estas propuestas, YOLO actúa como detector rápido de objetos de interés, y SAM se encarga de generar máscaras de segmentación refinadas.

En definitiva, el periodo 2020–2025 ha transformado radicalmente la segmentación de imágenes médicas, pasando de modelos entrenados por tarea y sensibles a los datos, a sistemas capaces de generalizar, eficientes en datos, interactivos y multimodales.

- Los transformers permiten capturar contexto global sin perder detalle.
- Modelos fundacionales como SAM, adaptados al contexto médico, ofrecen segmentación universal con mínimo reentrenamiento.
- Enfoques auto-supervisados y few-shot permiten entrenar con menos datos.
- Los modelos multimodales abren la puerta a sistemas clínicos más intuitivos.
- Modelos generativos y técnicas de difusión refinan y estabilizan segmentaciones difíciles.

2.1.4. Segmentación basada en U-Net

A medida que la segmentación de imágenes ha evolucionado, han surgido arquitecturas diseñadas específicamente para abordar los desafíos de precisión, eficiencia computacional y capacidad de generalización en distintos dominios. Entre estas, U-Net destacó como una de las arquitecturas más influyentes en el campo, particularmente en la segmentación de imágenes biomédicas, aunque su aplicabilidad se extendió a múltiples áreas de la visión por computador.

U-Net es un tipo de red neuronal convolucional introducida en 2015, específicamente desarrollada para llevar a cabo tareas de segmentación en imágenes médicas. Está formada por una estructura en forma de “U”, de ahí su nombre, donde está compuesta por un camino de contracción y abstracción de características a un espacio latente menor, seguido de otro camino de expansión de características concatenado con las características del camino de extracción de su mismo nivel, como muestra la Figura 2.5.

El camino de contracción (codificador) está compuesto por capas convolucionales 3×3 seguidas de funciones de activación ReLU y operaciones de *max-pooling* 2×2 , lo que permite extraer progresivamente características espaciales más abstractas a medida que se reduce la resolución de la imagen.

Por su parte, el camino de expansión (decodificador) emplea operaciones de sobremuestreo mediante convoluciones transpuestas (o *upsampling*), seguidas de convoluciones 2×2 y la

concatenación de mapas de activación provenientes del codificador en niveles equivalentes. Esta operación de *skip connections* permite recuperar información espacial fina que se pierde durante la compresión, y mejora la precisión de los contornos segmentados. Finalmente, se utiliza una convolución 1×1 en la capa de salida para reducir la dimensionalidad del mapa de características y generar el mapa de segmentación final.

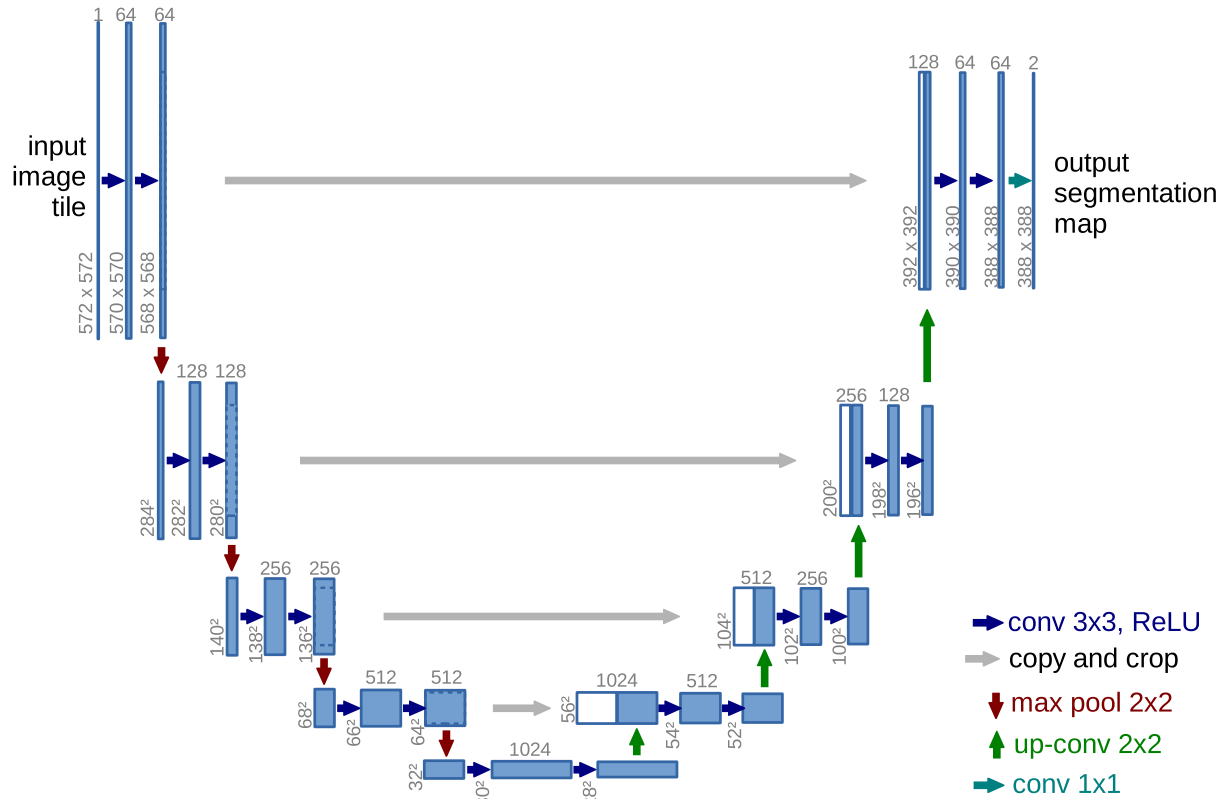


Figura 2.5: Arquitectura de la red U-Net para segmentación de imágenes biomédicas. Ronneberger et al. [36]

Sin embargo, a pesar de sus numerosas ventajas y su amplio uso en el ámbito de la segmentación médica, la arquitectura U-Net presenta también ciertas limitaciones que han motivado el desarrollo de múltiples variantes posteriores. Estas limitaciones incluyen tanto restricciones computacionales como desafíos inherentes a la calidad de segmentación en casos complejos:

- **Uso elevado de memoria.** U-Net tiene una arquitectura simétrica con muchas conexiones de concatenación, lo que incrementa el consumo de memoria.
- **Pérdida de información en los bordes.** Debido a las convoluciones sin relleno (*valid convolutions*), la imagen de salida es más pequeña que la de entrada.
- **Dificultad en la segmentación de objetos muy pequeños o con poca separación.** U-Net tiene dificultades para segmentar objetos muy pequeños o con poca separación, como células muy juntas en imágenes biomédicas.

- **Dependencia de los datos de entrenamiento.** Si los datos de entrenamiento no tienen suficiente variedad, el modelo puede no generalizar correctamente y cometer errores en imágenes nuevas.
- **Falta de procesamiento contextual a largo alcance.** Aunque U-Net captura información de múltiples escalas, tiene limitaciones para capturar relaciones espaciales a gran escala.

2.1.5. Nueva arquitectura: xLSTM(*Extended LSTM*)

Aunque U-Net es eficaz en segmentación de imágenes, presenta limitaciones para capturar dependencias de largo alcance, lo cual es crucial en contextos como la medicina.

Para abordar este desafío, se exploró la integración de mecanismos de memoria como LSTM, capaces de retener información a largo plazo. Sin embargo, a pesar de su efectividad, las LSTM se reemplazaron en gran parte por arquitecturas más recientes como los Transformers, que ofrecen mayor eficiencia y escalabilidad.

Recientemente, el trabajo de Beck et al. (2024) [5] ha propuesto una variante mejorada denominada xLSTM (*Extended LSTM*), diseñada para competir con modelos de última generación en términos de eficiencia y capacidad de modelado. Esta nueva arquitectura introduce mejoras clave sobre la LSTM tradicional, optimizando la retención de memoria y la capacidad de procesamiento en paralelo. En dicho trabajo se desarrollan dos variantes:

- **sLSTM (scalar LSTM):** Mantiene la memoria escalar tradicional pero introduce una nueva técnica de mezcla de memoria.
- **mLSTM (Matrix LSTM):** Utiliza una memoria basada en matrices y una regla de actualización de covarianza, lo que permite una mayor capacidad de almacenamiento, procesamiento en paralelo y un mejor manejo de datos complejos como tokens raros en modelado de lenguaje.

La integración de xLSTM en arquitecturas convolucionales como U-Net representa un avance significativo, al permitir una mejor captación de dependencias contextuales de largo alcance. Esto ha dado lugar al desarrollo de arquitecturas híbridas, como xLSTM-UNet, que combinan lo mejor de ambos enfoques para lograr segmentaciones más precisas y adaptadas a contextos complejos, especialmente en el ámbito biomédico y científico.

2.1.6. Segmentación basada en U-Net y xLSTM

El 24 de junio de 2024, se propuso un nuevo modelo denominado U-VixLSTM, el cual combina Redes Neuronales Convolucionales (CNNs) con Vision Extended Long Short-Term Memory (Vision-xLSTM) para mejorar la segmentación de imágenes médicas en 2D y 3D [10].

Este modelo, basado en la arquitectura U-Net, sustituye los módulos tradicionales por bloques xLSTM, los cuales destacan por su capacidad para capturar dependencias de largo alcance con una complejidad computacional lineal.

A diferencia de los métodos basados en *Vision Transformers* o modelos como *Mamba*, el xLSTM-UNet mejora la capacidad de análisis secuencial de las imágenes médicas mediante la integración de múltiples bloques xLSTM, permitiendo una extracción de características más robusta en distintas escalas.

Se han propuesto dos arquitecturas dentro de este enfoque:

- **xLSTM-UNet_Bot**: Introduce *mLSTM* exclusivamente en el *bottleneck* del modelo.
- **xLSTM-UNet_Enc**: Incorpora *mLSTM* en capas del codificador, lo que permite mejorar la captura de dependencias espaciales en diferentes niveles de abstracción (véase Figura 2.6).

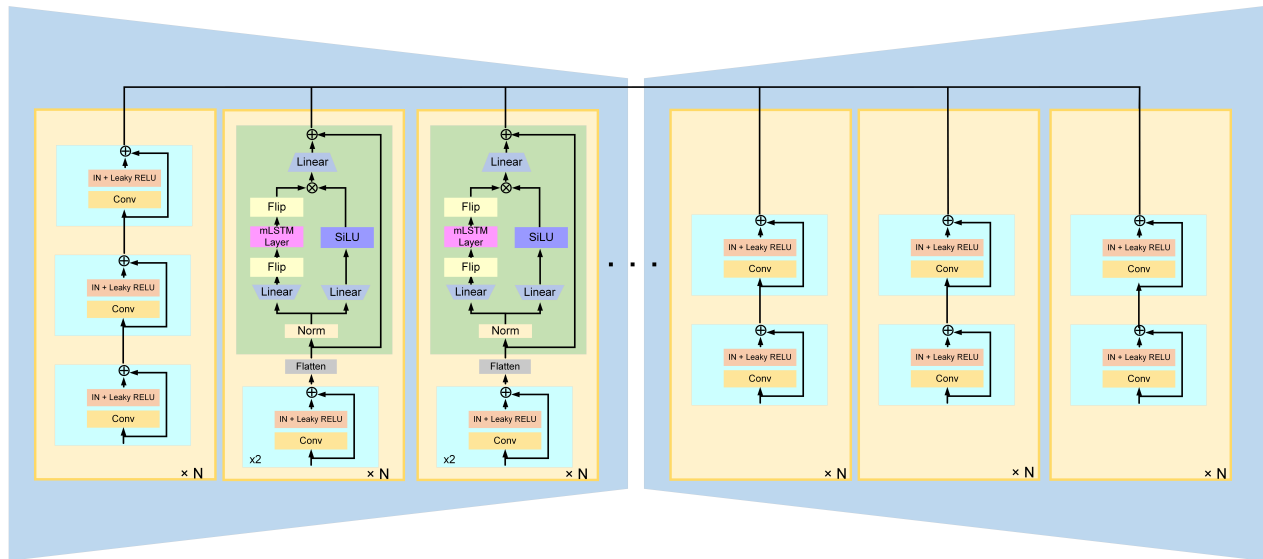


Figura 2.6: Arquitectura del modelo xLSTM-UNet (*xLSTM-UNet_Enc*). Imagen extraída de Chen et al. [10].

Los resultados experimentales demostraron que ambas variantes de *xLSTM-UNet* superan a modelos de referencia en tareas de segmentación médica en 2D y 3D, incluyendo imágenes de resonancia magnética abdominal, endoscopias, imágenes microscópicas de células y volúmenes de resonancia magnética cerebral. Así, este modelo se posiciona como una solución eficiente y precisa para la segmentación de imágenes biomédicas, mejorando la interpretación y el análisis de datos médicos a gran escala.

En línea con estos avances, el 1 de febrero de 2025 se presentó una nueva arquitectura basada en xLSTM que extiende aún más el enfoque anterior llamada ViLU-Net (*Vision LSTM U-Net*).

En esta propuesta, los bloques mLSTM no solo se integran en el codificador, como en la arquitectura mostrada en la Figura 2.6, sino también en el decodificador, permitiendo una transmisión más eficaz de las dependencias espaciales a lo largo de toda la red. Esta mejora refuerza la capacidad del modelo para capturar patrones contextuales tanto en la compresión como en la reconstrucción de las imágenes médicas.

Dicha arquitectura se implementó y evaluó en tareas de segmentación de órganos abdominales y tumores retroperitoneales, obteniendo mejoras significativas en métricas clave respecto a variantes anteriores de U-Net, como señalan los autores [16]. La estructura detallada del modelo, denominado ViLU-Net, se muestra en la Figura 2.7, donde se representa tanto la arquitectura general como los componentes internos que lo conforman.

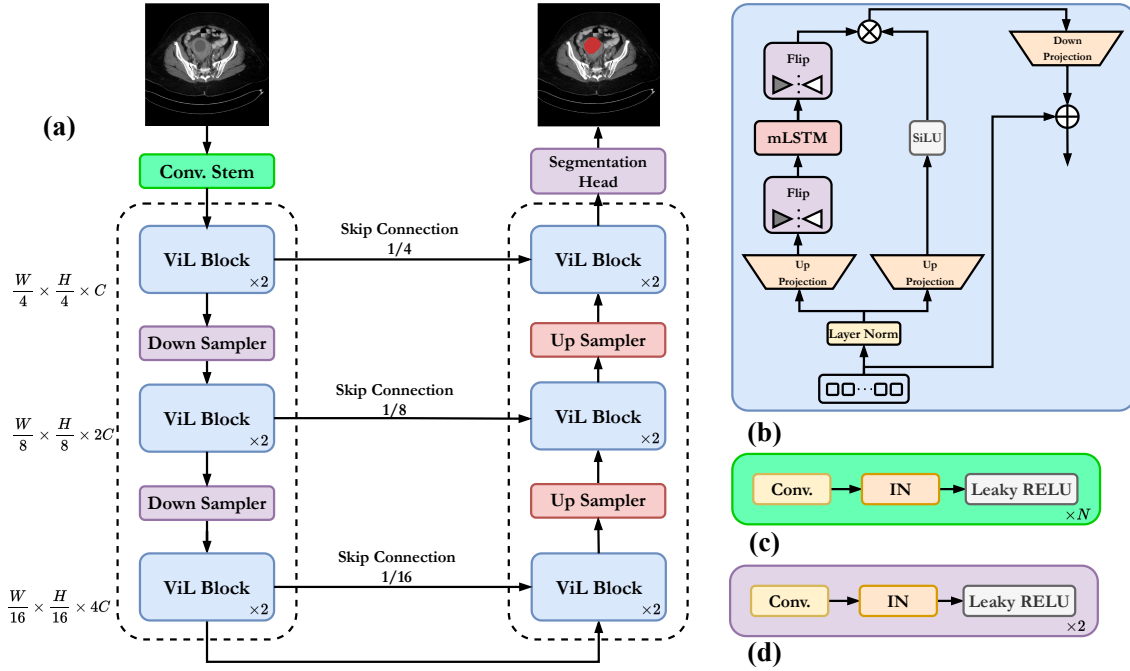


Figura 2.7: Esquema general de la arquitectura propuesta ViLU-Net, presentada por Heidari et al. [16]. (a) Diagrama general del modelo, (b) estructura interna del bloque ViL, (c) módulo de convolución inicial (Conv. Stem), y (d) bloques de muestreo hacia arriba y abajo (Up/Down Sampler). IN representa una operación de normalización por instancia (Instance Normalization).

Capítulo 3

Competencias y Sostenibilidad

3.1. Competencias específicas

Las competencias específicas son aprendizajes clave vinculados al contenido de la titulación. A continuación, se muestra cómo las más relevantes para el Trabajo Fin de Grado se desarrollaron y aplicaron a lo largo del proyecto.

- **ED1 - Diseño e implementación de sistemas inteligentes:** se desarrolló de forma integrada a lo largo del diseño e implementación de un sistema inteligente orientado a la segmentación de imágenes médicas, utilizando arquitecturas avanzadas como U-Net combinada con xLSTM.
- **ED2 - Adquisición y representación de conocimiento:** El proyecto implica la comprensión y modelado de estructuras médicas complejas (órganos, instrumentos de endoscopia, tumores cerebrales, etc.) de forma que puedan ser interpretadas por sistemas automáticos, facilitando así su análisis y uso en entornos clínicos.
- **ED3 - Aplicación de técnicas de aprendizaje automático:** Se aplicaron técnicas de inteligencia artificial, especialmente aprendizaje profundo, para extraer información clínica de imágenes médicas mediante modelos entrenados con datos anónimos y evaluados para garantizar precisión y fiabilidad.
- **ED4 - Análisis y resolución de problemas en IA aplicada:** se abordó un problema real en el ámbito médico, la segmentación precisa de imágenes, desde la identificación del reto hasta la implementación, validación y documentación de una solución basada en técnicas de inteligencia artificial.
- **ED6 - Análisis y modelado de datos en ciencia de datos:** Se abordó el tratamiento de datos complejos incluyendo análisis exploratorio, preprocesamiento, modelado y evaluación cuantitativa. Todo ello orientado a generar conocimiento útil y aplicable en contextos clínicos.

3.2. Alineamiento con los objetivos de desarrollo sostenible

El uso de arquitecturas híbridas de inteligencia artificial en la segmentación de imágenes médicas representa un avance tecnológico en el diagnóstico automatizado y aporta significativamente al desarrollo sostenible, alineándose con varios Objetivos de Desarrollo Sostenible (ODS) recogidos en la Tabla 3.1.

Cuadro 3.1: Grado de relación del TFG con los objetivos de desarrollo sostenible [34].

ODS	Grado de relación			
	0 No procede	1 Bajo	2 Medio	3 Alto
1 Fin de la pobreza	✓			
2 Hambre cero	✓			
3 Salud y bienestar				✓
4 Educación de calidad			✓	
5 Igualdad de género	✓			
6 Agua limpia y saneamiento	✓			
7 Energía asequible y no contaminante	✓			
8 Trabajo decente y crecimiento económico	✓			
9 Industria, innovación e infraestructuras				✓
10 Reducción de las desigualdades	✓			
11 Ciudades y comunidades sostenibles	✓			
12 Producción y consumo sostenibles	✓			
13 Acción por el clima	✓			
14 Vida submarina	✓			
15 Vida de ecosistemas terrestres	✓			
16 Paz, justicia e instituciones sólidas	✓			
17 Alianzas para lograr objetivos		✓		

- **ODS 3 - Salud y bienestar (Alto):** Este proyecto contribuye al objetivo de garantizar una vida sana mediante un sistema de segmentación de imágenes médicas basado en aprendizaje profundo, que mejora la precisión, velocidad y eficiencia del diagnóstico clínico. Al automatizar parte del análisis, reduce la carga del personal médico, acorta los tiempos de evaluación y optimiza la detección de patologías.
- **ODS 4 - Educación de calidad (Medio):** Esta iniciativa apoya una educación inclusiva y de calidad mediante un enfoque práctico y de investigación en inteligencia artificial médica. Utiliza herramientas accesibles, datos abiertos y modelos públicos, facilitando el aprendizaje colaborativo y sin barreras para estudiantes y profesionales.
- **ODS 9 - Industria, innovación e infraestructuras (Alto):** La incorporación de innovaciones tecnológicas resulta especialmente relevante cuando se aplican al ámbito médico. Este enfoque promueve el desarrollo de soluciones avanzadas mediante un sistema que segmenta imágenes médicas utilizando arquitecturas híbridas. Además, gracias a su diseño, que es a la vez escalable y portátil, este sistema puede implementarse en centros que tienen recursos limitados.
- **ODS 17 - Alianzas para lograr objetivos (Bajo):** El proyecto apoya la ciencia abierta y la cooperación internacional mediante el uso de datos y repositorios compartidos. Su participación en iniciativas como MICCAI, NeurIPS o BraTS fortalece redes de colaboración, facilita el intercambio de buenas prácticas y promueve la estandarización de herramientas en inteligencia artificial médica.

Capítulo 4

Modelos utilizados

4.1. Matrix Long Short-Term Memory (mLSTM)

Tal y como se indicó previamente, el núcleo de la arquitectura xLSTM (*Extended Long Short-Term Memory*) reside en dos modificaciones clave aplicadas a la memoria de la LSTM tradicional: por un lado, la sLSTM, que mantiene una celda de memoria escalar con mezcla entre memorias; por otro lado, la mLSTM, que introduce una celda de memoria matricial y una regla de actualización basada en el producto exterior (covarianza), lo que permite una implementación completamente paralelizable y de gran capacidad [5].

La mLSTM reemplaza el escalar de estado de celda $c_t \in \mathbb{R}$ por una matriz de memoria $C_t \in \mathbb{R}^{d \times d}$. Esta estructura permite almacenar relaciones ricas entre claves y valores mediante una operación de producto exterior y realizar recuperaciones eficientes a través de multiplicación matricial. El cálculo comienza con la proyección de la entrada x_t en tres vectores: clave k_t , valor v_t y consulta q_t :

$$k_t = \frac{1}{\sqrt{d}} W_k x_t + b_k \quad (4.1)$$

$$v_t = W_v x_t + b_v \quad (4.2)$$

$$q_t = W_q x_t + b_q \quad (4.3)$$

A continuación, se calculan las puertas de entrada y olvido:

$$i_t = \exp(\tilde{i}_t), \quad \tilde{i}_t = w_i^\top x_t + b_i \quad (4.4)$$

$$f_t = \sigma(\tilde{f}_t) \text{ OR } \exp(\tilde{f}_t), \quad \tilde{f}_t = w_f^\top x_t + b_f \quad (4.5)$$

donde:

- $x_t \in \mathbb{R}^d$ es el vector de entrada en el instante de tiempo t ,
- $w_i, w_f \in \mathbb{R}^d$ son vectores de pesos entrenables que proyectan la entrada x_t para calcular las preactivaciones de las puertas,
- $b_i, b_f \in \mathbb{R}$ son los términos de sesgo (*bias*) asociados a cada puerta.

La celda de memoria se actualiza guardando la relación entre un valor y su clave mediante un producto exterior:

$$C_t = f_t \cdot C_{t-1} + i_t \cdot (v_t k_t^\top) \quad (4.6)$$

Simultáneamente, se mantiene un estado normalizador n_t , que acumula las claves ponderadas:

$$n_t = f_t \cdot n_{t-1} + i_t \cdot k_t \quad (4.7)$$

Para recuperar la información, se proyecta la consulta q_t sobre la memoria, y se normaliza con el producto punto entre n_t y q_t , el cual se acota inferiormente para evitar inestabilidades numéricas:

$$\tilde{h}_t = \frac{C_t q_t}{\max(|n_t^\top q_t|, 1)} \quad (4.8)$$

Finalmente, se calcula la salida del bloque mediante una puerta de salida:

$$o_t = \sigma(\tilde{o}_t), \quad \tilde{o}_t = W_o x_t + b_o \quad (4.9)$$

$$h_t = o_t \odot \tilde{h}_t \quad (4.10)$$

La Figura 4.1 muestra de forma esquemática esta arquitectura interna de mLSTM, destacando el flujo de información desde la entrada hasta la recuperación de la salida mediante consultas.

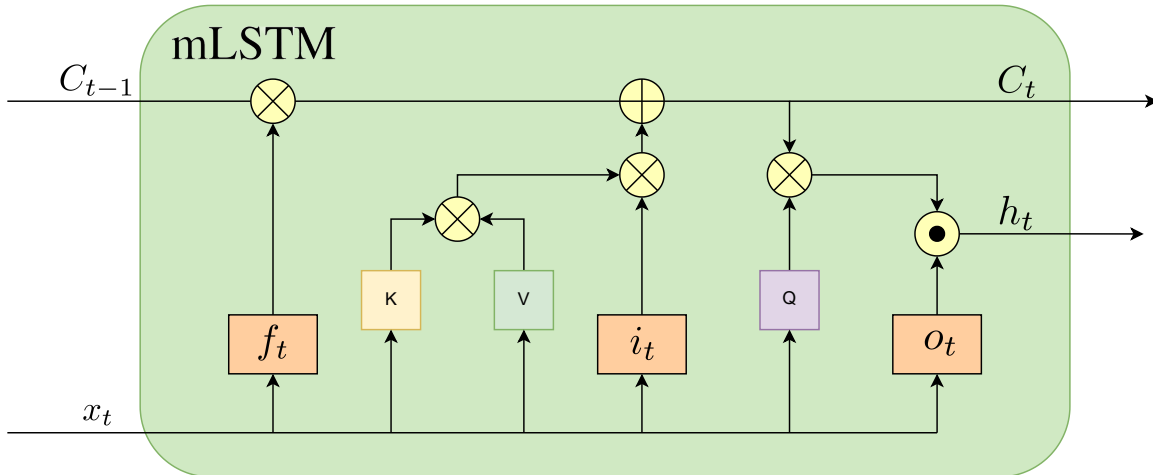


Figura 4.1: Diagrama del funcionamiento interno del bloque mLSTM. Se muestra cómo se combinan la memoria matricial, las puertas y la recuperación mediante consulta.

4.2. Adaptación de la mLSTM

Una vez explicada la mLSTM original, este apartado presenta una adaptación específica basada en el trabajo de [10], diseñada para tareas de visión por computador. Aunque se conservan los principios fundamentales de la mLSTM, como la memoria matricial y la actualización por covarianza, esta versión introduce ligeras modificaciones operativas que se detallan a continuación.

En particular, la mLSTM adquiere una relevancia especial en tareas de visión por computador, a través de la arquitectura Vision-xLSTM (ViL), una adaptación de xLSTM diseñada específicamente para este dominio. El bloque mLSTM puede integrarse en distintos componentes de la arquitectura, ya sea el codificador, el decodificador o el cuello de botella, y comienza procesando la entrada mediante la proyección en vectores de consulta, clave y valor (Q, K, V) :

$$Q = XW_Q, \quad K = XW_K, \quad V = XW_V \quad (4.11)$$

donde W_Q , W_K y W_V son matrices de pesos entrenables de dimensión $2Z \times d$, siendo d la dimensión proyectada. Estos vectores son posteriormente procesados por subcapas mLSTM, compuestas por múltiples cabezas de atención, cada una equipada con su propia memoria matricial.

El funcionamiento interno de la mLSTM comienza con el cálculo de las preactivaciones de las puertas de entrada y olvido:

$$\tilde{i} = W_I[Q, K, V] + B \quad (4.12)$$

$$\tilde{f} = W_F[Q, K, V] + B \quad (4.13)$$

donde $B \in \mathbb{R}^{N \times d_h}$ es una matriz de sesgo, y W_I, W_F son matrices de pesos. A partir de estas, se calcula una matriz de decaimiento de puertas.

$$D = \tilde{i} \oplus \log \sigma(\tilde{f}) \quad (4.14)$$

Aquí, $\log \sigma$ denota la activación log-sigmoide, y $\oplus$ representa la combinación elemento a elemento. Para garantizar la estabilidad numérica en los siguientes pasos, la matriz D se estabiliza aplicando una función exponencial:

$$\hat{D} = \exp(D) \quad (4.15)$$

Las puntuaciones de atención $S \in \mathbb{R}^{N \times N}$ se obtienen a partir del producto escalar entre los vectores Q y K , siguiendo un esquema similar al mecanismo de autoatención presente en los Transformers. Para preservar la direccionalidad temporal, se aplica una máscara causal $M \in \mathbb{R}^{N \times N}$, que restringe la atención exclusivamente a los parches anteriores al actual. El cálculo final de la atención consiste en normalizar el producto $QK^\top$ y aplicar dicha máscara:

$$S = \text{Softmax} \left(\frac{QK^\top}{\sqrt{d}} + M \right) \quad (4.16)$$

La matriz de puntajes de atención se somete a una multiplicación elemento a elemento con $\hat{D}$, para formar la matriz de combinación $C \in \mathbb{R}^{N \times d_h}$.

$$C = S^\top \hat{D} \quad (4.17)$$

Finalmente, el estado de celda actualizado se calcula mediante una operación de producto exterior entre la matriz de combinación C y los vectores de valor V :

$$\tilde{h}_t = C \otimes V \quad (4.18)$$

La Figura 4.2 ilustra de forma esquemática la arquitectura del bloque mLSTM, destacando la organización en múltiples cabezas de atención, cada una con su correspondiente memoria matricial. Esta representación visual permite comprender mejor el flujo de información entre los distintos componentes, así como la manera en que se combinan atención y memoria en el marco de la arquitectura Vision-xLSTM. En conjunto, el bloque mLSTM ofrece una solución potente y flexible para modelar dependencias complejas en tareas visuales, consolidándose como un componente clave dentro de las nuevas variantes de xLSTM.

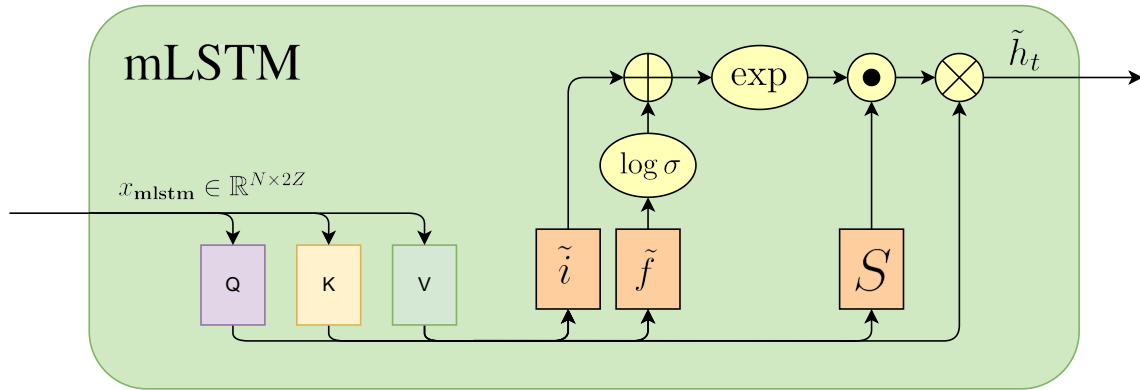


Figura 4.2: Arquitectura del bloque mLSTM con múltiples cabezas de atención y memoria matricial adaptada.

A modo de síntesis, la versión original de la mLSTM (Figura 4.1) y su adaptación para visión por computador (Figura 4.2) comparten elementos conceptuales clave, como el uso de una memoria matricial y la actualización basada en productos exteriores. Sin embargo, presentan diferencias importantes en su diseño y funcionalidad.

4.3. ViL Block (*Vision-xLSTM Block*)

Tras introducir el funcionamiento interno del bloque mLSTM, es fundamental entender cómo este se integra dentro de las arquitecturas específicas. El *ViLBlock* actúa como un componente esencial dentro de la arquitectura xLSTM-UNet para tareas de segmentación biomédica en 2D y 3D [10]. Su diseño comienza con una normalización de la entrada y una proyección a una dimensión ampliada:

$$x \in \mathbb{R}^{N \times Z} \xrightarrow{\text{LayerNorm}} y \in \mathbb{R}^{N \times Z} \xrightarrow{\text{Proyección}} x_{\text{proj}} \in \mathbb{R}^{N \times 2Z} \quad (4.19)$$

donde cada vector de entrada $x_i \in \mathbb{R}^Z$ se normaliza mediante normalización por capas (*Layer Normalization*). Esta consiste en calcular la media y la desviación estándar de las activaciones de un conjunto de neuronas dentro de una capa, ajustándolas para que tengan una distribución con media cero y varianza unitaria. Con ello se mejora la estabilidad numérica y se acelera la convergencia durante el entrenamiento:

$$\mu_i = \frac{1}{Z} \sum_{j=1}^Z x_{ij}, \quad \sigma_i^2 = \frac{1}{Z} \sum_{j=1}^Z (x_{ij} - \mu_i)^2 \quad (4.20)$$

$$\hat{x}_{ij} = \frac{x_{ij} - \mu_i}{\sqrt{\sigma_i^2 + \epsilon}}, \quad y_{ij} = \gamma_j \cdot \hat{x}_{ij} + \beta_j \quad (4.21)$$

Posteriormente, se aplica una proyección lineal que expande la dimensionalidad de cada vector, proyectándolo a un espacio de incrustamiento duplicado.

$$x_{\text{proj},i} = W y_i + b \quad (4.22)$$

con $W \in \mathbb{R}^{2Z \times Z}$ y $b \in \mathbb{R}^{2Z}$, resultando en una salida $x_{\text{proj}} \in \mathbb{R}^{N \times 2Z}$.

La señal proyectada se bifurca en dos caminos paralelos: $x_{\text{mLSTM}} \in \mathbb{R}^{N \times 2Z}$ y $y \in \mathbb{R}^{N \times 2Z}$. El primero se somete a una convolución 1D causal seguida de una activación *SiLU*:

$$x_{\text{conv}} = \text{SiLU}(\text{CausalConv1D}(x_{\text{mLSTM}})) \quad (4.23)$$

La convolución causal asegura que cada salida y_t dependa solo de entradas pasadas o presentes respetando el orden de la información, y la activación SiLU introduce una no linealidad suave que, a diferencia de funciones como ReLU, no anula los valores negativos por completo, lo que puede facilitar una mejor propagación del gradiente y un aprendizaje más estable:

$$x_{\text{conv},t} = \sum_{j=0}^{k-1} w_j \cdot x_{t-j}, \quad \text{donde } x_{t-j} = 0 \text{ si } t-j < 0 \quad (4.24)$$

$$\text{SiLU}(x_{\text{conv},t}) = x_{\text{conv},t} \cdot \sigma(x_{\text{conv},t}) \quad (4.25)$$

El resultado x_{conv} se alimenta al bloque mLSTM, que genera una representación de memoria enriquecida $\tilde{h}_t$. Cabe señalar que el funcionamiento detallado de esta subcapa se describió previamente en la Sección 4.2, donde se explica su dinámica basada en atención, puertas y memorias matriciales:

$$\tilde{h}_t = \text{mLSTM}(x_{\text{conv}}) \quad (4.26)$$

Posteriormente, se aplica una conexión residual modulada mediante un escalado entrenable β , denominado *Learnable Residual Scaling*, que permite modificar de forma aprendible el impacto del camino original x_{conv} en la salida del bloque. Esta operación queda definida como:

$$\tilde{h}_{\text{skip}} = \tilde{h}_t + \beta \cdot x_{\text{conv}} \quad (4.27)$$

Finalmente, se realiza una combinación con el segundo camino y , el cual se sometió a una función de activación SiLU, mediante un producto elemento a elemento, seguida de una proyección lineal descendente que restaura la dimensionalidad original Z :

$$o = W_D^\top (\tilde{h}_{\text{skip}} \otimes \text{SiLU}(y)) \quad (4.28)$$

donde $\tilde{h}_{\text{skip}} = x_{\text{conv}} \in \mathbb{R}^{N \times 2Z}$ y $y \in \mathbb{R}^{N \times 2Z}$. La operación $\otimes$ denota el producto Hadamard (elemento a elemento), aplicado sobre cada vector de características. Esta proyección asegura que la salida del bloque $o_{\text{final}} \in \mathbb{R}^{N \times Z}$ conserve la dimensionalidad original de la entrada, permitiendo su integración directa en el flujo del modelo.

Una representación esquemática del flujo de datos dentro del ViLBlock se muestra a continuación en la Figura 4.3. En ella se visualizan los principales componentes descritos anteriormente, la normalización inicial, la proyección ascendente, la bifurcación de caminos, la subcapa mLSTM, y la combinación residual escalada, facilitando así la comprensión del recorrido secuencial que sigue la información a lo largo del bloque.

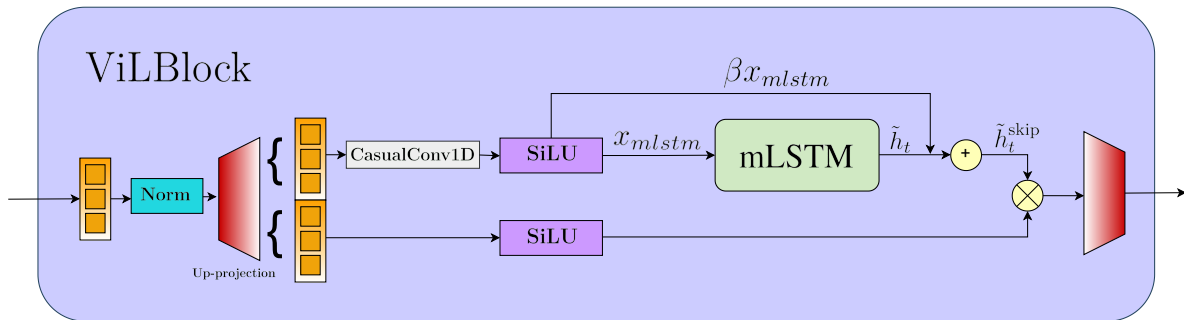


Figura 4.3: Esquema del ViLBlock con bifurcación de caminos, subcapa mLSTM, y combinación residual escalada.

4.4. Arquitecturas basadas en U-Net y xLSTM

Una vez explicados los componentes principales, es fundamental presentar dos bloques adicionales sobre los que se construye la arquitectura xLSTM-UNet: el *BasicResBlock* y el *BasicBlockD*.

El primero implementa un bloque residual clásico compuesto por dos convoluciones seguidas de normalización y activación (LeakyReLU), junto con una proyección opcional mediante convolución 1×1 en la rama residual, tal como se muestra en la Figura 4.4. Cabe destacar que este bloque no reduce la dimensionalidad espacial de la entrada, ya que todas las convoluciones se aplican con *stride* unitario, y se utiliza tanto en el codificador como en el decodificador de la arquitectura xLSTM-UNet para preservar y refinar las representaciones internas sin pérdida de resolución.

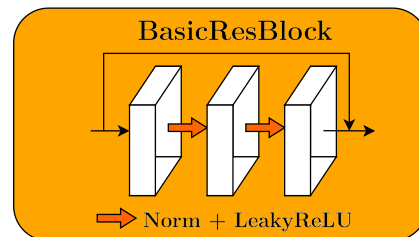


Figura 4.4: Arquitectura del bloque *BasicResBlock*.

Por otro lado, el bloque *BasicBlockD* está compuesto por los mismos elementos fundamentales que el *BasicResBlock*, es decir, secuencias de convolución, normalización y activación, con la diferencia clave de que no incorpora conexión residual. Se trata de un bloque puramente secuencial, compuesto por dos capas convolucionales encadenadas. A diferencia del *BasicResBlock*, este bloque sí puede modificar la resolución espacial de la entrada en función de los parámetros utilizados (por ejemplo, el valor del *stride*), lo que lo hace adecuado para las etapas de reducción o transformación dimensional. Su estructura se muestra en la Figura 4.5.

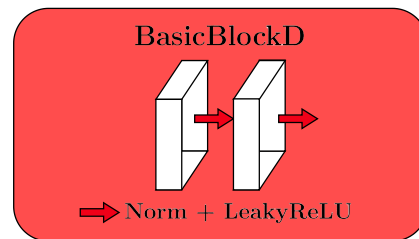


Figura 4.5: Arquitectura del bloque *BasicBlockD*.

Una vez descritos todos los componentes principales, es posible presentar las arquitecturas completas. Las variantes de xLSTM-UNet introducidas a continuación comparten una estructura común compuesta por un codificador con un bloque inicial seguido de siete etapas jerárquicas, un cuello de botella que actúa como zona de máxima compresión y representación abstracta de las características, y un decodificador constituido por seis etapas que aplican operaciones de sobremuestreo (*upsampling*) y se conectan mediante enlaces de tipo *skip connection* con las salidas correspondientes del codificador. Cabe destacar que las tres arquitecturas que se presentan a continuación, *xLSTM-UNet_Bot*, *xLSTM-UNet_Enc* y *ViLU-Net*, se diferencian únicamente en la localización del bloque que contiene la capa xLSTM, lo cual da lugar a sus respectivas denominaciones.

4.4.1. xLSTM-UNet_Bot

En primer lugar, el nombre de esta arquitectura xLSTM-UNet_Bot viene de que el bloque ViLBlock se encuentra en el cuello de botella (Bottleneck). El codificador y decodificador tienen las mismas características descritas en el párrafo anterior.

La Figura 4.6 muestra un esquema simplificado de la arquitectura, en el que se observa que el bloque *ViLBlock* se encuentra en el cuello de botella, justo después de la última etapa del codificador y antes del inicio del decodificador. El esquema no representa todas las etapas: se muestra únicamente el bloque inicial y las tres primeras etapas del codificador, seguidas de puntos suspensivos que indican su continuación. Del lado del decodificador, se incluyen las tres últimas etapas, siendo la salida del último bloque el mapa de segmentación final.

En el diagrama, las flechas azules representan el flujo interno de información entre bloques dentro de la misma etapa, mientras que las flechas rojas indican un cambio de etapa, usualmente acompañado de una reducción en la resolución espacial realizada por el bloque *BasicBlockD*. Las flechas moradas, por su parte, corresponden a las operaciones de sobremuestreo que permiten duplicar la resolución espacial al mismo tiempo que ajustan la dimensión de los canales de los mapas de características.

Los bloques naranjas corresponden al *BasicResBlock*, mostrado en la Figura 4.4. Por su parte, los bloques rojos representan el *BasicBlockD*, mostrado en la Figura 4.5. Los bloques blancos representan conexiones directas entre codificador y decodificador, conocidas como *skip connections*, y los bloques lilas corresponden a la salida de la etapa anterior del decodificador antes de su combinación con la información del codificador.

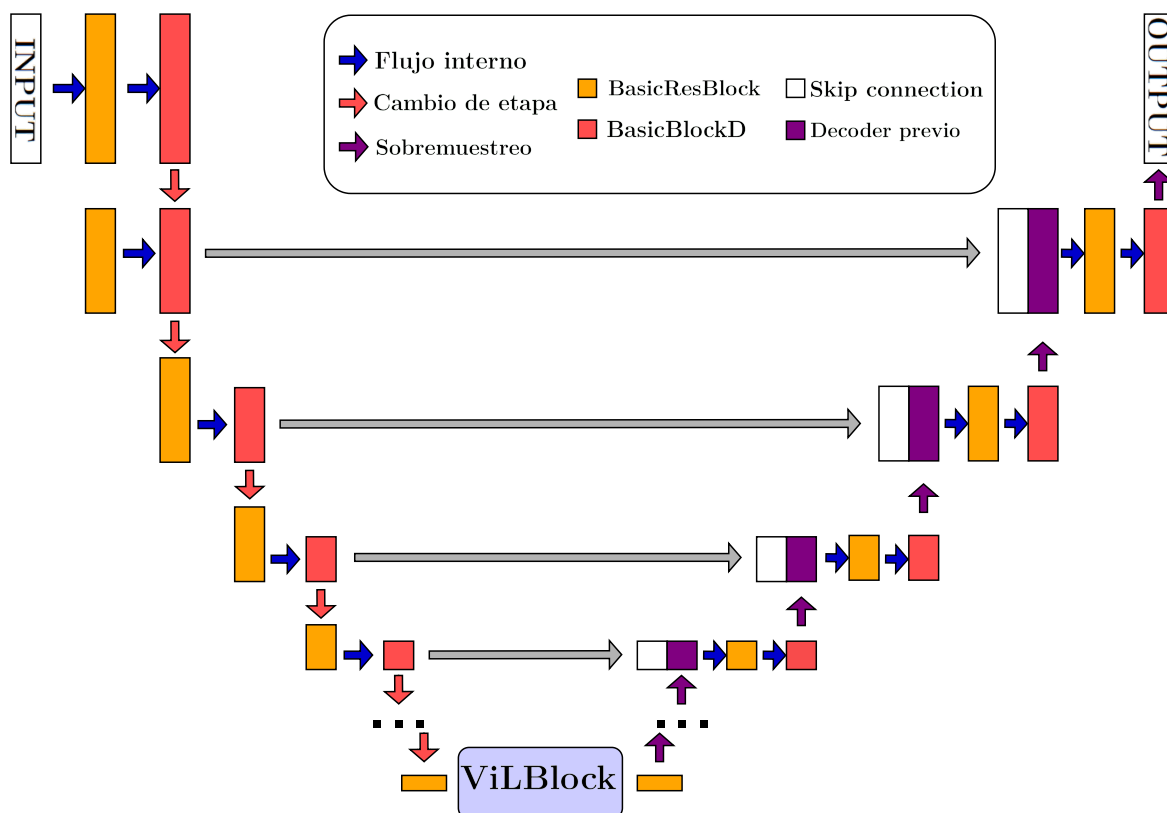


Figura 4.6: Arquitectura de $xLSTM-UNet_Bot$.

4.4.2. xLSTM-UNet_Enc

La arquitectura *xLSTM-UNet_Enc* se caracteriza por ubicar los bloques *ViLBlock* de forma intercalada a lo largo del codificador. En concreto, estos bloques se insertan en las etapas 2, 4 y 6 del codificador, lo que permite una integración progresiva de mecanismos de atención y modelado temporal a medida que se profundiza. Esta disposición se diferencia de la variante *Bot*, donde el bloque *ViLBlock* se encuentra únicamente en el cuello de botella.

Además, en esta variante se introduce un bloque adicional del tipo *BasicBlockD* en la etapa 1 del codificador, lo que refuerza la capacidad de extracción de características en las primeras etapas del procesamiento.

La Figura 4.7 muestra un esquema simplificado de la arquitectura *xLSTM-UNet_Enc*. Al igual que en la figura anterior, se omitieron algunas etapas por cuestiones de claridad, manteniéndose el bloque inicial y las cinco primeras etapas del codificador, así como las cinco últimas etapas del decodificador. La presencia de puntos suspensivos en el diagrama indica la existencia de etapas intermedias no representadas explícitamente.

En el diagrama, las convenciones gráficas se mantienen consistentes con la Figura 4.6: flechas azules para flujo interno, flechas rojas para cambios de etapa con posible reducción espacial, y flechas moradas para las operaciones de sobremuestreo. Los bloques naranjas, rojos, lilas y blancos también conservan su misma interpretación como *BasicResBlock*, *BasicBlockD*, salida previa del decodificador y *skip connection*, respectivamente. Los bloques azul claro representan los *ViLBlock*, cuya ubicación intercalada en el codificador es distintiva de esta arquitectura.

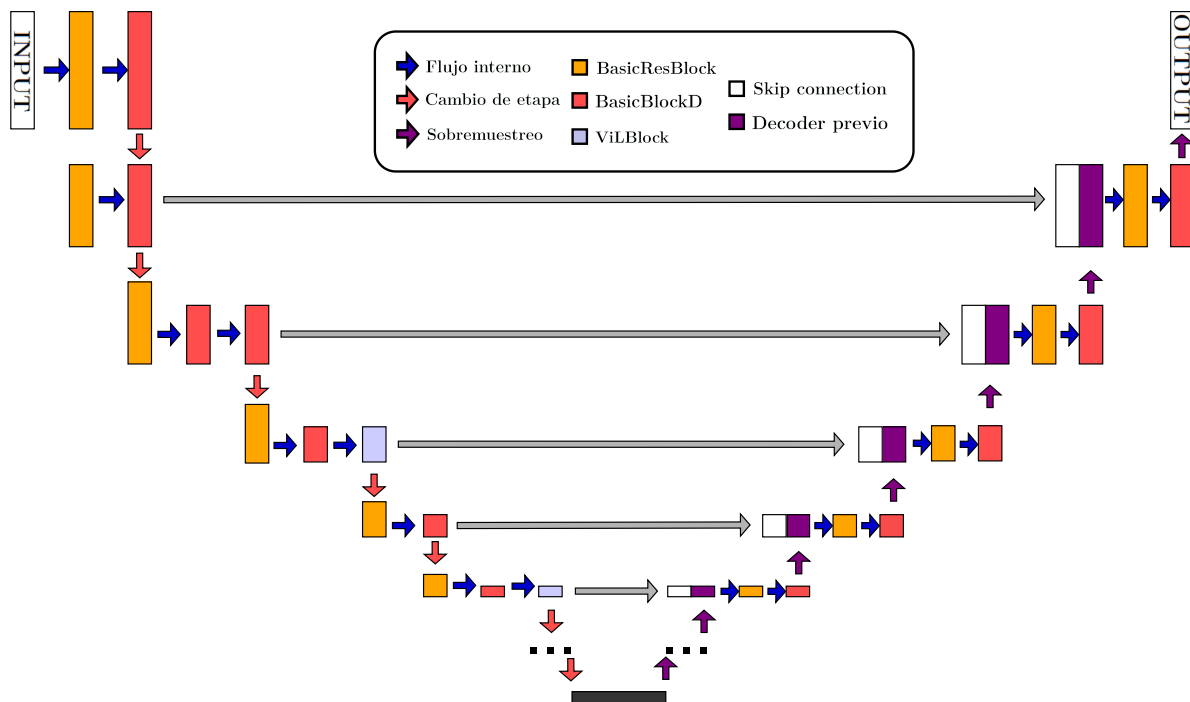


Figura 4.7: Arquitectura de *xLSTM-UNet_Enc*.

4.4.3. xLSTM-UNet_EncDec

La arquitectura *xLSTM-UNet_EncDec* se caracteriza por integrar bloques *ViLBlock* tanto en el codificador como en el decodificador, lo que da origen a su nombre. A diferencia de las variantes anteriores, esta configuración busca capturar dependencias espaciales y temporales en ambas fases del modelo. Específicamente, se incorporan bloques *ViLBlock* en dos etapas del decodificador, replicando la posición relativa de los bloques ya empleados en el codificador de la arquitectura descrita en la Subsection 4.4.2. Asimismo, se añade un bloque adicional del tipo *BasicBlockD* en la misma etapa del decodificador donde dicho bloque fue incorporado en el codificador.

La Figura 4.8 muestra un esquema simplificado de la arquitectura, con las cinco primeras etapas del codificador y las cinco últimas del decodificador. La última etapa del codificador no se conecta con el decodificador, lo cual queda reflejado en el esquema mediante la ausencia de flecha de conexión en dicha etapa. tampoco se ilustra, aunque incluye una operación de *upsampling*, seguida de un *BasicResBlock* y un *BasicBlockD*. Al igual que en las figuras anteriores, se mantienen las mismas convenciones gráficas. Destacan los bloques en azul claro, correspondientes a los *ViLBlock*, cuya presencia en ambas fases es distintiva de esta variante. Esta propuesta guarda similitud conceptual con la arquitectura presentada por Heidari et al. [16], que plantea una integración extensa de mecanismos de atención y memoria a lo largo de toda la U-Net. No obstante, en este trabajo se opta por una integración más limitada de los bloques *ViLBlock* en el decodificador, debido a restricciones de memoria GPU. La extensión de capas xLSTM a todas las etapas del decodificador, como en el estudio original, resultó inviable por exceder los límites de VRAM. Por ello, se reutilizan únicamente las etapas ya enriquecidas con atención en el codificador, buscando un equilibrio entre rendimiento y capacidad representativa.

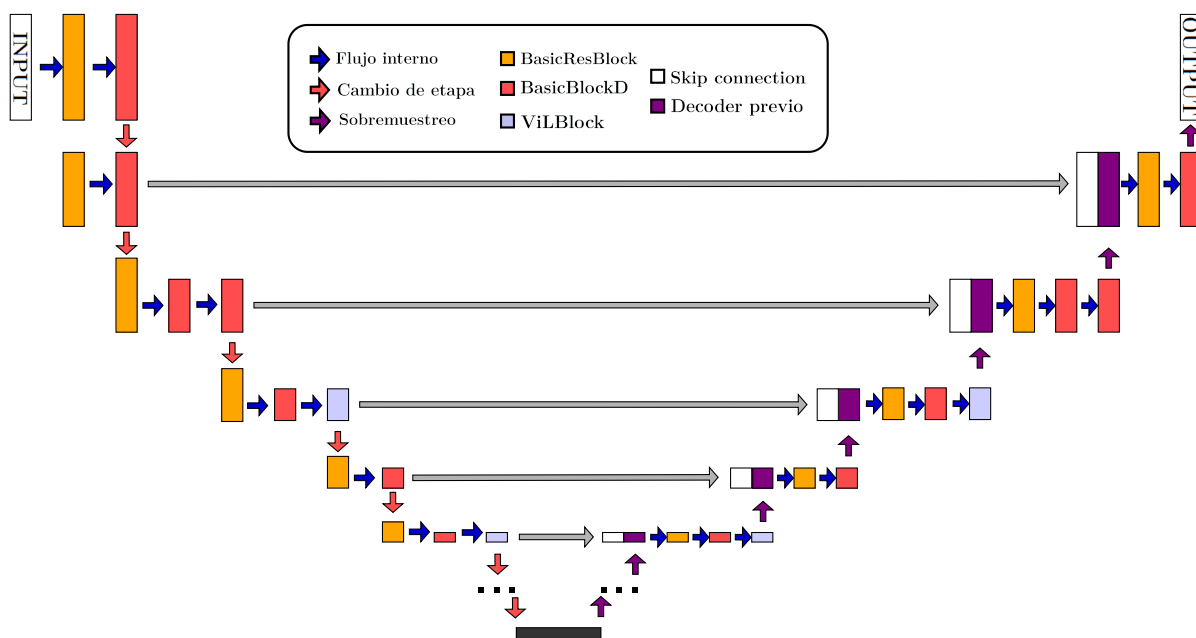


Figura 4.8: Arquitectura de *xLSTM-UNet_EncDec*.

Capítulo 5

Desarrollo

5.1. Conjuntos de datos

En las secciones previas se presentaron los conjuntos de datos empleados a lo largo del desarrollo del trabajo. En total, se emplearon cuatro conjuntos de datos, los cuales son los mismos que se emplean en el artículo original de referencia [10].

5.1.1. Resonancias magnéticas del abdomen (Abdomen MRI)

Este conjunto de datos se emplea para la tarea de segmentación de órganos abdominales y procede del desafío MICCAI 2022 AMOS Challenge [28]. Consta de 60 casos de resonancia magnética (MRI) utilizados para el entrenamiento y 50 destinados al test, todos ellos anotados manualmente por radiólogos expertos.

El conjunto incluye 13 clases correspondientes a distintas estructuras anatómicas (hígado, riñón derecho, bazo, páncreas, aorta, vena cava inferior, glándula suprarrenal derecha, glándula suprarrenal izquierda, vesícula biliar, esófago, estómago, duodeno y riñón izquierdo), las cuales están presentes simultáneamente en todas las imágenes del conjunto de datos. Las imágenes se encuentran en formato *.nii.gz*, un estándar utilizado para almacenar datos volumétricos médicos, y corresponden a estudios de resonancia magnética (MRI). Un ejemplo de la segmentación anatómica de estos órganos se muestra en la Figura 5.1. Dado que los datos presentan una estructura tridimensional, se realizaron experimentos tanto en 2D como en 3D, aprovechando la naturaleza volumétrica de las imágenes para explorar ambos enfoques de segmentación.

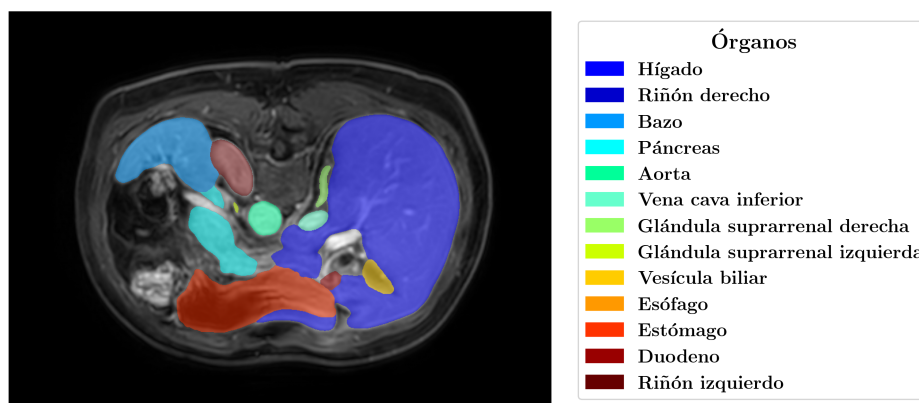


Figura 5.1: Ejemplo de segmentación de órganos abdominales en una imagen de resonancia magnética (MRI) del conjunto de datos AMOS.

5.1.2. Imágenes de instrumentos de endoscopia (Endoscopy Images)

Este conjunto pertenece al desafío MICCAI 2017 EndoVis Robotic Instrument Segmentation Challenge [1], centrado en la segmentación de instrumentos quirúrgicos en imágenes obtenidas mediante endoscopia asistida por el sistema quirúrgico da Vinci. Este sistema permite a los cirujanos operar usando brazos robóticos que controlan desde una consola. Les da una imagen ampliada y en alta definición del área quirúrgica, y les ayuda a hacer movimientos más precisos que con las manos directamente. Consta de 1800 imágenes para entrenamiento y 1200 para test. Cada imagen es una captura individual de un vídeo quirúrgico, almacenada en formato *.png*, representando una imagen RGB en la que cada canal contiene la intensidad de uno de los tres colores básicos. El conjunto incluye siete clases de instrumentos además del fondo: pinza bipolar, pinza Prograsp, portaagujas grande, sellador de vasos, retractor de presión, tijeras curvas monopolares y sonda de ultrasonido.

En este caso, a diferencia del conjunto de datos de Abdomen MRI, no todas las clases están presentes simultáneamente en cada imagen, ya que la aparición de los instrumentos depende del momento quirúrgico capturado. Esta variabilidad aumenta la dificultad de la tarea de segmentación semántica, ya que el modelo debe identificar no solo múltiples clases sino también su presencia intermitente. Un ejemplo de la segmentación de instrumentos en imágenes endoscópicas se presenta en la Figura 5.2. Dado que se trabaja con imágenes individuales en 2D y en formato de imagen estándar, los experimentos realizados sobre este conjunto se desarrollaron exclusivamente en dos dimensiones.

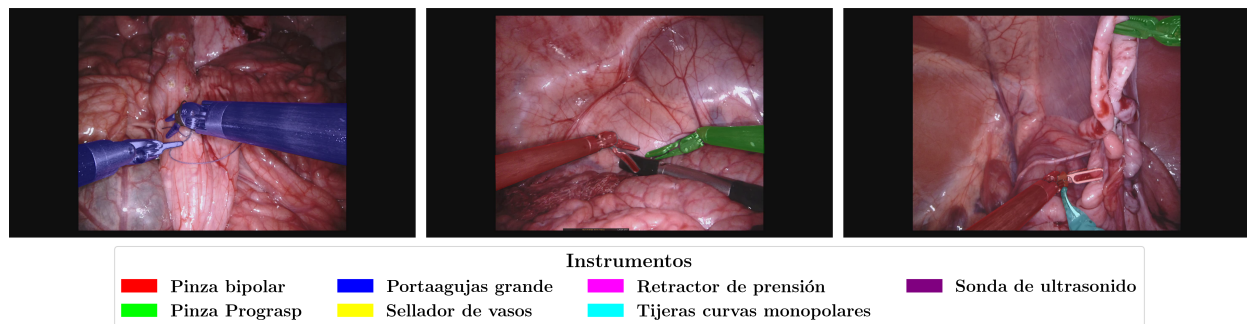


Figura 5.2: Ejemplo de segmentación de instrumentos quirúrgicos en imágenes endoscópicas del desafío MICCAI 2017 EndoVis Challenge.

5.1.3. Imágenes microscópicas (Microscopy Images)

Este conjunto de datos procede del desafío NeurIPS 2022 Cell Segmentation Challenge [27] y está destinado a la segmentación de células en imágenes de microscopía. Se utilizaron 800 imágenes para el entrenamiento y 200 imágenes para test. Cada imagen está en formato *.png* y corresponde a una imagen en color RGB. El conjunto define tres clases principales: fondo, interior de la célula y borde de la célula. Un ejemplo de estas segmentaciones puede verse en la Figura 5.3. Dado que las imágenes son bidimensionales, los experimentos se realizaron exclusivamente en 2D.

Una característica destacada de este conjunto de datos es su elevada diversidad, ya que incluye más de 1.000 imágenes etiquetadas procedentes de más de 50 experimentos biológicos distintos. Las imágenes abarcan varias modalidades de microscopía, como campo claro (brightfield), fluorescencia, contraste de fases (phase-contrast, PC) y contraste de interferencia diferencial (DIC). Asimismo, se observa una notable variabilidad en el origen celular (procedente tanto de tejidos como de cultivos celulares), en las técnicas de tinción empleadas (por ejemplo, tinciones Jenner-Giemsa para campo claro o anticuerpos específicos en fluorescencia), en los tipos de microscopios utilizados, así como en la morfología de las células, que pueden presentar formas redondeadas, alargadas o irregulares. Las anotaciones se realizaron y verificaron manualmente por biólogos expertos, asegurando un control de calidad riguroso para garantizar la precisión de las segmentaciones.

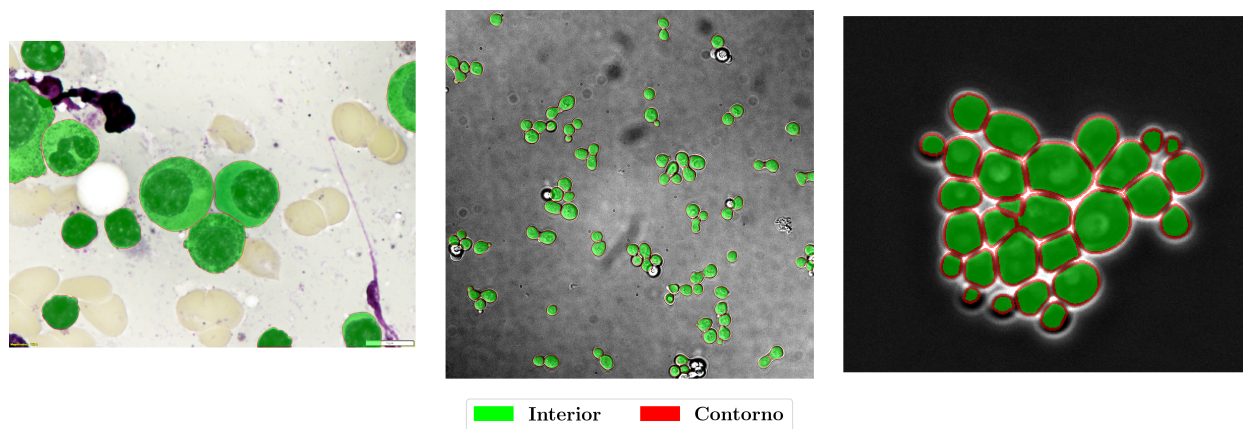


Figura 5.3: Ejemplo de segmentación celular en imágenes de microscopía del desafío NeurIPS 2022 Cell Segmentation Challenge.

5.1.4. Imágenes de tumores cerebrales (BraTS2023 - MEN)

El conjunto de datos BraTS2023-MEN [30] forma parte del desafío internacional BraTS2023 y está compuesto por volúmenes tridimensionales de resonancia magnética cerebral, específicamente diseñados para la segmentación automática de *meningiomas*, que son tumores extraaxiales comúnmente benignos que se desarrollan en las meninges. Los tumores extraaxiales son aquellos que se encuentran fuera del parénquima cerebral, es decir, fuera del tejido del cerebro propiamente dicho. Se localizan generalmente en las cubiertas del cerebro (como las meninges), en los nervios craneales o en estructuras óseas del cráneo. Se consideran comúnmente benignos, ya que en la mayoría de los casos no son cancerosos ni invaden tejidos cercanos de forma agresiva. Además, el hecho de que se desarrollen en las meninges implica que estos tumores afectan a las membranas que recubren y protegen el cerebro y la médula espinal.

Cada volumen incluye cuatro modalidades diferentes de resonancia magnética: T1, T1 con gadolinio (T1Gd), T2 y FLAIR. La modalidad T1 proporciona una vista anatómica general del cerebro, con buen contraste entre sustancia gris y blanca, y el líquido cefalorraquídeo aparece oscuro. T1Gd, obtenida tras la administración de un agente de contraste (gadolinio),

resalta las regiones del tumor que captan contraste, lo que indica una mayor vascularización o actividad tumoral. La modalidad T2 es útil para observar la presencia de líquidos o edemas, ya que estas áreas aparecen brillantes, al igual que el líquido cefalorraquídeo. Por su parte, la modalidad FLAIR (una variante de T2) permite suprimir la señal del líquido cefalorraquídeo, facilitando la identificación de lesiones cercanas al tumor. En este conjunto, el objetivo es segmentar tres regiones clave asociadas al meningioma: la región del tumor que capta contraste (*Enhancing Tumor*), el núcleo tumoral que no presenta captación de contraste (*Non-Enhancing Tumor Core*) y el área circundante con hiperintensidad en FLAIR (*Surrounding FLAIR Hyperintensity*), que puede indicar edema o infiltración tumoral. Un ejemplo de estas segmentaciones se muestra en la Figura 5.4.

Las imágenes están almacenadas en formato *.nii.gz*, que preserva la estructura tridimensional de los datos. Para el proceso experimental, se utilizaron 800 volúmenes para entrenamiento y 200 para evaluación. Los datos se organizan como volúmenes 4D, donde las tres primeras dimensiones corresponden al espacio (alto, ancho, profundidad) y la cuarta a los canales de las modalidades.

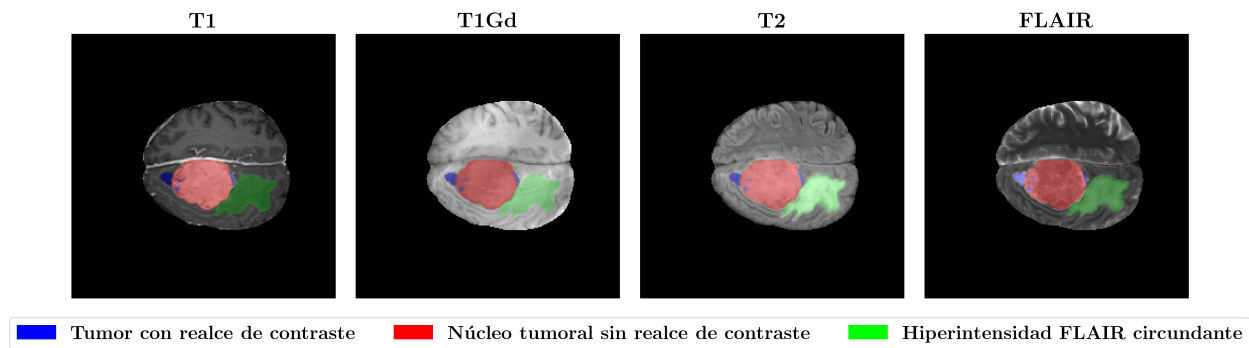


Figura 5.4: Ejemplo de segmentación de un tumor cerebral en las distintas modalidades de resonancia magnética del desafío BraTS2023-MEN.

Cabe destacar que los tres primeros conjuntos de datos (*Abdomen MRI*, *Endoscopy Images* y *Microscopy Images*) están disponibles públicamente en el repositorio de GitHub asociado al artículo original. Por otro lado, el conjunto de datos *BraTS2023-MEN* no se encuentra disponible de forma abierta y se solicitó oficialmente a través de la página del desafío.

En definitiva, aunque todos los conjuntos utilizados se orientan a la segmentación semántica, presentan particularidades que influyen en el enfoque experimental. En *Abdomen MRI*, todas las clases están presentes en cada imagen, lo que facilita una segmentación multicategoría coherente. En cambio, en *Endoscopy Images*, la variabilidad en la presencia de clases entre imágenes incrementa la complejidad. Además, las imágenes abarcan desde capturas 2D (endoscópicas o de microscopía) hasta volúmenes 3D multimodales, como en BraTS2023, que incluye cuatro modalidades distintas para una mejor caracterización tumoral. Esta diversidad en modalidades, formatos y estructuras evidencia la complejidad inherente a la segmentación en imágenes médicas.

5.2. Técnicas de preprocesamiento

El preprocesamiento de las imágenes se realizó exclusivamente mediante la función proporcionada por la librería nnU-Net v2 [10]. Esta herramienta analiza las propiedades y adapta automáticamente todo el pipeline de segmentación según las propiedades del conjunto de datos, tal como describen sus autores.

El proceso se divide en tres etapas principales. La primera es la creación de una huella digital (*fingerprint*) del conjunto de datos. En esta etapa se genera un archivo JSON que resume las características más importantes, como el tamaño de las imágenes, el espacio entre vóxeles en cada dimensión (es decir, la distancia real entre puntos en cada dirección del volumen) y los valores de intensidad de cada canal.

Seguidamente, comienza la fase de planificación del experimento. Utilizando la información del archivo JSON, el sistema decide a qué tamaño deben ajustarse los vóxeles para que todos los volúmenes tengan una escala similar. Para eso, suele usar el valor intermedio (la mediana) del espacio entre vóxeles en cada dirección. Además, detecta si las imágenes son anisotrópicas, es decir, si los vóxeles están mucho más separados en una dirección que en las otras, lo que puede afectar cómo se procesan. En nnU-Net v2, un conjunto se considera anisotrópico si la separación entre vóxeles en un eje supera en más de tres veces la de los otros dos. Según esto, se elige una estrategia adecuada, como usar modelos 2D para volúmenes muy anisotrópicos.

La fase final de preprocesamiento en nnU-Net aplica recorte del fondo, normalización de intensidades y remuestreo espacial para unificar las características de entrada y facilitar el aprendizaje del modelo. En primer lugar, se aplica un recorte a la región de interés, eliminando áreas vacías de fondo que no aportan información relevante. Esta operación reduce el tamaño de los volúmenes y hace que el modelo se enfoque en el tejido significativo. Las máscaras de segmentación se recortan de forma equivalente, marcando con una etiqueta de ignorado (-1) las regiones descartadas.

A continuación, se normalizan las intensidades por canal según el tipo de modalidad de imagen. En imágenes de tomografía computarizada (CT), se aplica una normalización basada en percentiles y ajuste global, mientras que en resonancia magnética (MRI) se emplea una normalización z-score por caso, considerando únicamente los píxeles distintos de cero. Este paso se realiza antes del remuestreo para conservar la distribución original de intensidades.

El último paso es modificar el tamaño de las imágenes para que todas tengan la misma escala, según lo definido en la planificación. Este paso ajusta la resolución a una escala uniforme, utilizando interpolación y vecino más cercano para las máscaras de segmentación. En caso de que el volumen resultante sea más pequeño que el tamaño de parche requerido, se rellena con ceros, asegurando que todos los datos de entrada tengan dimensiones compatibles con la arquitectura del modelo.

Finalmente, los volúmenes preprocesados se dividen en parches y se almacenan en disco como archivos organizados por configuración (2D o 3D). Este flujo garantiza que todos los datos de entrada estén espacial y estadísticamente normalizados, maximizando así la eficacia del entrenamiento.

5.3. Descripción de la metodología

El desarrollo de este proyecto se estructuró conforme a una metodología iterativa-incremental [26], escogida debido a su eficacia para gestionar la complejidad y la incertidumbre de los procesos de investigación y desarrollo de modelos avanzados de segmentación de imágenes.

Esta metodología permite dividir el trabajo en pequeños incrementos funcionales, con ciclos de retroalimentación y validaciones continuas, lo que facilita la detección temprana de errores, la adaptación a cambios y la optimización progresiva del modelo. La elección de este enfoque se justifica también porque el desarrollo iterativo ha demostrado ser más eficaz que los modelos secuenciales tradicionales en proyectos donde los requisitos pueden evolucionar o no estar completamente definidos desde el inicio, como es el caso.

El proceso de desarrollo se organizó en fases claramente diferenciadas, que se repiten en cada iteración:

- **Análisis de requisitos y planificación:** Identificación de los objetivos específicos de cada iteración, así como selección y preparación de los datos de entrada.
- **Diseño e implementación:** Definición o ajuste de la arquitectura del modelo, en este caso, una combinación de U-Net con la variante xLSTM.
- **Pruebas y evaluación:** Entrenamiento del modelo, análisis de métricas de rendimiento como el Coeficiente de Similitud de Dice, y evaluación comparativa con versiones anteriores.
- **Retroalimentación y mejora:** Incorporación de los aprendizajes obtenidos para redefinir objetivos en la siguiente iteración.

Cada incremento añade nuevas capacidades o mejoras sobre el sistema anterior, en línea con el principio descrito por el modelo iterativo-incremental: desarrollar un sistema completo en pequeños pasos funcionales, en lugar de esperar a completar todo el proyecto para validar su funcionamiento.

5.4. Hiperparámetros del entrenamiento

El entrenamiento de los modelos desarrollados en este trabajo se configuró mediante un conjunto de hiperparámetros cuidadosamente ajustados en función de la arquitectura, el tipo de datos y las limitaciones computacionales. A continuación, se describen los principales parámetros involucrados en el proceso, así como las técnicas de optimización y la función de pérdida utilizada.

5.4.1. Iteraciones por época y por validación

En cada época, el modelo se entrena mediante un número determinado de iteraciones de entrenamiento y se evalúa un número de iteraciones sobre el conjunto de validación. La métrica clave aquí es el número de lotes que se procesan por época, denominado iteraciones por época (*Itr/ep.*). Este valor, junto con el tamaño del lote, determina cuántas muestras se utilizan realmente en cada ciclo de entrenamiento. Por ejemplo, con un tamaño de lote (*batch*

size) de 30 y 2 iteraciones por época, se procesan 60 imágenes por ciclo de entrenamiento. Si se desea que el modelo procese la totalidad del conjunto de entrenamiento en cada época, el número de iteraciones por época debe calcularse utilizando la siguiente fórmula:

$$\text{Itr/ep.} = \left\lceil \frac{N_{\text{entrenamiento}}}{\text{batch size}} \right\rceil \quad (5.1)$$

donde $N_{\text{entrenamiento}}$ es el número total de muestras en el conjunto de entrenamiento, y $\lceil \cdot \rceil$ denota el redondeo hacia arriba.

De manera similar, las iteraciones de validación por época (ItrV/ep.) definen cuántos lotes se utilizan para evaluar el modelo después de cada época.

Durante las fases iniciales del desarrollo, se experimentó con diferentes configuraciones del número de iteraciones por época, recorriendo la mitad, la totalidad o incluso el doble del conjunto de entrenamiento, con el objetivo de analizar su impacto en el rendimiento del modelo durante la primera iteración.

5.4.2. Número de épocas (*epochs*)

El número de épocas determina cuántas veces se recorre el conjunto de entrenamiento completo (o su equivalente, en función del número de iteraciones por época). Este parámetro influye directamente en el grado de ajuste del modelo, siendo uno de los principales factores a monitorizar para evitar el sobreajuste. En experimentos incrementales, se almacenaron puntos intermedios del modelo (*checkpoints*) cada 20 épocas con el fin de evaluar su evolución progresiva.

Es importante destacar que el número de épocas a utilizar dependerá también de las condiciones del entorno de ejecución, ya que influye directamente en el tiempo total requerido para completar un experimento. En entornos con recursos computacionales limitados o tiempos de entrenamiento prolongados por época (por ejemplo, debido a imágenes de alta resolución o modelos complejos), puede ser necesario reducir el número total de épocas. Por el contrario, cuando el tiempo de ejecución por época es bajo, se puede optar por realizar un mayor número de épocas sin que ello comprometa la viabilidad temporal del experimento.

5.4.3. Tamaño del lote (*batch size*)

El tamaño del lote define cuántas muestras se procesan simultáneamente en cada iteración. Valores altos permiten aprovechar mejor la paralelización en GPU, pero requieren mayor memoria. En este trabajo se emplearon tamaños de lote entre 1 y 30, ajustados a las características del conjunto de datos y a las restricciones de memoria de la plataforma utilizada (Google Colab o entorno físico).

Concretamente, los tamaños de lote se adaptaron en función de la disponibilidad de memoria GPU en cada entorno de ejecución. En Google Colab, con un límite aproximado de 15 GB de memoria GPU, fue necesario utilizar tamaños de lote más pequeños para evitar errores de desbordamiento de memoria. Por otro lado, en el entorno físico, donde se dispone de una GPU con 24 GB de memoria, fue posible utilizar lotes más grandes. Estos detalles se

describen con mayor precisión en la Sección 1.3, dedicada a los recursos utilizados durante el proyecto.

5.4.4. Tasa de aprendizaje (*learning rate*)

La tasa de aprendizaje (η) regula la magnitud de las actualizaciones de los pesos del modelo en cada paso de optimización. Valores elevados aceleran el entrenamiento, pero pueden generar inestabilidad, mientras que valores bajos proporcionan convergencia más estable, pero lenta. Se emplearon tasas entre 0.0005 y 0.005 según el experimento y la arquitectura utilizada.

En este trabajo, no se realizaron experimentaciones sistemáticas exhaustivas sobre la tasa de aprendizaje. En su lugar, se partió de los valores recomendados en Chen et al. [10]. Si durante las primeras épocas se observaba inestabilidad en el entrenamiento, como pérdidas que no disminuían o aumentaban bruscamente, la tasa de aprendizaje se ajustaba reduciéndola un orden de magnitud. Este enfoque empírico permitió mejorar la estabilidad sin necesidad de una búsqueda extensa de hiperparámetros.

La gestión de la tasa de aprendizaje se automatizó a través de un planificador de tipo decaimiento polinómico (`PolyLRScheduler`), que ajusta el valor de η dinámicamente en función de la época actual. La fórmula empleada tiene la forma:

$$\eta_t = \eta_0 \cdot \left(1 - \frac{t}{T}\right)^p \quad (5.2)$$

donde η_0 es la tasa inicial, t es la época actual, T el número total de épocas, y p el exponente de decaimiento (por defecto cercano a 0.9). Este enfoque permite iniciar el entrenamiento con un aprendizaje más agresivo y reducirlo progresivamente.

5.4.5. Decaimiento de pesos y optimizador AdamW

Como técnica de regularización, se utilizó decaimiento de pesos (*weight decay*), una penalización de tipo L2 sobre los pesos del modelo que ayuda a evitar su crecimiento descontrolado. Esta técnica se implementa sumando el cuadrado de los pesos a la función de pérdida total, lo que favorece modelos más generalizables. El esquema de actualización de pesos es el siguiente:

$$w_t = w_{t-1} - \eta \cdot (\nabla L(w_{t-1}) + \lambda w_{t-1}) \quad (5.3)$$

donde:

- w_t es el peso actualizado,
- η es la tasa de aprendizaje,
- $\nabla L(w_{t-1})$ es el gradiente de la pérdida,
- λ es el coeficiente de decaimiento de pesos.

Este esquema se aplicó mediante el optimizador AdamW, una variante del optimizador Adam que desacopla explícitamente el weight decay del término de regularización del gradiente, ofreciendo mejores propiedades de convergencia en problemas de segmentación.

5.4.6. Función de pérdida compuesta: Entropía Cruzada y Dice

Para entrenar los modelos, se utilizó una función de pérdida compuesta que combina la pérdida de entropía cruzada (*Cross Entropy Loss*) y la pérdida de Dice suavizada (*Soft Dice Loss*), ponderadas por coeficientes ajustables (w_{CE} y w_{Dice}):

$$\mathcal{L} = w_{CE} \cdot \mathcal{L}_{CE} + w_{Dice} \cdot \mathcal{L}_{Dice} \quad (5.4)$$

La pérdida de entropía cruzada compara la probabilidad predicha $\hat{y}_i$ con la etiqueta real y_i . Se define como:

$$\mathcal{L}_{CE} = - \sum_i y_i \log(\hat{y}_i), \quad (5.5)$$

La pérdida de Dice suavizada evalúa directamente la superposición entre la predicción y la máscara real (*ground truth*). Está dada por:

$$\mathcal{L}_{Dice} = 1 - \frac{2 \sum p_i g_i}{\sum p_i + \sum g_i} \quad (5.6)$$

donde p_i y g_i son los valores predichos y reales, respectivamente.

Esta combinación permite equilibrar la penalización sobre errores de clasificación globales y errores en la superposición espacial, mejorando el rendimiento del modelo en regiones pequeñas o desbalanceadas.

5.4.7. Promedio Exponencialmente Móvil (EMA)

Durante el entrenamiento, se utilizó un mecanismo de suavizado basado en el cálculo del Promedio Exponencialmente Móvil (EMA), aplicado a la métrica Pseudo Dice Score. Esta métrica evalúa únicamente la superposición en las clases de interés (excluyendo el fondo), lo que proporciona una medida más representativa del rendimiento real del modelo en tareas de segmentación médica. El EMA permite mitigar la variabilidad de las métricas de evaluación entre épocas consecutivas, ofreciendo una estimación más estable del progreso del modelo.

$$EMA_t = \alpha \cdot EMA_{t-1} + (1 - \alpha) \cdot Dice_t \quad (5.7)$$

donde $\alpha = 0,9$ es el factor de suavizado, EMA_t es el valor suavizado en la época actual, EMA_{t-1} el de la época anterior, y $Dice_t$ el valor del Pseudo Dice Score calculado en la época t , en función de la coincidencia entre las predicciones del modelo y las máscaras reales de segmentación en dicha iteración del entrenamiento.

5.5. Fases de desarrollo

El proceso de desarrollo del proyecto se estructuró en tres iteraciones principales, suficientes para lograr una mejora progresiva de la arquitectura híbrida xLSTM-U-Net. Estas fases permitieron validar, ajustar y optimizar el modelo en función de los aprendizajes obtenidos en los ciclos anteriores. A través de este enfoque iterativo, se logró una evolución continua del rendimiento mediante experimentación controlada, análisis comparativo y realimentación sistemática.

5.5.1. Primera Iteración: Experimentos preliminares en Google Colaboratory

Durante la primera iteración del estudio, se llevaron a cabo una serie de experimentos preliminares utilizando la plataforma Google Colaboratory. El objetivo principal de esta fase fue comprobar que la integración entre la arquitectura U-Net y xLSTM funcionaba correctamente en un entorno con recursos limitados. En esta etapa inicial se abordó la fase de modelado y definición de requisitos. Para ello, se seleccionaron tres conjuntos de datos con características diversas: AbdomenMRI (5.1.1), Endovis (5.1.2) y NeurIPS (5.1.3). De cada uno de ellos se extrajeron subconjuntos específicos de imágenes para entrenamiento y prueba, adaptados a las capacidades computacionales del entorno.

Posteriormente, en la fase de diseño e implementación, se comenzó con una proceso de familiarización con el código base proporcionado por los autores del artículo original que introduce la arquitectura xLSTM integrada en U-Net [10]. Este código sirvió como punto de partida para el desarrollo. Se analizó el preprocesamiento de datos, la arquitectura propuesta y su implementación, con especial énfasis en el uso de funciones especializadas para el entrenamiento y la evaluación proporcionadas por la librería. A partir de dicho análisis, se construyeron y entrenaron dos variantes clave del modelo: *xLSTM-UNet_Bot* y *xLSTM-UNet_Enc* (descritas en las Secciones 4.4.1 y 4.4.2).

En la etapa de pruebas y despliegue se llevaron a cabo múltiples experimentos de entrenamiento y validación. Se emplearon diferentes configuraciones para cada combinación de conjunto de datos y arquitectura, con el fin de evaluar el comportamiento de los modelos en condiciones variadas. Inicialmente, se utilizaron los valores de tasa de aprendizaje y tamaño de lote recomendados por los autores del artículo de referencia. Posteriormente, se exploraron configuraciones alternativas más adecuadas a las limitaciones computacionales de Colab y a las particularidades de cada conjunto de datos. Las variables analizadas incluyeron el número de iteraciones de entrenamiento y validación por época, el número total de épocas, el tamaño del lote, la tasa de aprendizaje y el tiempo total de entrenamiento estimado en minutos.

Durante el proceso de entrenamiento, se dividió el conjunto de datos en un 80 % para el entrenamiento propiamente dicho y un 20 % para la validación, permitiendo así un seguimiento del rendimiento del modelo en cada época. La Tabla 5.1 resume las configuraciones más representativas empleadas en esta primera iteración experimental para los conjuntos de datos AbdomenMRI (en versiones 2D y 3D), Endovis y NeurIPS.

Dataset	Arquitectura	Itr/ep.	ItrV/ep.	Ep.	Bs	Lr	Tpo.
1. AbdomenMRI (2D)	xLSTMBot	2	5	10	30	0.005	5
2. AbdomenMRI (2D)	xLSTMEnc	150	60	50	2	0.0025	94
3. AbdomenMRI (2D)	xLSTMBot	50	60	50	30	0.005	100
4. AbdomenMRI (2D)	xLSTMBot	50	60	75	30	0.005	140
5. AbdomenMRI (2D)	xLSTMEnc	150	60	75	2	0.0025	143
6. AbdomenMRI (3D)	xLSTMBot	2	5	10	2	0.005	1
7. AbdomenMRI (3D)	xLSTMBot	10	25	50	2	0.005	30
8. AbdomenMRI (3D)	xLSTMBot	24	6	25	2	0.005	20
9. AbdomenMRI (3D)	xLSTMBot	24	6	50	2	0.005	39
10. AbdomenMRI (3D)	xLSTMBot	24	6	75	2	0.005	57
11. Endovis (2D)	xLSTMBot	500	400	20	2	0.001	76
12. Endovis (2D)	xLSTMBot	500	400	50	2	0.001	199
13. Endovis (2D)	xLSTMBot	720	369	20	2	0.001	107
14. NeurIPS (2D)	xLSTMBot	67	16	25	12	0.001	59
15. NeurIPS (2D)	xLSTMEnc	160	40	25	4	0.0005	75
16. NeurIPS (2D)	xLSTMBot	67	16	50	12	0.001	111
17. NeurIPS (2D)	xLSTMEnc	160	40	50	4	0.0005	147

Cuadro 5.1: Configuraciones de entrenamiento durante la primera iteración experimental.

Finalmente, en la etapa de documentación y presentación, se recopilieron los resultados obtenidos y se identificaron aquellas configuraciones e hiperparámetros que mostraron un mejor rendimiento. Esta información fue esencial para definir el enfoque y las mejoras a implementar en la siguiente iteración del proyecto. La Sección 6.2 presenta los resultados detallados de los experimentos.

5.5.2. Segunda Iteración: Desarrollo incremental de los experimentos

La segunda iteración se centró en el estudio del comportamiento evolutivo de los modelos mediante un enfoque incremental. Para ello, se guardaron y evaluaron versiones del modelo cada 20 épocas, lo que permitió analizar cómo evolucionaban las métricas de rendimiento en el conjunto de prueba a lo largo del entrenamiento.

Se seleccionaron tres experimentos representativos: segmentación de órganos abdominales en 2D, segmentación en 3D del mismo dominio y segmentación de instrumentos quirúrgicos en imágenes de endoscopia en 2D. Adicionalmente, se obtuvo acceso al conjunto de datos Brain Tumor Segmentation (BraTS2023 5.1.4), utilizado para pruebas preliminares en tareas de segmentación cerebral en 3D. En el caso del conjunto BraTS2023, se observó que cada época requería aproximadamente una hora de procesamiento debido a la complejidad del volumen tridimensional y la cantidad de canales de entrada (cuatro por imagen). Esta duración hacía inviable completar el experimento en la infraestructura disponible en Google Colab, tal como se detalla en la Tabla 5.2.

Dataset	Arquitectura	Itr/ep.	ItrV/ep.	Ep.	Bs	Lr	Tpo.
18. BraTS2023 (3D)	xLSTMBot	320	80	25	2	0.005	1 Ep./1h.

Cuadro 5.2: Planificación del experimento para el dataset de segmentación de tumores cerebrales.

Con base en los experimentos más prometedores de la primera iteración, se diseñaron tres pruebas adicionales siguiendo un enfoque incremental, cuyos parámetros de entrenamiento se detallan en la Tabla 5.3. Para el conjunto AbdomenMRI en 2D, se utilizó una configuración compuesta por 50 iteraciones de entrenamiento y 60 de validación por época, un tamaño de lote de 30, una tasa de aprendizaje de 0.005 y un total de 180 épocas. En el caso de AbdomenMRI en 3D, se definieron 24 iteraciones de entrenamiento y 6 de validación por época, con un tamaño de lote de 2, tasa de aprendizaje de 0.005 y 160 épocas. Finalmente, para el conjunto Endovis en 2D, se empleó una configuración con 500 iteraciones de entrenamiento y 400 de validación por época, un tamaño de lote de 2, tasa de aprendizaje de 0.001 y una duración total de 80 épocas, ajustada para mantener los tiempos de ejecución dentro de límites razonables.

Dataset	Arquitectura	Itr/ep.	ItrV/ep.	Ep.	Bs	Lr	Tpo.(h)
3. AbdomenMRI (2D)	xLSTMBot	50	60	180	30	0.005	5
8. AbdomenMRI (3D)	xLSTMBot	24	6	160	2	0.005	2
12. Endovis (2D)	xLSTMBot	500	400	80	2	0.001	5

Cuadro 5.3: Experimentos evaluados de forma incremental.

Finalmente, se analizaron y recopilaron los resultados obtenidos en cada uno de estos entrenamientos para estudiar la evolución del rendimiento del modelo, así como los tiempos de ejecución asociados.

5.5.3. Tercera Iteración: Ejecución de experimentos de larga duración en equipo físico

En esta tercera iteración se realizó la transición a un equipo físico con mayor potencia de cómputo, debido a que las capacidades de Google Colab resultaban insuficientes para ejecutar experimentos de larga duración. Esta etapa representó un avance importante, ya que se desplegó el sistema en un equipo físico real, donde se realizaron no solo el entrenamiento, sino también la inferencia y la evaluación del modelo.

En primer lugar, se decidió aplicar validación cruzada con k particiones (*k-fold cross-validation*) debido a la variabilidad de los resultados causada por la inicialización aleatoria de los modelos. Esta decisión, correspondiente a la fase inicial de requisitos y modelado, buscó obtener una estimación del rendimiento más robusta y confiable al reducir la dependencia de una única partición de los datos. La validación cruzada *k-fold* consiste en dividir el conjunto de datos en k partes iguales (particiones), utilizando en cada iteración una de estas particiones como conjunto de validación y las $k-1$ restantes para el entrenamiento. El proceso se

repite k veces, y se promedian los resultados obtenidos. Dependiendo del tiempo de ejecución requerido por cada experimento, se optó por un valor de $k = 10$ en los casos con menor carga computacional, y $k = 5$ en aquellos experimentos que demandaban más recursos.

Por otro lado, se implementó la arquitectura de tipo *xLSTM-UNet-EncDec* descrita en la Sección 4.4.3, y se aplicó exclusivamente al conjunto AbdomenMRI 2D, ya que su complejidad computacional la hacía inviable para conjuntos 3D o de mayor tamaño.

A partir de esa decisión, y ya dentro de la fase de diseño e implementación, se tuvieron que ajustar los hiperparámetros según las características de cada experimento. En primer lugar, el tamaño de lote se pudo aumentar en algunos casos gracias a la disponibilidad de memoria VRAM suficiente en el equipo físico. Además, se tuvo que recalcular el número de iteraciones por época y de validación en función de la partición de los datos determinada por el valor de k en la validación cruzada. También se midieron los tiempos de ejecución por época y, con base en estos resultados, se decidió el número total de épocas a entrenar: a menor tiempo por época, mayor era la cantidad de épocas que se podía permitir. Finalmente, se conservaron los valores de la tasa de aprendizaje utilizados en iteraciones anteriores, ya que habían demostrado ser suficientemente robustos y eficaces durante el entrenamiento previo. Para cada una de las k particiones definidas, se entrenó y evaluó el modelo desde cero utilizando los $k-1$ subconjuntos para entrenamiento y el restante para validación.

1. Resonancias magnéticas del abdomen en 2D

Este conjunto de datos es el más pequeño de todos, lo que permite realizar experimentos con un mayor número de épocas sin comprometer los recursos computacionales. En este caso, se diseñaron seis experimentos: tres de ellos alcanzan las 500 épocas y, tras analizar las gráficas de la evolución del promedio exponencialmente móvil en el conjunto de validación, se tomó la decisión de ampliar a 700 épocas en los tres restantes. Además, se utilizó un esquema de validación cruzada con $k = 10$ (*10-fold cross-validation*), con el objetivo de obtener métricas más robustas y generalizables. La siguiente Tabla 5.4 representa la configuración de cada experimento, incluyendo los hiperparámetros, el tiempo promedio de entrenamiento por partición (*fold*) en minutos y el tiempo total de entrenamiento estimado en horas.

Arquitectura	Itr/ep.	ItrV/ep.	Ep.	Bs	Lr	T/fold (min)	T total (h)
xLSTMBot	2	1	500	30	0.005	18	3
xLSTMEnc	18	2	500	3	0.00125	45	7:30
xLSTMEncDec	27	3	500	2	0.00125	77	13
xLSTMBot	2	1	700	30	0.005	24	4
xLSTMEnc	18	2	700	3	0.00125	52	9
xLSTMEncDec	27	3	700	2	0.00125	102	17

Cuadro 5.4: Configuración y tiempos de los experimentos con AbdomenMRI 2D en la tercera iteración.

En la Figura 5.5 se presentan las curvas de validación correspondientes a los primeros

tres experimentos, entrenados durante 500 épocas. Cada línea representa la evolución del coeficiente DICE suavizado mediante promedio exponencial (EMA-DSC) para uno de los 10 folds del esquema de validación cruzada excluyendo la clase de fondo. Se observa que, si bien todas las arquitecturas muestran una mejora progresiva en la métrica a lo largo de las épocas, en algunos casos, particularmente en las arquitecturas xLSTM-UNet_Enc y xLSTM-UNet_EncDec, la curva aún no se estabilizó completamente al final de la época 500, lo que sugiere que el entrenamiento podría beneficiarse de un mayor número de épocas. Además, xLSTM-UNet_Enc evidencia un mejor desempeño promedio en validación, alcanzando valores más altos de EMA-DSC en comparación con las otras variantes, mientras que xLSTM-UNet_EncDec muestra una mayor consistencia entre particiones.

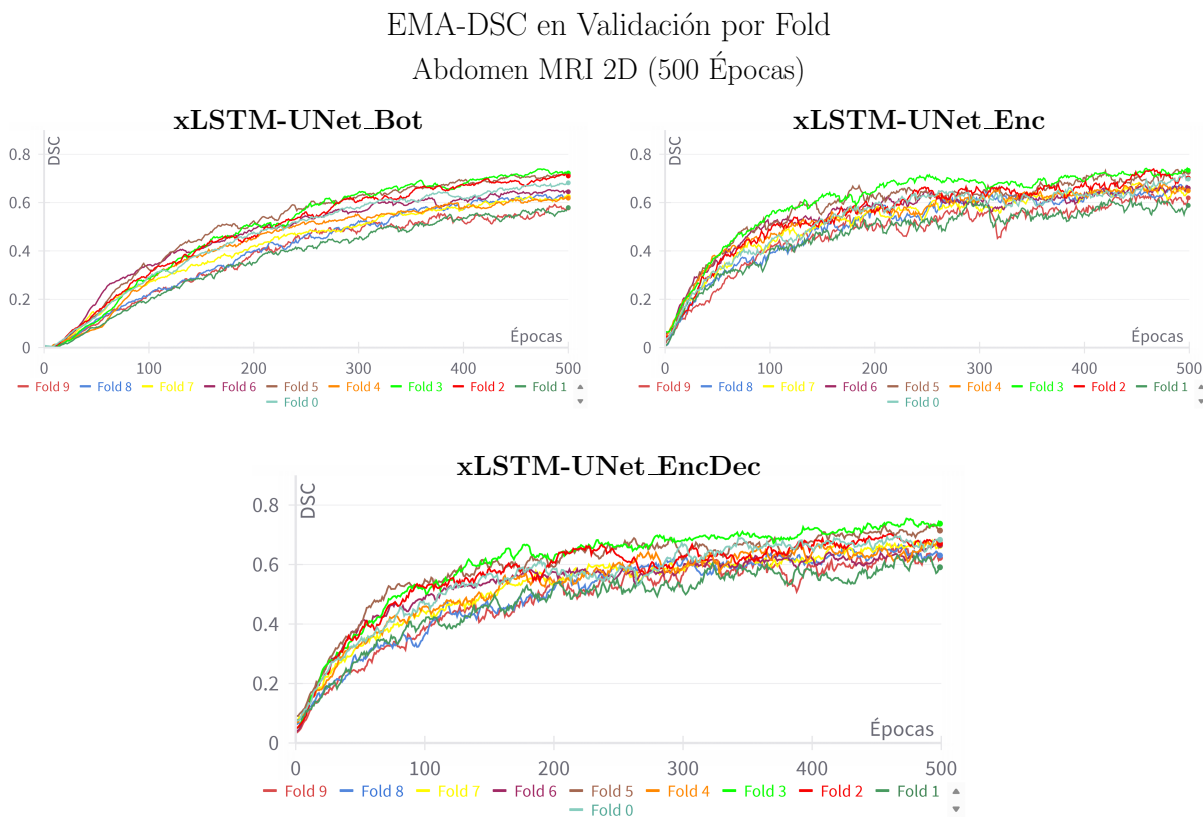


Figura 5.5: Curvas de validación (EMA-DSC) por *fold* en los experimentos con 500 épocas para el conjunto Abdomen MRI 2D.

En la Figura 5.6 se visualiza el comportamiento de las mismas arquitecturas extendidas hasta 700 épocas. El incremento en la duración del entrenamiento permitió una estabilización más clara de las curvas, especialmente en xLSTM-UNet_Enc, donde se consolida como la arquitectura con mejor rendimiento promedio sobre los subconjuntos de validación. Asimismo, se confirma que xLSTM-UNet_EncDec mantiene una dispersión reducida entre los distintos *folds*, reforzando su estabilidad durante la validación cruzada.

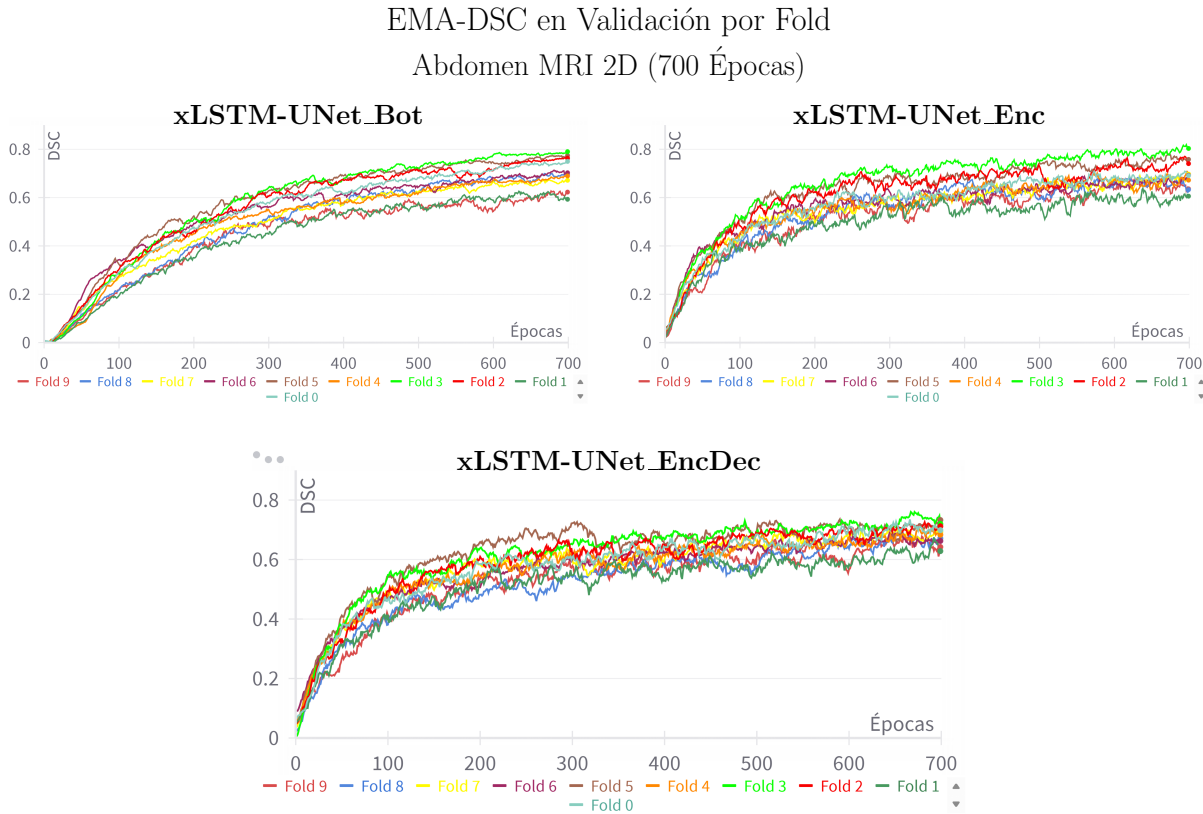


Figura 5.6: Curvas de validación (EMA-DSC) por *fold* en los experimentos con 700 épocas para el conjunto Abdomen MRI 2D.

2. Resonancias magnéticas del abdomen en 3D

Este conjunto de datos corresponde a la modalidad tridimensional original, ya que las resonancias magnéticas del abdomen son naturalmente tridimensionales. A diferencia del caso 2D, en estos experimentos se utilizó exclusivamente la arquitectura xLSTM-UNet_Bot. Esta decisión se tomó debido a limitaciones de memoria RAM disponibles, que impidieron entrenar variantes más complejas como xLSTM-UNet_Enc o xLSTM-UNet_EncDec en 3D. Además, dicha arquitectura fue la única utilizada en el estudio original, lo cual permite establecer una comparación directa y consistente con trabajos previos. Al igual que en el caso anterior, se empleó un esquema de validación cruzada con $k = 10$ (*10-fold cross-validation*) con el fin de garantizar una evaluación robusta. La configuración detallada del experimento, incluyendo los hiperparámetros empleados, el tiempo promedio de entrenamiento por *fold* y el tiempo total estimado, se resume en la Tabla 5.5.

Arquitectura	Itr/ep.	ItrV/ep.	Ep.	Bs	Lr	T/fold (min)	T total (h)
xLSTMBot	27	3	200	2	0.005	35	6

Cuadro 5.5: Configuración y tiempos de los experimentos con AbdomenMRI 3D en la tercera iteración.

En la Figura 5.7 se muestra la evolución del coeficiente suavizado a lo largo de las 200 épocas de entrenamiento para cada una de las 10 particiones del esquema de validación cruzada. A diferencia de los experimentos en 2D, se observa un entrenamiento notablemente más estable, con curvas más suaves y consistentes entre los distintos subconjuntos. Este comportamiento se atribuye en parte a la mayor riqueza de información presente en las imágenes tridimensionales, lo cual facilita que el modelo aprenda patrones estructurales más robustos. Asimismo, se alcanza un rendimiento superior en comparación con las variantes en 2D, reflejado en valores más altos de la métrica estudiada en la fase de validación.

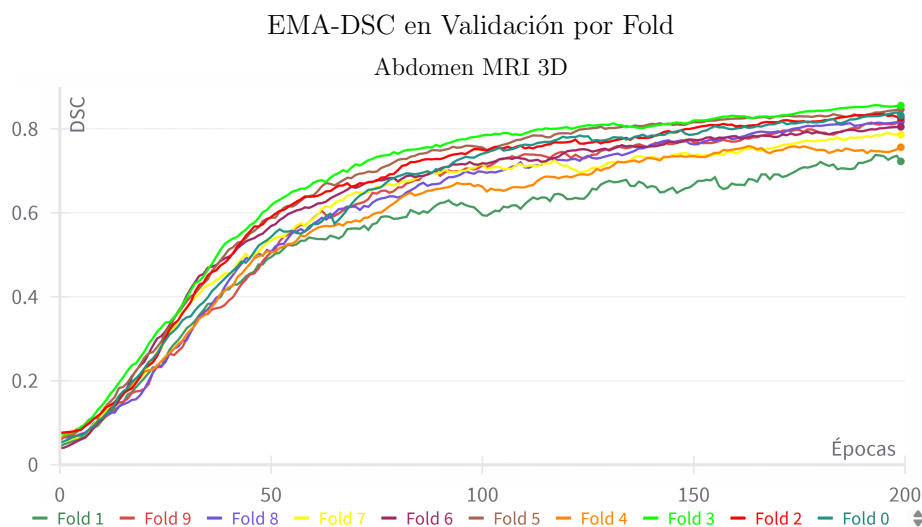


Figura 5.7: Curvas de validación (EMA-DSC) por *fold* en los experimentos con Abdomen MRI 3D.

3. Imágenes de instrumentos de endoscopia en 2D

En este caso se cambia de modalidad y dominio de los datos: pasamos de imágenes médicas de órganos abdominales a un nuevo entorno visual compuesto por instrumentos de endoscopia en 2D. Debido a las diferencias en la naturaleza de los datos y a las restricciones computacionales, se optó por realizar los experimentos únicamente con la arquitectura xLSTM-UNet_Bot. Se descartó la arquitectura xLSTM-UNet_Enc, ya que requería aproximadamente una hora por época de entrenamiento, lo que excedía tanto los recursos disponibles como los objetivos prácticos del proyecto.

Además, la duración del entrenamiento por partición alcanzaba las 4 horas incluso con la arquitectura más ligera, lo que motivó una reducción en el esquema de validación cruzada a $k = 5$ (*5-fold cross-validation*) para mantener los tiempos dentro de los márgenes razonables. La configuración detallada del experimento, junto con los tiempos aproximados de entrenamiento por subconjunto y el total estimado, se recoge en la Tabla 5.6.

Arquitectura	Itr/ep.	ItrV/ep.	Ep.	Bs	Lr	T/fold (h)	T total (h)
xLSTMBot	720	180	100	2	0.001	4	20

Cuadro 5.6: Configuración y tiempos de los experimentos con Endovis 2D en la tercera iteración.

En la Figura 5.8 se presenta la evolución del coeficiente suavizado (EMA-DSC) durante las 100 épocas de entrenamiento en cada una de las 5 particiones del esquema de validación cruzada aplicado al conjunto Endovis 2D. A pesar de tratarse de un dominio visual distinto al de las resonancias abdominales, se observa un comportamiento de entrenamiento progresivo y estable, con mejoras consistentes en la métrica a lo largo del tiempo.

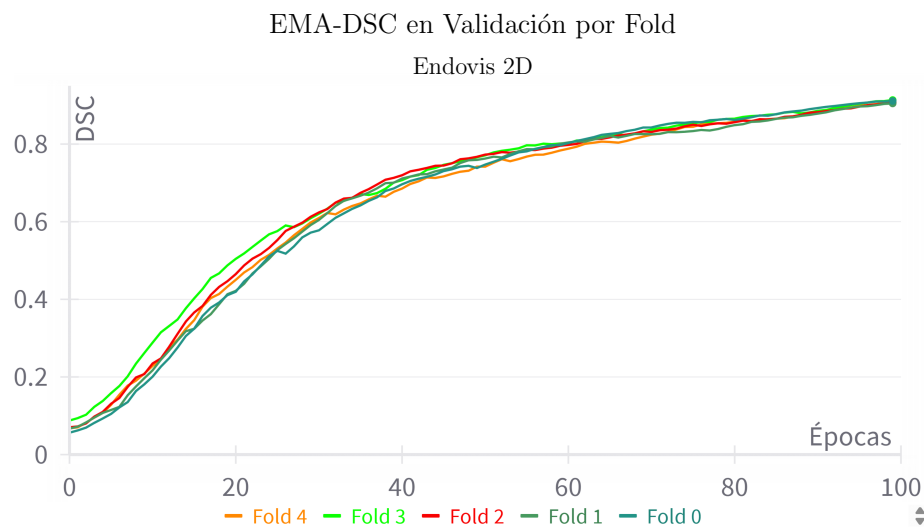


Figura 5.8: Curvas de validación (EMA-DSC) por *fold* en los experimentos con el conjunto Endovis 2D.

4. Imágenes microscópicas en 2D

Este conjunto de datos presenta una peculiaridad en la segmentación, ya que contiene dos clases diferenciadas: el interior y el borde celular. Esta última clase, al representar una fracción mucho menor de la superficie total, introduce un desafío adicional para el modelo, especialmente en términos de balance de clases y precisión. Para este experimento se utilizaron las arquitecturas xLSTM-UNet_Bot y xLSTM-UNet_Enc. No se pudo ejecutar xLSTM-UNet_EncDec debido a la gran cantidad de imágenes de entrenamiento y su impacto en los tiempos de procesamiento y uso de memoria. Sin embargo, al tratarse de imágenes en 2D en formato PNG, el tiempo de ejecución resultó considerablemente menor que en otros escenarios, lo cual permitió mantener un esquema de validación cruzada con $k = 10$ (*10-fold cross-validation*) sin comprometer la viabilidad del experimento. La Tabla 5.7 resume los hiperparámetros, tiempos promedio por subconjunto y duración total estimada de cada configuración.

Arquitectura	Itr/ep.	ItrV/ep.	Ep.	Bs	Lr	T/fold (min)	T total (h)
xLSTMBot	60	7	100	12	0.001	57	10
xLSTMEnc	120	14	100	6	0.0015	66	11

Cuadro 5.7: Configuración y tiempos de los experimentos con NeurIPS 2D en la tercera iteración.

En la Figura 5.9 se ilustra la evolución del coeficiente analizado a lo largo de las 100 épocas de entrenamiento para cada uno de los 10 subconjuntos del conjunto de entrenamiento NeurIPS 2D. Se comparan las arquitecturas xLSTM-UNet_Bot y xLSTM-UNet_Enc, ambas mostrando un comportamiento de entrenamiento estable y progresivo. En ambos casos, las curvas alcanzan valores elevados hacia el final del entrenamiento, con poca variabilidad entre particiones, lo que indica una buena generalización. La arquitectura xLSTM-UNet_Enc consigue un rendimiento ligeramente superior, especialmente en las primeras etapas del entrenamiento, probablemente gracias a su mayor capacidad para capturar patrones espaciales complejos, como los bordes celulares, que representan una proporción menor del área segmentada y constituyen un reto adicional para el modelo.

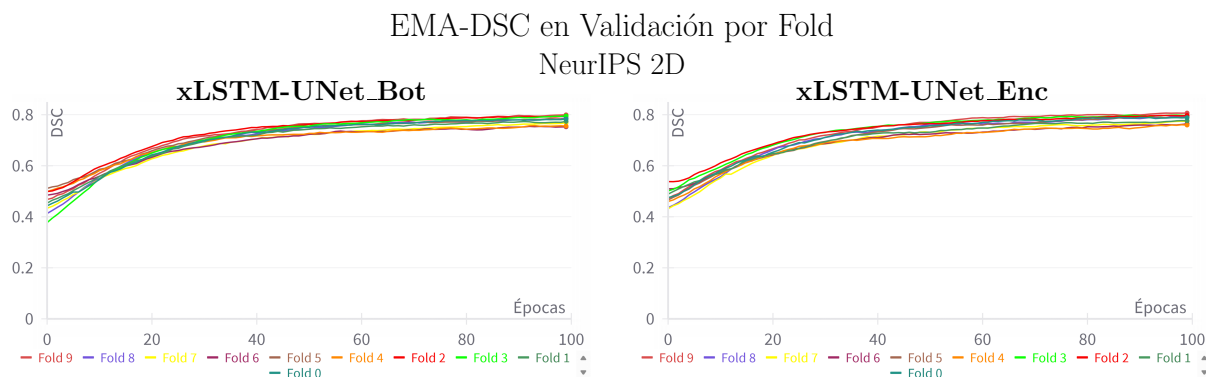


Figura 5.9: Curvas de validación (EMA-DSC) por *fold* en los experimentos con el conjunto NeurIPS 2D.

5. Imágenes de tumores cerebrales en 3D

Este conjunto de datos representa el experimento más exigente en cuanto a volumen y complejidad, ya que se compone de más de 800 volúmenes tridimensionales de resonancia magnética cerebral, cada uno representado como una secuencia de cortes 2D. Las segmentaciones abarcan tres clases clínicas relevantes: *Enhancing Tumor* (región del tumor que capta contraste), *Non-Enhancing Tumor Core* (núcleo tumoral sin captación) y *Surrounding FLAIR Hyperintensity* (área peritumoral con hiperintensidad en FLAIR, relacionada con edema o infiltración).

Cada imagen contiene además cuatro canales correspondientes a distintas modalidades de RM (T1, T1Gd, T2 y FLAIR), lo que introduce un enfoque multimodal al problema. Debido a los elevados tiempos de ejecución y restricciones de memoria, solo se llevaron a cabo experimentos con las arquitecturas xLSTM-UNet_Bot y xLSTM-UNet_Enc, descartando xLSTM-UNet_EncDec. Asimismo, por razones prácticas, se utilizó validación cruzada con $k = 5$ (*5-fold cross-validation*).

El entrenamiento se realizó en dos fases: inicialmente se ejecutaron 50 épocas, y posteriormente se reanudó el entrenamiento desde el último punto de guardado (*checkpoint*) hasta alcanzar un total acumulado de 100 épocas. Esta estrategia permitió optimizar el uso de recursos computacionales sin comprometer la profundidad del entrenamiento. La Tabla 5.8 presenta la configuración detallada de los experimentos, incluyendo los hiperparámetros empleados y los tiempos de ejecución.

Arquitectura	Itr/ep.	ItrV/ep.	Ep.	Bs	Lr	T/fold (h)	T total (h)
xLSTMBot	640	160	100	1	0.001	10	50
xLSTMEnc	640	160	100	1	0.001	11	55

Cuadro 5.8: Configuración y tiempos de los experimentos con BraTS2023 3D en la tercera iteración.

En la Figura 5.10 se muestran las primeras 50 épocas de entrenamiento en los experimentos realizados sobre el conjunto BraTS2023-MEN en su representación 3D. Se comparan las arquitecturas xLSTM-UNet_Bot y xLSTM-UNet_Enc a lo largo de los 5 subconjuntos de validación cruzada. Ambas arquitecturas muestran una mejora progresiva, con tendencias ascendentes consistentes entre los diferentes subconjuntos, aunque aún lejos de una convergencia definitiva. Estas curvas sirvieron como base para justificar la reanudación del entrenamiento, lo cual permitió continuar optimizando sin reiniciar el proceso.

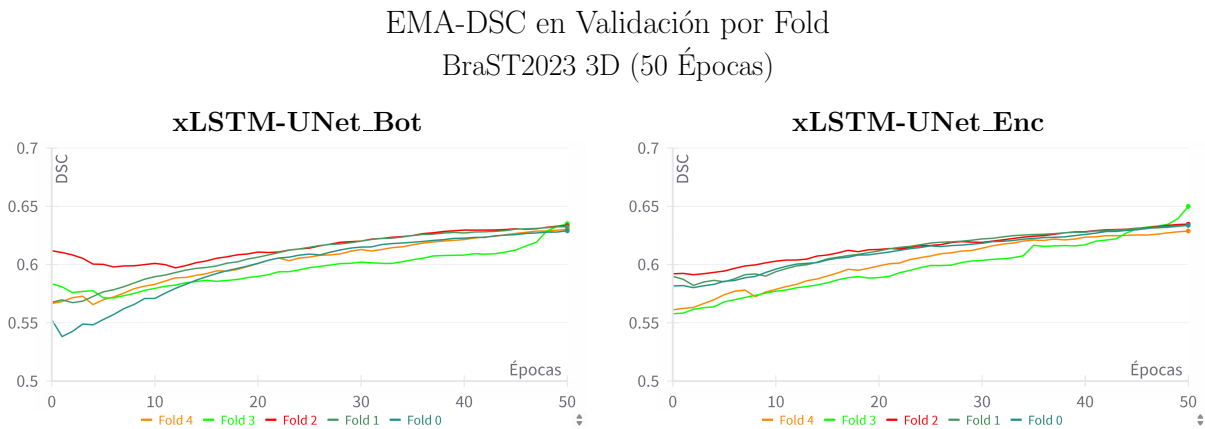


Figura 5.10: Curvas de validación (EMA-DSC) por *fold* en los experimentos con 50 épocas para el conjunto BraTS2023 3D.

Una vez ampliado y reentrenado el modelo, en la Figura 5.11 se observa el comportamiento de validación hasta las 100 épocas. A medida que el entrenamiento progresa, comienza a evidenciarse una mayor separación entre las particiones, particularmente en los subconjuntos 2, 3 y 4, los cuales destacan por alcanzar valores significativamente más altos que el resto. Esta divergencia sugiere una sensibilidad elevada a la variabilidad *interfold*, posiblemente causada por la distribución heterogénea de los subtipos tumorales o por diferencias anatómicas complejas presentes en los pacientes. Aun así, se mantiene una tendencia general de mejora continua en ambas arquitecturas.

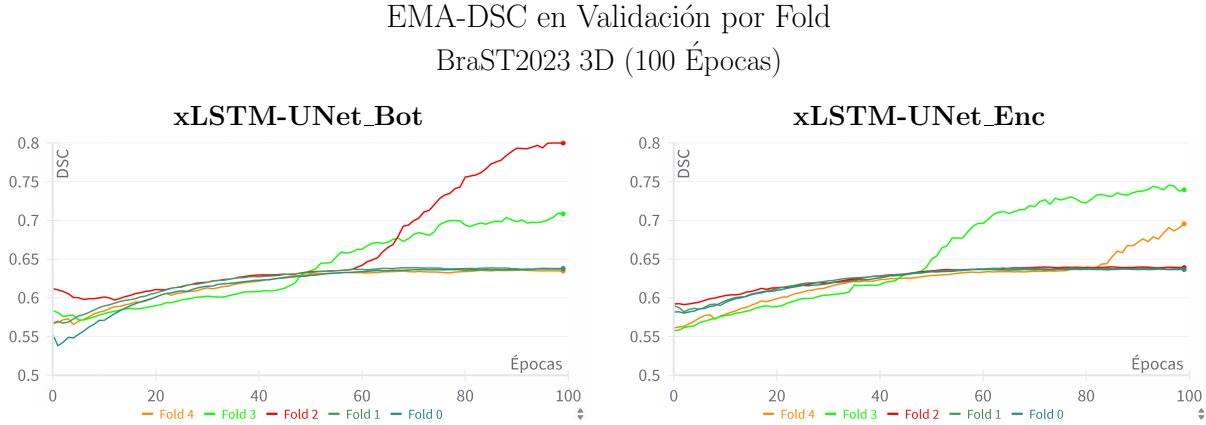


Figura 5.11: Curvas de validación (EMA-DSC) por *fold* en los experimentos con 100 épocas para el conjunto BraTS2023 3D.

Capítulo 6

Resultados

En cada una de las iteraciones realizadas durante el desarrollo del proyecto, se evaluó el rendimiento de los modelos entrenados mediante el uso de métricas específicas de segmentación. Estos análisis permitieron comparar distintas configuraciones, identificar patrones de mejora y ajuste progresivo de la arquitectura y los hiperparámetros en ciclos posteriores.

6.1. Métricas de rendimiento

Para evaluar la calidad de los resultados obtenidos por los modelos de segmentación entrenados, se utilizaron métricas estándar ampliamente reconocidas en el ámbito de la visión por computador, específicamente diseñadas para tareas de segmentación. Las métricas seleccionadas permiten cuantificar distintos aspectos del rendimiento, tales como la precisión de la segmentación, la superposición entre predicción y verdad de referencia, y la distancia entre las superficies segmentadas.

6.1.1. Coeficiente de Similaridad de Dice (DSC)

El Coeficiente de Similaridad de Dice (en inglés, *Dice Similarity Coefficient*) es una métrica utilizada para evaluar el grado de superposición entre dos segmentaciones binarias: la segmentación predicha por el modelo y la segmentación de referencia (*ground truth*). Su valor oscila entre 0 y 1, donde 1 indica una coincidencia perfecta.

$$\text{Dice} = \frac{2 \cdot |A \cap B|}{|A| + |B|} \quad (6.1)$$

Donde:

- A es la máscara de verdad de referencia (*Ground Truth*),
- B es la máscara predicha,
- $|A|$ y $|B|$ son el número de elementos verdaderos (o vóxeles positivos) en cada máscara,
- $|A \cap B|$ es el número de elementos verdaderos en la intersección de ambas máscaras (es decir, los verdaderos positivos).

Esta métrica es especialmente útil en contextos con clases desbalanceadas, como la segmentación médica, donde las regiones de interés suelen ser mucho menores que el fondo.

6.1.2. Índice de Dice de Superficie Normalizado (NSD)

El Índice de Dice de Superficie Normalizado (en inglés, *Normalized Surface Dice*) es una métrica utilizada para evaluar la calidad de segmentaciones en imágenes médicas, enfocándose en la precisión de los contornos o superficies segmentadas. A diferencia del coeficiente de Dice tradicional, que mide la superposición de volúmenes, el NSD se centra en la coincidencia de las superficies dentro de un margen de tolerancia predefinido.

$$\text{NSD}_\tau = \frac{|\partial A_\tau| + |\partial B_\tau|}{|\partial A| + |\partial B|} \quad (6.2)$$

Donde:

- ∂A es la superficie de la máscara de verdad de referencia (*Ground Truth*),
- ∂B es la superficie de la máscara predicha,
- $|\partial A|$ y $|\partial B|$ representan el área total de las superficies de cada máscara,
- ∂A_τ es el subconjunto de ∂A cuyos puntos están a una distancia menor o igual que τ de ∂B ,
- ∂B_τ es el subconjunto de ∂B cuyos puntos están a una distancia menor o igual que τ de ∂A ,
- τ es el umbral de tolerancia en milímetros definido para cada clase.

El NSD toma valores entre 0 y 1, donde un valor de 1 indica que toda la superficie predicha se encuentra dentro del margen de tolerancia con respecto a la superficie real, lo que representa una coincidencia perfecta dentro del umbral definido. Sin embargo, un valor de 0 significa que ninguna parte de la superficie predicha coincide con la real dentro de dicho margen.

En este trabajo se utilizaron diferentes valores de tolerancia según el conjunto de datos. Para el conjunto AbdomenMRI, se definieron tolerancias específicas por órgano, tales como 5 mm para el hígado y el páncreas, 3 mm para los riñones, bazo y esófago, 2 mm para vasos como la aorta, la vena cava inferior, las glándulas suprarrenales y la vesícula biliar, 5 mm para el estómago y 7 mm para el duodeno. Por otro lado, en el conjunto EndoVis, se empleó una tolerancia fija de 2 mm para todos los instrumentos evaluados. Las tolerancias descritas se tomaron del artículo original, lo que permite comparar directamente los resultados.

6.1.3. Precision, Sensibilidad y F1 Score

En el caso particular del conjunto de datos de imágenes de microscopía para segmentación celular, proporcionado por el desafío de segmentación celular de NeurIPS 2022, se empleó la métrica F1 score como criterio principal de evaluación. A diferencia de otras tareas de segmentación médica abordadas en este proyecto, donde se utilizaron métricas como el DSC o el NSD, la evaluación de la segmentación de células se realizó mediante el F1 Score debido a la naturaleza específica de este problema.

La segmentación celular suele implicar un fuerte desequilibrio entre clases, ya que las regiones correspondientes a los bordes representan una fracción muy pequeña de la célula. Este desbalance puede afectar negativamente la eficacia de métricas basadas en la superposición de volumen como el DSC, que no penalizan de forma suficientemente diferenciada los falsos positivos y falsos negativos. Por el contrario, el F1 score, al combinar de manera equilibrada la precisión y la sensibilidad, proporciona una medida más adecuada y robusta del rendimiento del modelo en estos contextos. Las métricas involucradas se definen de la siguiente manera:

$$\text{Precisión} = \frac{|A \cap B|}{|B|} \quad (6.3)$$

$$\text{Sensibilidad (Recall)} = \frac{|A \cap B|}{|A|} \quad (6.4)$$

$$\text{F1 Score} = \frac{2 \cdot |A \cap B|}{|A| + |B|} = \frac{2TP}{2TP + FP + FN} \quad (6.5)$$

Donde:

- A es la máscara de verdad de referencia (*Ground Truth*),
- B es la máscara predicha,
- $|A|$ y $|B|$ representan el número de elementos positivos en cada una,
- $|A \cap B|$ corresponde a los verdaderos positivos (TP),
- FP son los falsos positivos, y FN los falsos negativos.

La métrica F1 oscila entre 0 y 1, donde un valor de 1 indica una predicción perfecta y un valor de 0 indica que no hubo coincidencia entre las predicciones y la referencia.

6.2. Resultados de la primera iteración

Durante la Sección 5.5.1, se explicaron los experimentos preliminares desarrollados en la plataforma de Google Colaboratory. En total, se trabajó con tres conjuntos de datos, siendo uno de ellos (Resonancias magnéticas del abdomen) empleado en dos modalidades: 2D y 3D.

6.2.1. Resonancias magnéticas del abdomen en 2D

Se analizaron cinco configuraciones experimentales correspondientes al conjunto de datos AbdomenMRI en su modalidad 2D (configuraciones 1 a 5 en la Tabla 5.1). Las pruebas incluyeron variaciones en la arquitectura utilizada (xLSTMEnc frente a xLSTMBot), así como en el número de épocas, tamaño de lote y tasa de aprendizaje.

Los mejores resultados se obtuvieron en la Configuración 4, que empleó la arquitectura xLSTMBot, con un total de 75 épocas y un *batch size* de 30. Esta configuración alcanzó un valor de DSC de 0.696 y un NSD de 0.759, superando al resto de configuraciones evaluadas.

Experimento	Arquitectura	DSC	NSD
1. AbdomenMRI (2D)	xLSTMBot	0.050	0.037
2. AbdomenMRI (2D)	xLSTMEnc	0.557	0.575
3. AbdomenMRI (2D)	xLSTMBot	0.679	0.743
4. AbdomenMRI (2D)	xLSTMBot	0.696	0.759
5. AbdomenMRI (2D)	xLSTMEnc	0.616	0.643

Cuadro 6.1: Resultados de DSC y NSD para experimentos con AbdomenMRI (2D) en la iteración 1.

6.2.2. Resonancias magnéticas del abdomen en 3D

Dado el elevado consumo de recursos computacionales asociado al procesamiento de imágenes tridimensionales, en esta modalidad únicamente se evaluó la arquitectura *xLSTM-Bottleneck*, la cual presenta un menor uso de memoria en comparación con otras variantes. Los resultados obtenidos se presentan en la Tabla 6.2. La mejor configuración se observó en el experimento 10, en el cual se utilizaron 75 épocas de entrenamiento y un tamaño de lote de 2. Esta configuración alcanzó un valor de DSC de 0.674 y un NSD de 0.723, superando al resto de experimentos evaluados en esta modalidad.

Experimento	DSC	NSD
6. AbdomenMRI (3D)	0.068	0.066
7. AbdomenMRI (3D)	0.307	0.31
8. AbdomenMRI (3D)	0.351	0.354
9. AbdomenMRI (3D)	0.619	0.664
10. AbdomenMRI(3D)	0.674	0.723

Cuadro 6.2: Resultados de DSC y NSD para experimentos con AbdomenMRI (3D) en la iteración 1.

6.2.3. Imágenes de instrumentos de endoscopia

Para el conjunto de datos de instrumentos de endoscopia, que representa el mayor volumen de imágenes de entrenamiento entre los analizados, se optó por utilizar únicamente la arquitectura *xLSTMBottleneck*, debido a las restricciones impuestas por los recursos computacionales disponibles.

Los resultados obtenidos se presentan en la Tabla 6.3. Este conjunto de datos plantea un reto particular: si bien el modelo logra segmentar correctamente la presencia de instrumentos quirúrgicos en las imágenes, no logra distinguir adecuadamente entre los 7 tipos diferentes de instrumentos incluidos en las etiquetas. Como consecuencia, durante la fase de inferencia, el modelo tiende a predecir múltiples instrumentos simultáneamente, incluso cuando la imagen de entrada contiene solamente uno. Esta confusión entre clases conduce a valores reducidos

en las métricas de evaluación, como el coeficiente de Dice (DSC) y la distancia normalizada (NSD).

Experimento	DSC	NSD
11. Endovis (2D)	0.184	0.196
12. Endovis (2D)	0.328	0.343
13. Endovis (2D)	0.329	0.343

Cuadro 6.3: Resultados de DSC y NSD para experimentos con Endovis (2D) en la iteración 1.

6.2.4. Imágenes microscópicas

Para concluir esta primera iteración experimental, se llevaron a cabo pruebas sobre el conjunto de datos NeurIPS, orientadas a la tarea de segmentación celular. Este conjunto está compuesto por imágenes etiquetadas provenientes de más de 50 experimentos biológicos independientes, lo que introduce una elevada variabilidad en las características visuales de las muestras.

Dado el contexto biológico del problema, se consideró pertinente evaluar el rendimiento del modelo no solo mediante el coeficiente de similitud de Dice (DSC), sino también a través del *F1 Score*. En consecuencia, un modelo que prediga con mayor precisión el interior celular puede obtener un valor de DSC elevado, aunque no refleje un rendimiento realista en la segmentación total. Por este motivo, el *F1 Score* se incluye como métrica complementaria, al combinar de forma equilibrada la precisión y la sensibilidad. Esto permite una evaluación más robusta del desempeño del modelo en contextos donde las clases minoritarias (como los bordes celulares) son relevantes para la tarea. Los resultados de esta serie de experimentos se resumen en la Tabla 6.4. La mejor configuración se observó en el experimento 17, empleando la arquitectura *xLSTMEnc*, con un valor de DSC de 0.717 y un F1 Score de 0.486.

Experimento	Arquitectura	DSC	F1
14. NeurIPS (2D)	xLSTMBot	0.669	0.339
15. NeurIPS (2D)	xLSTMEnc	0.684	0.339
16. NeurIPS (2D)	xLSTMBot	0.708	0.477
17. NeurIPS (2D)	xLSTMEnc	0.717	0.486

Cuadro 6.4: Resultados de DSC y F1 Score para experimentos con NeurIPS (2D) en la iteración 1.

6.3. Resultados de la segunda iteración

En la segunda iteración (descrita en la Sección 5.5.2), se planteó como objetivo principal analizar de forma evolutiva el rendimiento de los modelos a lo largo del entrenamiento, evaluándolos en intervalos regulares de 20 épocas. Para ello, se seleccionaron tres experimentos representativos, correspondientes a tres modalidades distintas. A continuación, se presentan los resultados obtenidos para cada una de estas modalidades, incluyendo la evolución de las métricas de evaluación a lo largo del entrenamiento y las observaciones más relevantes derivadas de los experimentos.

6.3.1. Resonancias magnéticas del abdomen en 2D

Para esta modalidad, se realizó un entrenamiento progresivo de hasta 180 épocas analizando el comportamiento cada 20 épocas. Las métricas consideradas fueron el coeficiente de similitud de Dice (DSC) y la distancia de superficie normalizada (NSD), las cuales se utilizaron para valorar la calidad de la segmentación.

La Figura 6.1 presenta la evolución de ambas métricas en el conjunto de prueba. Se observa un incremento significativo durante las primeras 60 épocas, especialmente entre las épocas 20 y 60, donde se produce la mayor ganancia en precisión. A partir de la época 80, ambas curvas tienden a estabilizarse, alcanzando un valor máximo de 0.731 para DSC y 0.789 para NSD.

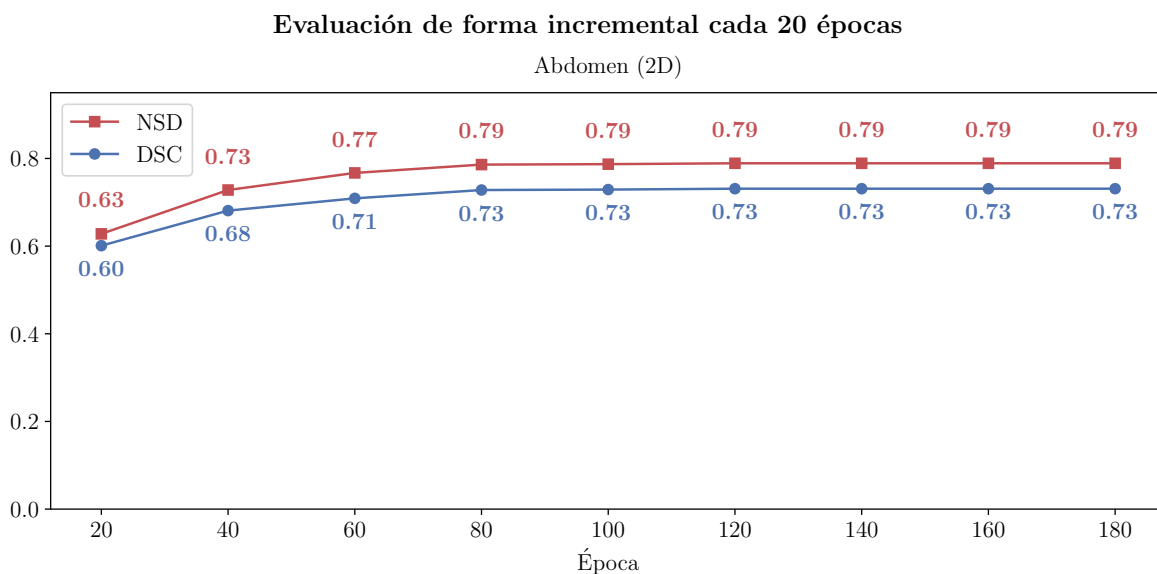


Figura 6.1: Evolución de las métricas DSC y NSD del entrenamiento incremental del modelo en AbdomenMRI (2D), evaluado cada 20 épocas.

Además del análisis numérico, la Figura 6.2 muestra un ejemplo cualitativo. Se aprecia cómo, con el avance de las épocas, el modelo no solo mejora la definición de los bordes entre estructuras, sino que también comienza a capturar regiones pequeñas que inicialmente eran ignoradas o mal segmentadas.

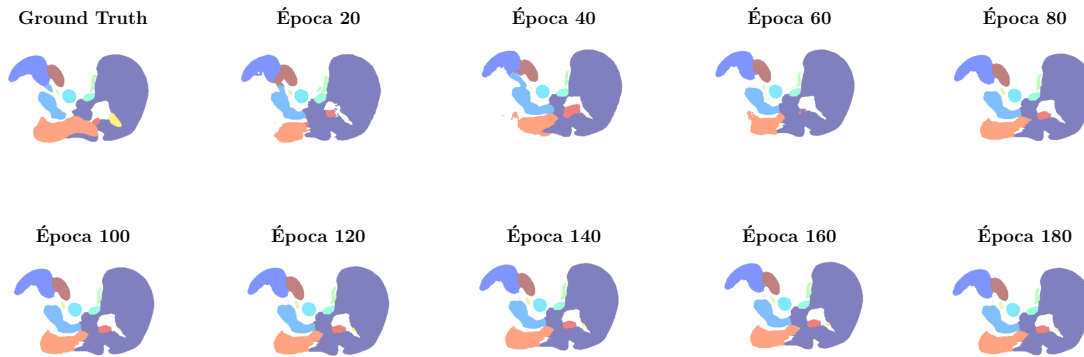


Figura 6.2: Ejemplo cualitativo de segmentación para la modalidad AbdomenMRI (2D) evaluando el progreso cada 20 épocas.

6.3.2. Resonancias magnéticas del abdomen en 3D

En el caso de las resonancias magnéticas abdominales en 3D, el modelo muestra un rendimiento notablemente superior en comparación con la modalidad 2D. Esto se debe a que las imágenes 3D contienen más información por representar directamente el volumen completo del abdomen, mientras que en 2D se pierde parte de esa información al analizar un corte de forma individual. Gracias a esta ventaja, el modelo puede aprender mejor la forma y disposición de los órganos, lo que mejora la precisión de la segmentación. La Figura 6.3 muestra la evolución de las métricas DSC y NSD a lo largo del entrenamiento incremental del modelo. Se observa un crecimiento constante del rendimiento hasta aproximadamente la época 100, donde ambas métricas alcanzan una meseta, estabilizándose en torno a 0.76 para NSD y 0.71 para DSC.

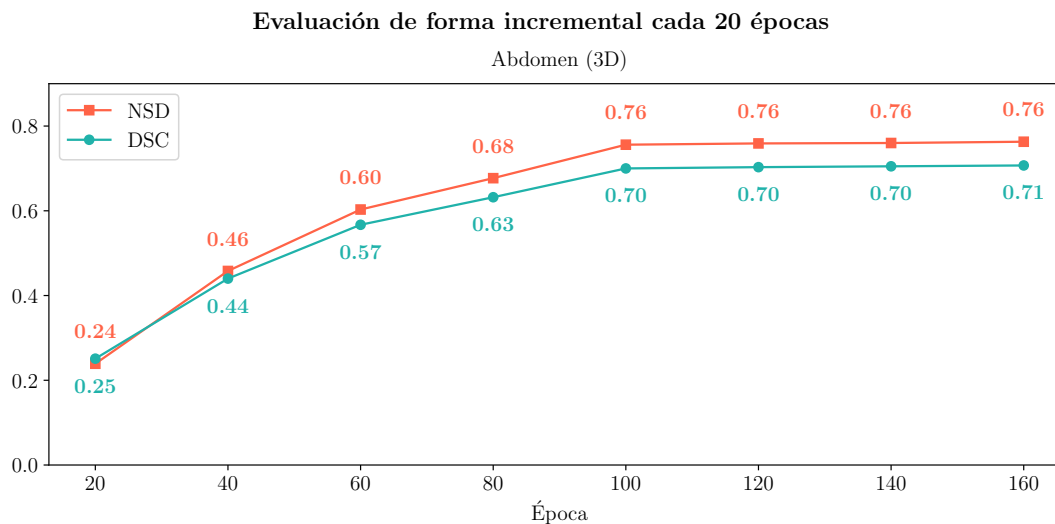


Figura 6.3: Evolución de las métricas DSC y NSD del entrenamiento incremental del modelo en AbdomenMRI (3D), evaluado cada 20 épocas.

Al igual que para las resonancias en 2D, el análisis cualitativo en 3D se realiza para visualizar el progreso del modelo no solo a través de métricas numéricas, sino también mediante un ejemplo visual. La Figura 6.4 muestra cortes axiales de distintas secciones del volumen en las diferentes etapas del entrenamiento. En total, cada estudio contiene 32 cortes, de los cuales se seleccionaron aquellos en los que la superficie de los órganos segmentados es más visible y donde la información varía significativamente entre cortes. Esto se debe a que, en imágenes 3D, cada corte representa una porción del volumen completo en un plano transversal, y cada uno aporta información complementaria sobre la forma y posición de los órganos. Se incluyen los cortes 8, 16, 20 y 24, en los que se puede apreciar cómo el modelo mejora progresivamente en la identificación y delimitación de estructuras complejas. A medida que avanza el entrenamiento, se observa una segmentación más precisa, especialmente hasta la época 100, tras la cual los cambios se estabilizan.

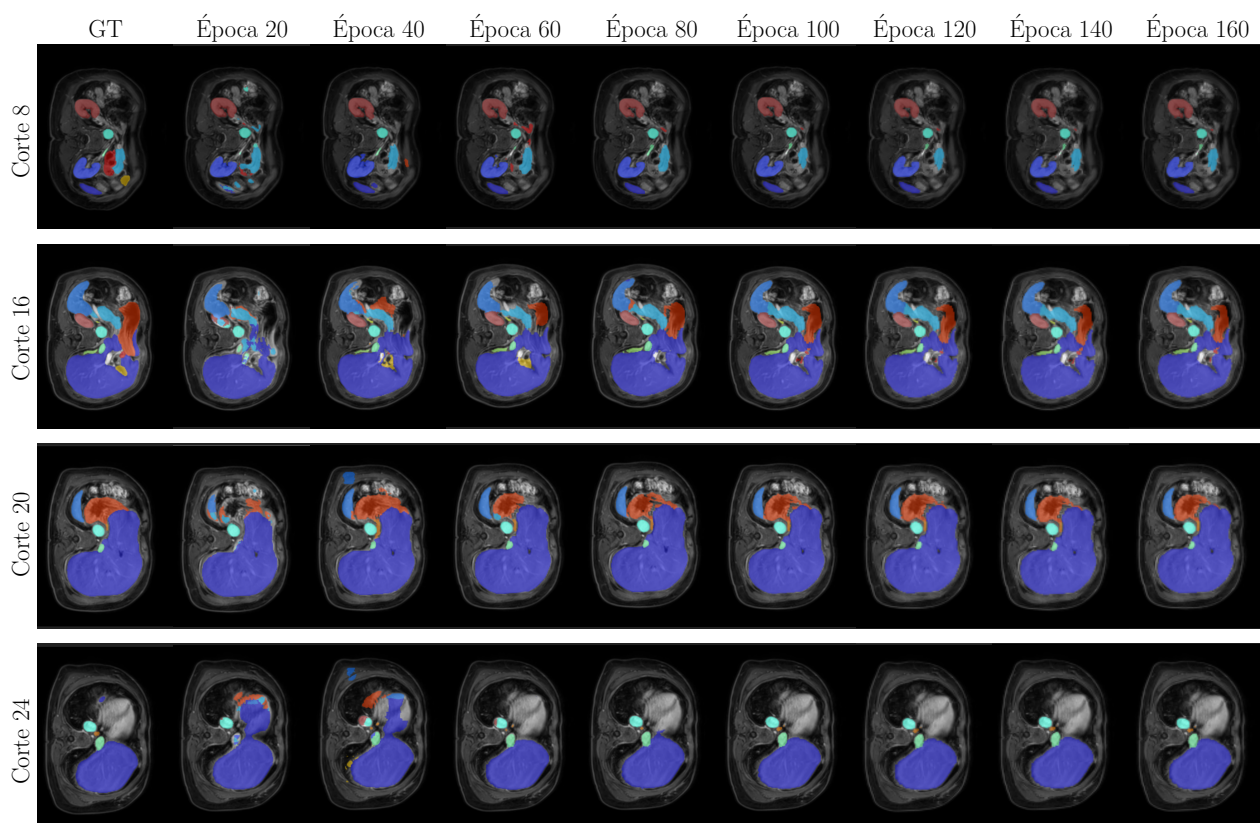


Figura 6.4: Ejemplo cualitativo de segmentación para la modalidad AbdomenMRI (3D), evaluando el progreso cada 20 épocas.

6.3.3. Imágenes de instrumentos de endoscopia

Cabe recordar que este conjunto de datos no solo requiere que el modelo segmente correctamente los instrumentos quirúrgicos, sino que también los distinga entre sí, abarcando un total de siete instrumentos diferentes que no aparecen de forma simultánea en las imágenes. Además, el dataset Endovis, como ya se ha mencionado, contiene una mayor cantidad de muestras tanto para entrenamiento como para evaluación, lo que incrementa la complejidad

del problema, ya que el modelo debe abstraer patrones generales a partir de una mayor variabilidad. Sin embargo, como se observa en la Figura 6.5, el modelo muestra una mejora progresiva en su rendimiento sobre el conjunto de prueba a medida que avanza el entrenamiento. No obstante, dicha mejora es considerablemente lenta en comparación con otros conjuntos de datos. Aun así, esta evolución refleja la capacidad del modelo para adaptarse y aprender de la información disponible a lo largo de las épocas.

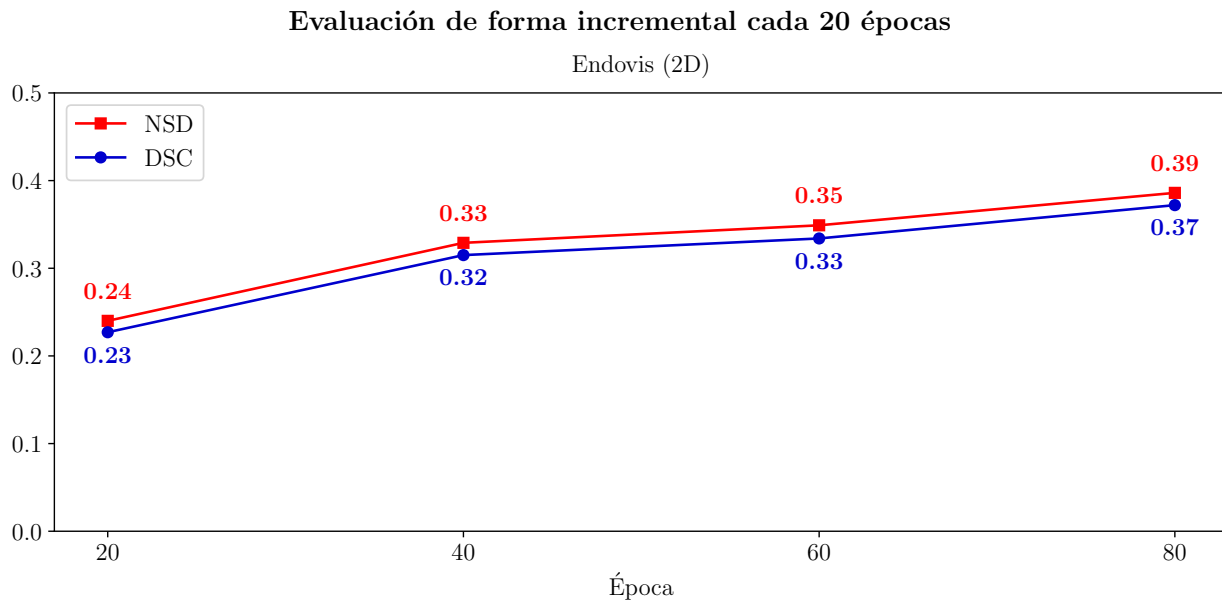


Figura 6.5: Evolución de las métricas DSC y NSD del entrenamiento incremental del modelo en Endovis (2D), evaluado cada 20 épocas.

El análisis cualitativo para este conjunto de datos se presenta en la Figura 6.6, donde se seleccionan tres imágenes representativas con distintos instrumentos quirúrgicos. En cada fila se muestra la imagen original con su segmentación de referencia (*Ground Truth*) a la izquierda, seguida de las predicciones del modelo a lo largo del entrenamiento (épocas 20, 40, 60 y 80). Cada clase de instrumento está representada mediante un color distinto, mientras que las regiones oscuras o mal segmentadas indican áreas sin predicción o con confusión entre clases.

Este ejemplo plasma lo ya comentado y muestra que el modelo es capaz de identificar de forma estructural las regiones correspondientes a los instrumentos, logrando una segmentación coherente. Sin embargo, la predicción semántica sigue siendo un desafío: el modelo tiende a identificar más clases de las que realmente están presentes, como se observa en las predicciones donde aparecen instrumentos no incluidos en la segmentación real. Esta sobrepredicción es indicativa de una confusión semántica, a pesar de la mejora en la precisión espacial a lo largo de las épocas.

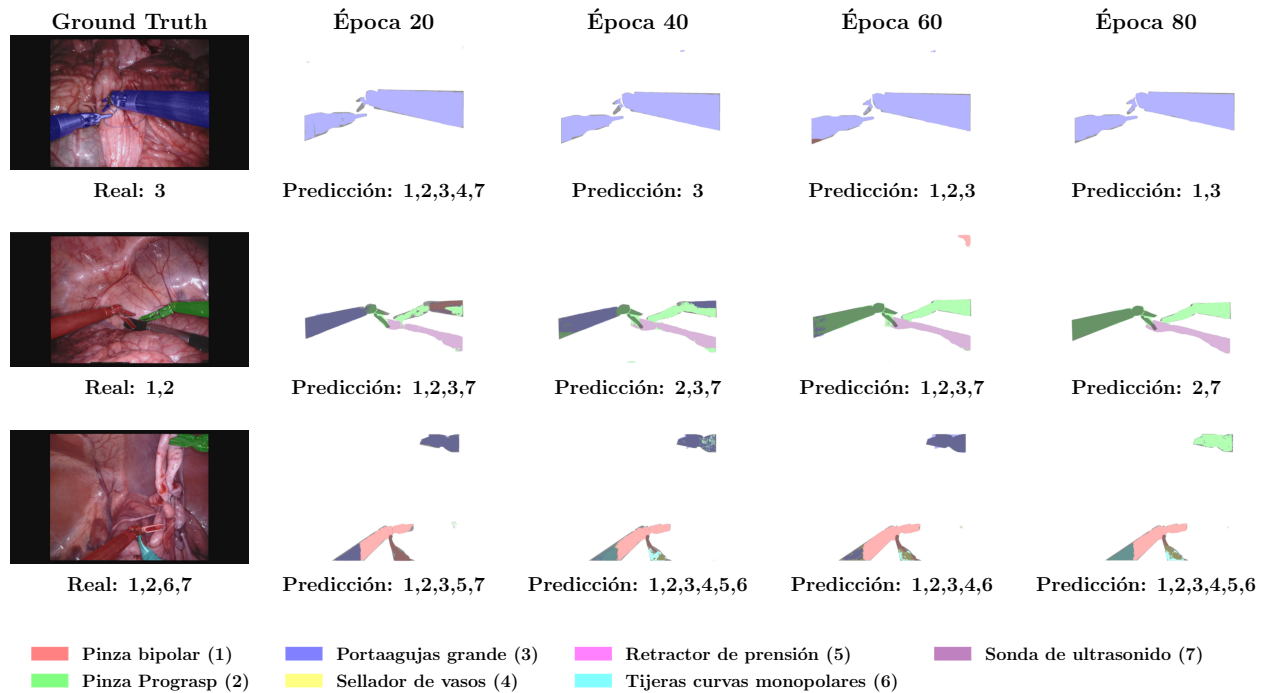


Figura 6.6: Ejemplo cualitativo del rendimiento del modelo en Endovis (2D), evaluando el progreso cada 20 épocas.

6.4. Resultados de la tercera iteración

En la última etapa de este estudio, se llevaron a cabo experimentos de larga duración, descritos en la Sección 5.5.3, para cada conjunto de datos. En estos experimentos, se empleó validación cruzada mediante k -folds para asegurar una evaluación robusta del desempeño del modelo. Posteriormente, se entrenó y desplegó el modelo en el equipo físico con el fin de realizar la inferencia y evaluación correspondientes. A continuación, se presentan los resultados obtenidos, organizados por conjunto de datos y modalidad.

6.4.1. Resonancias magnéticas del abdomen en 2D

Para este conjunto de datos, se realizaron tres experimentos de larga duración utilizando validación cruzada con $k = 10$ folds, sumando un total de 700 épocas de entrenamiento por experimento. En la Figura 6.7, se presentan los valores promedio de las métricas DSC (*Dice Similarity Coefficient*) y NSD (*Normalized Surface Dice*) mediante un gráfico de barras. Las líneas verticales en color negro representan la desviación estándar correspondiente a cada métrica entre los diferentes folds.

En general, el modelo que contiene la xLSTM en el cuello de botella (xLSTMBot) mostró un rendimiento apenas superior en ambas métricas, mientras que las demás variantes, xLSTMEnc y xLSTMEncDec, presentaron resultados muy similares entre sí. Además, cabe destacar que los modelos con un mayor número de componentes xLSTM tienden a requerir un mayor consumo de recursos computacionales. Las desviaciones típicas son pequeñas, lo que sugiere una baja variabilidad en los resultados obtenidos.

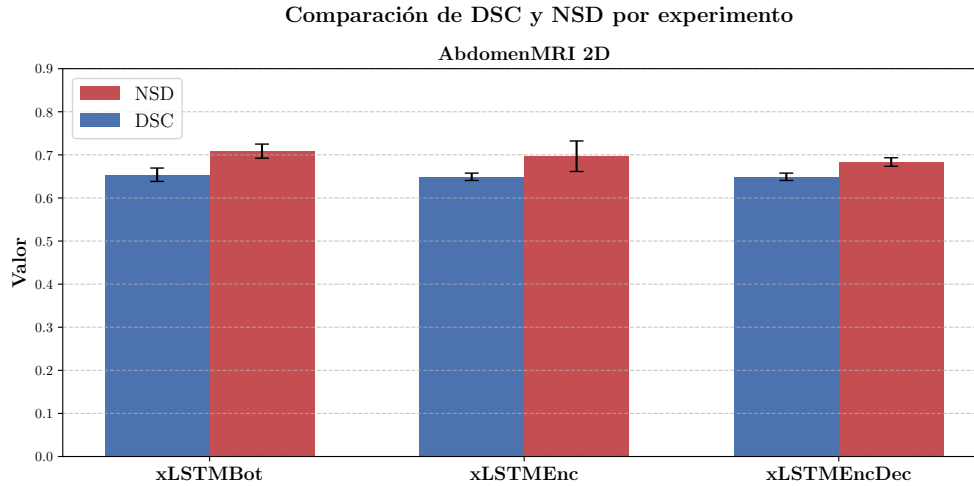


Figura 6.7: Comparación de las métricas DSC y NSD para los diferentes modelos en el conjunto de datos AbdomenMRI 2D en la tercera iteración.

Con el fin de evaluar los modelos desarrollados, se comparan sus resultados con los descritos en el artículo original y con otros métodos representativos del estado del arte disponibles en el mismo ([10]). La Tabla 6.5 muestra que, aunque las variantes propuestas (xLSTM-Bot, xLSTMEnc y xLSTMEncDec) presentan métricas DSC y NSD ligeramente inferiores en términos absolutos, destacan por una menor desviación estándar. Esto indica una mayor consistencia entre los subconjuntos de validación cruzada.

Método	DSC ↑	NSD ↑
Métodos del estado del arte		
nnU-Net	0.745 ± 0.112	0.815 ± 0.115
SegResNet	0.732 ± 0.138	0.803 ± 0.139
UNETR	0.575 ± 0.167	0.631 ± 0.186
SwinUNETR	0.703 ± 0.135	0.767 ± 0.144
U-Mamba_Bot	0.759 ± 0.105	0.829 ± 0.107
U-Mamba_Enc	0.763 ± 0.108	0.833 ± 0.109
Resultados del artículo original		
xLSTMBot	0.764 ± 0.101	0.832 ± 0.103
xLSTMEnc	0.775 ± 0.095	0.837 ± 0.095
Resultados de este trabajo		
xLSTMBot	0.654 ± 0.016	0.709 ± 0.016
xLSTMEnc	0.649 ± 0.009	0.697 ± 0.035
xLSTMEncDec	0.649 ± 0.009	0.683 ± 0.010

Cuadro 6.5: Resultados comparativos de DSC y NSD para AbdomenMRI 2D.

6.4.2. Resonancias magnéticas del abdomen en 3D

La modalidad 3D para la segmentación de órganos abdominales proporciona mayor información espacial al modelo, y las imágenes cuentan con una calidad superior en comparación con la modalidad 2D. Como resultado, los modelos alcanzan un rendimiento significativamente mejor en esta modalidad.

Sin embargo, a pesar de este incremento en rendimiento, no se logra igualar ni superar los resultados del artículo original. La Tabla 6.6 presenta una comparación entre los métodos del estado del arte, los resultados originales y los obtenidos en este trabajo. Se observa que, si bien las métricas alcanzadas por este estudio son ligeramente inferiores en términos absolutos, la desviación estándar es considerablemente menor, lo cual indica una mayor estabilidad y consistencia entre los diferentes *folds* evaluados.

Método	DSC $\uparrow$	NSD $\uparrow$
Métodos del estado del arte		
nnU-Net	0.831 ± 0.077	0.900 ± 0.073
SegResNet	0.815 ± 0.096	0.884 ± 0.092
UNETR	0.687 ± 0.149	0.744 ± 0.163
SwinUNETR	0.757 ± 0.139	0.822 ± 0.141
U-Mamba_Bot	0.845 ± 0.067	0.912 ± 0.063
Resultados del artículo original		
xLSTMBot	0.848 ± 0.077	0.915 ± 0.060
Resultados de este trabajo		
xLSTMBot	0.791 ± 0.005	0.854 ± 0.005

Cuadro 6.6: Resultados comparativos de DSC y NSD para AbdomenMRI 3D.

La Figura 6.8 muestra un ejemplo representativo de la segmentación en resonancia magnética, tomado cada 4 cortes aproximadamente de un total de 31. La predicción corresponde al modelo con mejor desempeño (validación con subconjunto número 6), con un Dice de 0.799 y un NSD de 0.862. Se observa una adecuada correspondencia con la segmentación real, aunque aún se presentan pequeños errores en ciertas estructuras.

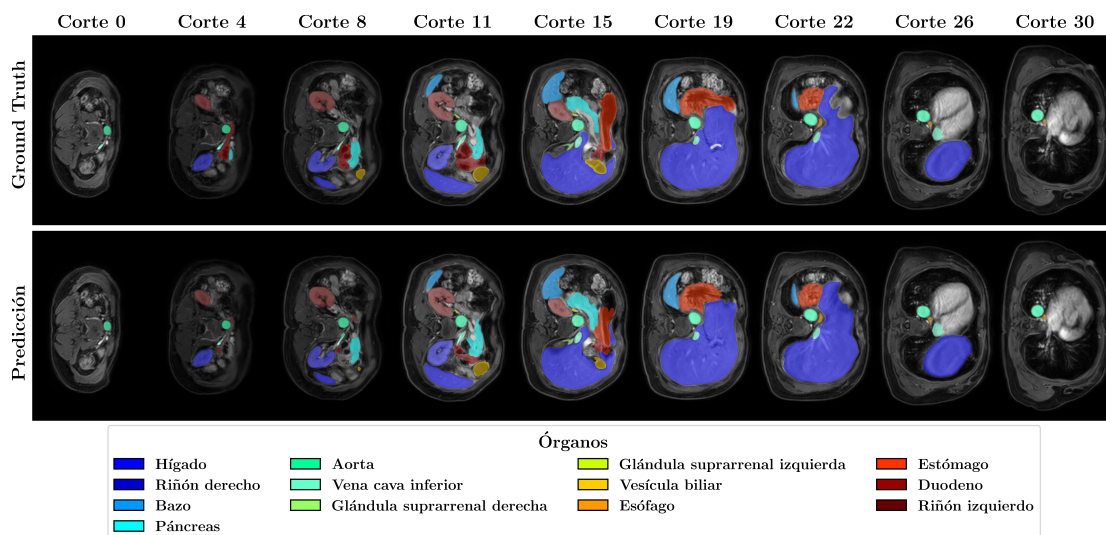


Figura 6.8: Ejemplo cualitativo de segmentación en AbdomenMRI 3D, tercera iteración.

6.4.3. Imágenes de instrumentos de endoscopia

Mientras que en la segmentación de órganos abdominales se obtuvieron modelos con un desempeño bastante preciso, en el caso de los instrumentos de endoscopia se presenta una problemática similar a la ya descrita: el modelo logra segmentar, pero no de manera semántica. Por este motivo, como se observa en la Tabla 6.7, el modelo entrenado durante este estudio se encuentra entre los de peor rendimiento.

Esta deficiencia se debe, en parte, a las limitaciones impuestas por los objetivos del trabajo de fin de grado, que impidieron la realización de experimentos más extensos. Además, el tiempo de entrenamiento del modelo era considerablemente alto, lo que obligó a reducir la validación cruzada a 5 *folds*, tal como ya se ha mencionado. A esto se suma la falta de información en el artículo original, el cual no especificaba el valor de ciertos hiperparámetros, lo que imposibilitó la reproducción exacta de los experimentos descritos.

Método	DSC $\uparrow$	NSD $\uparrow$
Métodos del estado del arte		
nnU-Net	0.626 ± 0.302	0.641 ± 0.307
SegResNet	0.582 ± 0.327	0.597 ± 0.330
UNETR	0.502 ± 0.320	0.517 ± 0.324
SwinUNETR	0.553 ± 0.309	0.568 ± 0.312
U-Mamba_Bot	0.654 ± 0.301	0.669 ± 0.305
Resultados del artículo original		
xLSTMBot	0.684 ± 0.301	0.700 ± 0.305
xLSTMEnc	0.684 ± 0.302	0.700 ± 0.307
Resultados de este trabajo		
xLSTMBot	0.469 ± 0.017	0.483 ± 0.017

Cuadro 6.7: Resultados comparativos de DSC y NSD para Endovis 2D.

Al igual que en la sección anterior, en la Figura 6.9 se presenta un ejemplo cualitativo. En ella se puede observar con claridad que, cuando hay más de una clase presente, el modelo es incapaz de distinguir entre ellas, llegando incluso a mezclar distintos instrumentos en algunos casos.

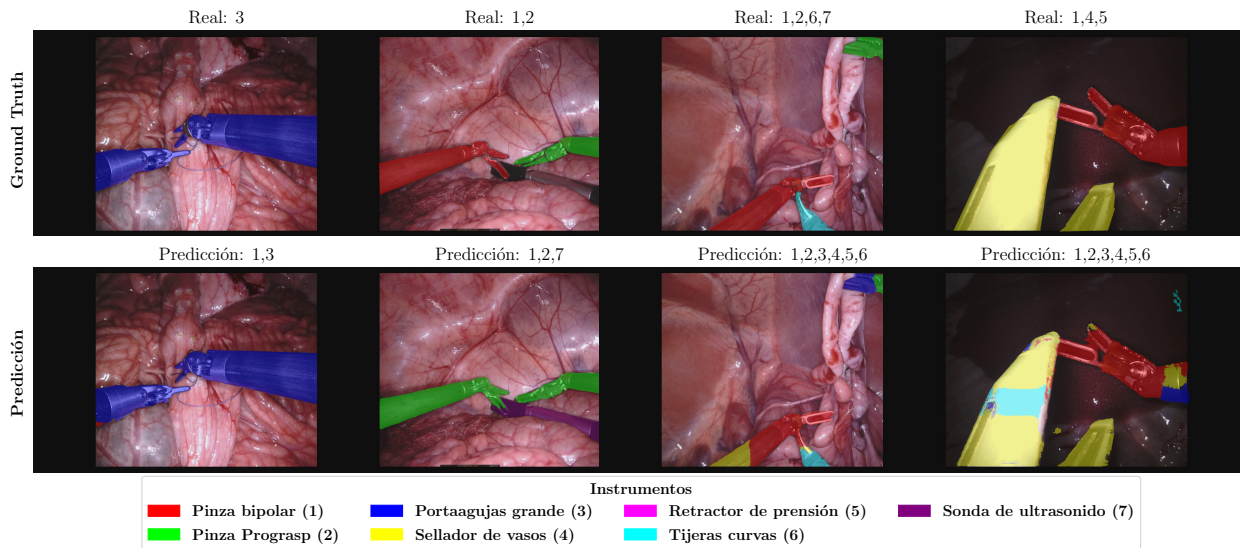


Figura 6.9: Ejemplo cualitativo de segmentación en Endovis 2D, tercera iteración.

6.4.4. Imágenes microscópicas

La problemática en torno a este conjunto de datos se mencionó desde la fase inicial del estudio. El modelo debe distinguir entre interior celular y borde celular, siendo este último una región muy reducida de la imagen, en muchos casos apenas perceptible a simple vista.

En este contexto, los resultados obtenidos en el presente trabajo muestran un desempeño competitivo, aunque por debajo de los reportados tanto en el artículo original como en algunos métodos del estado del arte. Específicamente, las arquitecturas propuestas (xLSTMBot y xLSTMEnc) alcanzan valores de F1-score más bajos, como se puede observar en la Tabla 6.8. No obstante, cabe destacar que los modelos entrenados presentan una desviación estándar considerablemente menor, lo cual sugiere una mayor estabilidad y consistencia en las predicciones, incluso si el rendimiento promedio es inferior. Esta estabilidad podría ser beneficiosa en aplicaciones donde la robustez y la reproducibilidad son prioritarias, abriendo la puerta a futuras mejoras mediante ajustes en la arquitectura o en la estrategia de entrenamiento.

Método	F1 $\uparrow$
Métodos del estado del arte	
nnU-Net	0.538 ± 0.266
SegResNet	0.541 ± 0.263
UNETR	0.436 ± 0.257
SwinUNETR	0.397 ± 0.262
U-Mamba_Bot	0.539 ± 0.282
Resultados del artículo original	
xLSTMBot	0.561 ± 0.278
xLSTMEnc	0.582 ± 0.239
Resultados de este trabajo	
xLSTMBot	0.449 ± 0.014
xLSTMEnc	0.510 ± 0.012

Cuadro 6.8: Resultados comparativos de F1-Score para NeurIPS 2D.

Sin embargo, más allá del rendimiento global, es necesario analizar el comportamiento de cada arquitectura en detalle. La Figura 6.10 muestra los valores de DSC obtenidos en cada *fold* para ambas arquitecturas. Los resultados indican que ambas variantes presentan un rendimiento muy similar a lo largo de las 10 particiones, con valores de DSC que oscilan entre aproximadamente 0.69 y 0.72. No se observa una ventaja sustancial y consistente de una arquitectura sobre la otra; sin embargo, la versión con xLSTM en el codificador muestra una leve superioridad en algunos *folds*, lo que podría sugerir una ligera ventaja en términos de generalización.

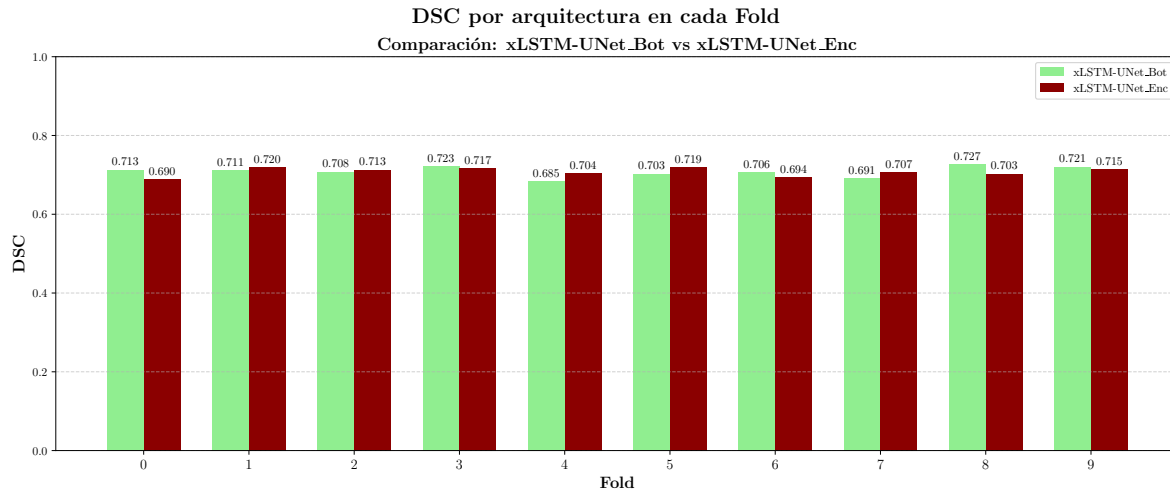


Figura 6.10: DSC por clase en cada *fold* para ambas arquitecturas en imágenes de microscopía.

Por último, la Figura 6.11 muestra tres ejemplos cualitativos de segmentación comparando las salidas de ambas arquitecturas frente a la segmentación real. La arquitectura con xLSTM en el codificador (xLSTMEnc) presenta una delineación del borde más precisa y continua en comparación con la versión con xLSTM en el cuello de botella. Cabe destacar que los ejemplos seleccionados provienen de los modelos validados con los *folds* de mayor rendimiento en el conjunto de prueba para cada arquitectura: el subconjunto 9 para xLSTMBot y el subconjunto 1 para xLSTMEnc.

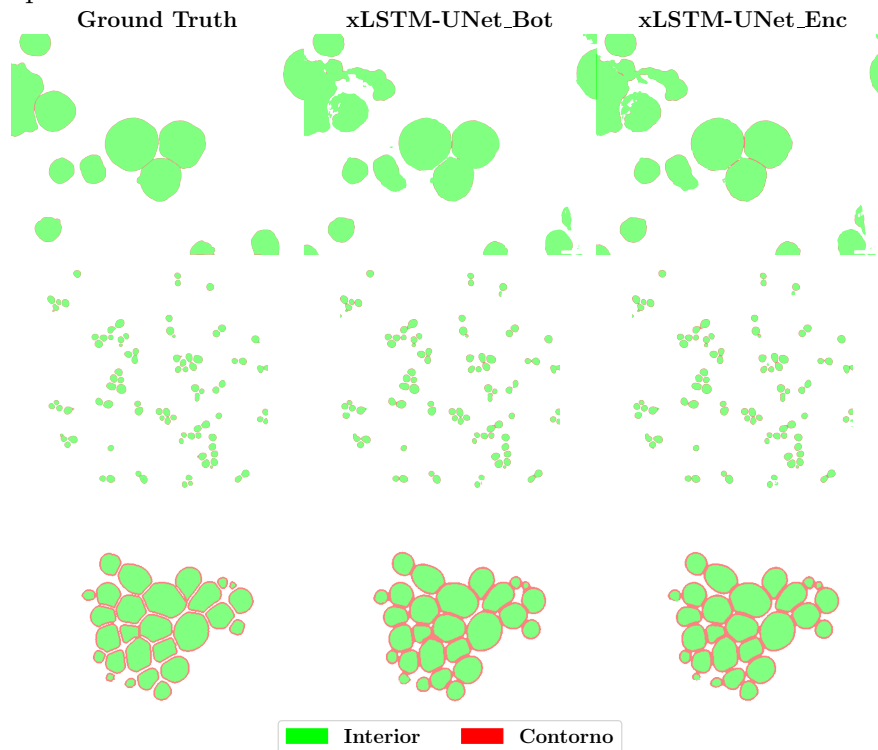


Figura 6.11: Ejemplo cualitativo de segmentación celular en NeurIPS 2D, tercera iteración.

6.4.5. Segmentación de tumores cerebrales

Para finalizar con la tercera y última iteración de este estudio, se presentan los resultados de segmentación de tumores cerebrales. El rendimiento final de los modelos entrenados durante 100 épocas muestra un coeficiente de Dice promedio de 0.47 para ambas arquitecturas. Sin embargo, esta medida no es suficientemente representativa del comportamiento real del modelo, ya que al analizar la métrica DSC por clase, se evidencia un claro desequilibrio en el rendimiento.

Por esta razón, la Figura 6.12 muestra el rendimiento promedio por clase y arquitectura en el conjunto de prueba, calculado a partir de los 5 *folds*. Se observa que ambos modelos tienen un rendimiento muy bajo en la clase Tumor con realce de contraste (*Enhancing Tumor*), siendo prácticamente incapaces de predecir dicha clase en la mayoría de los casos. Por el contrario, en la clase que representa el núcleo tumoral sin realce de contraste (*Non-Enhancing Tumor*), que corresponde a una región mucho más extensa en la imagen, ambas arquitecturas logran desempeños considerablemente superiores. Además, la desviación estándar para la clase ET es notablemente más alta, lo que indica una gran variabilidad en la capacidad del modelo para segmentar esta región. Esta inestabilidad sugiere que el modelo no logra generalizar correctamente en regiones pequeñas y con menor representación espacial, lo cual sigue siendo un reto importante en segmentación médica.

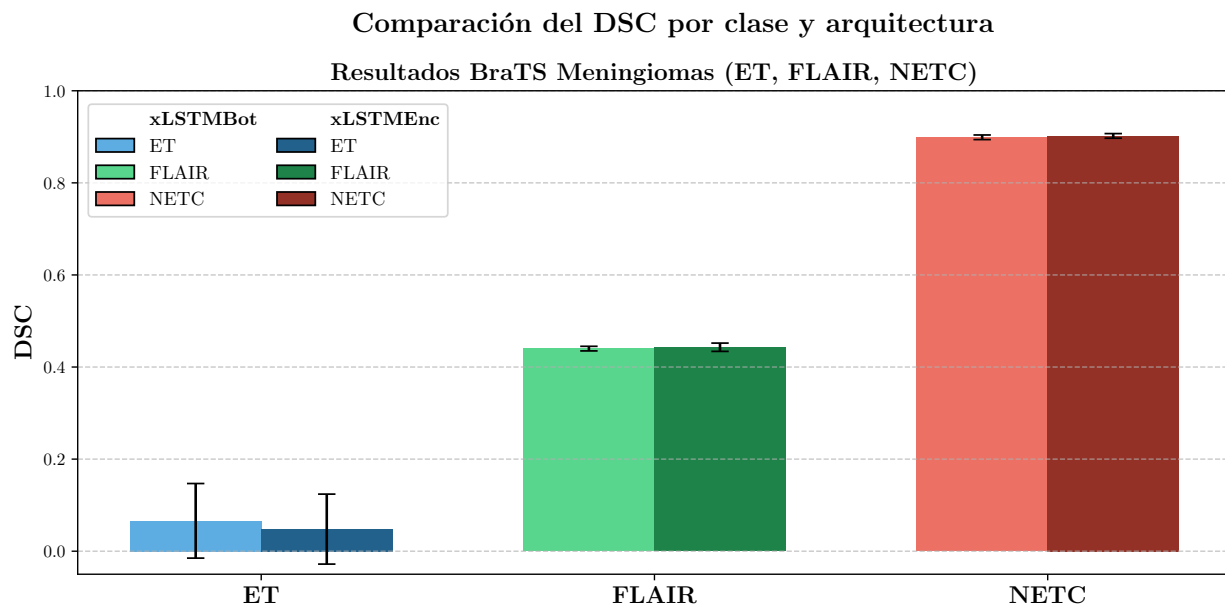


Figura 6.12: Comparación del DSC por clase (ET, FLAIR, NETC) entre las arquitecturas xLSTMBot y xLSTMEnc en el conjunto de prueba de BraTS2023-MEN.

Este conjunto de datos tridimensional cuenta con cuatro canales de entrada correspondientes a las distintas modalidades de resonancia magnética (T1, T1Gd, T2 y FLAIR). En la Figura 6.13 se muestra un corte representativo que contiene información visible de las tres clases a segmentar. Las imágenes corresponden a las predicciones generadas por el modelo de cada arquitectura que obtuvo el mejor rendimiento en el conjunto de prueba. En este corte se aprecia cómo, aunque de forma parcial y limitada, la arquitectura que incorpora el módulo

xLSTM en el codificador logra identificar regiones asociadas a la clase más conflictiva: el tumor con realce de contraste (ET). Esta clase, al ser de menor tamaño y más difícil de distinguir, tiende a ser ignorada por otros modelos, pero en este caso comienza a ser detectada, lo que sugiere una leve mejora.

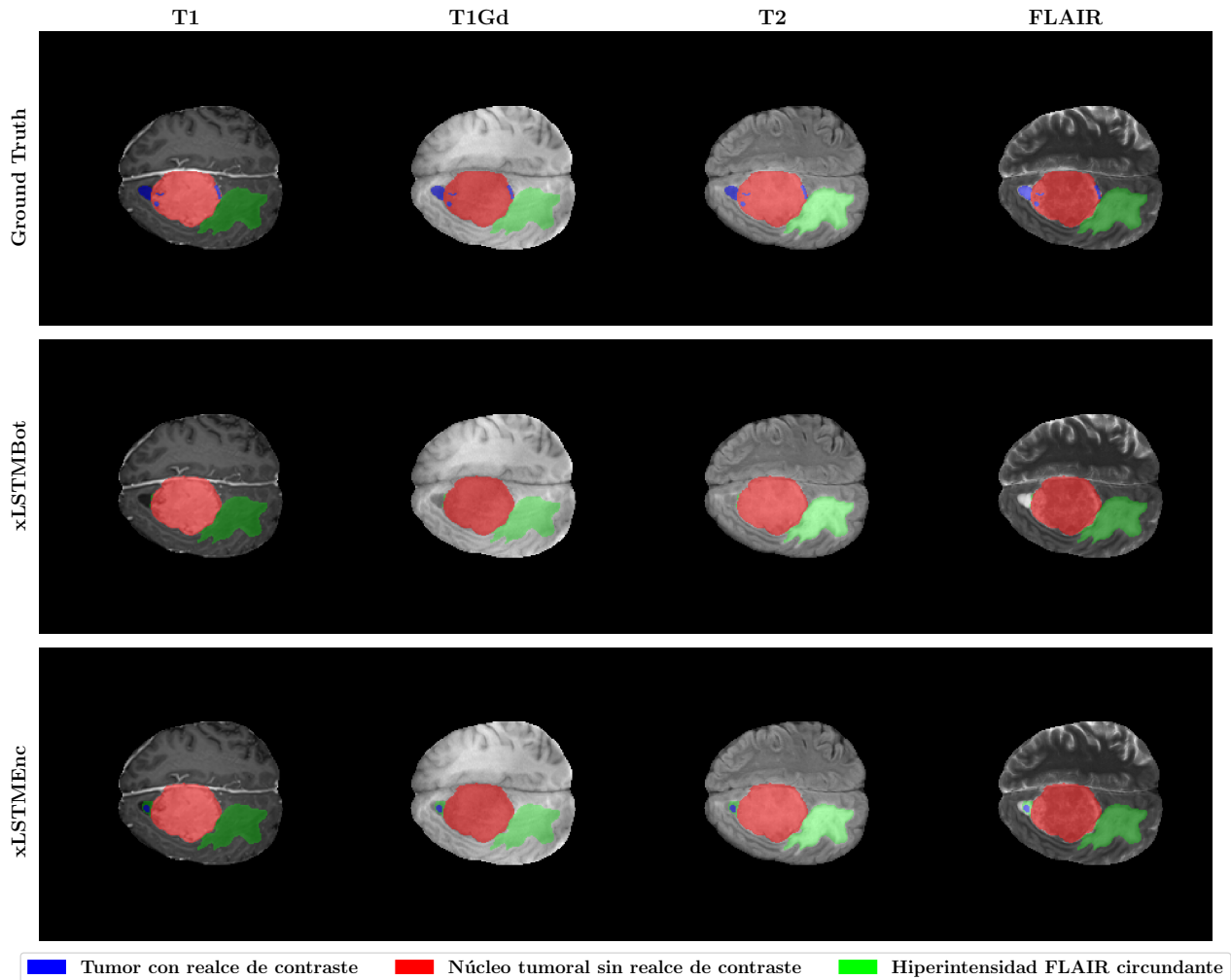


Figura 6.13: Ejemplo cualitativo de segmentación en un corte axial con las cuatro modalidades de entrada (T1, T1Gd, T2, FLAIR).

Por último, para visualizar la estructura tridimensional del volumen, se escogió la modalidad T1, ya que proporciona una vista anatómica general del cerebro con buen contraste entre sustancia gris y blanca. La Figura 6.14 muestra un ejemplo tridimensional representado mediante 9 cortes axiales seleccionados de los 155 que componen cada volumen.

En esta figura se ilustran las predicciones generadas por los modelos con mejor rendimiento en el conjunto de prueba para cada arquitectura, en comparación con la segmentación real. A lo largo de los cortes puede observarse la distribución espacial de las tres clases objetivo: tumor con realce de contraste (ET), núcleo tumoral sin realce (NETC) e hiperintensidad FLAIR circundante.

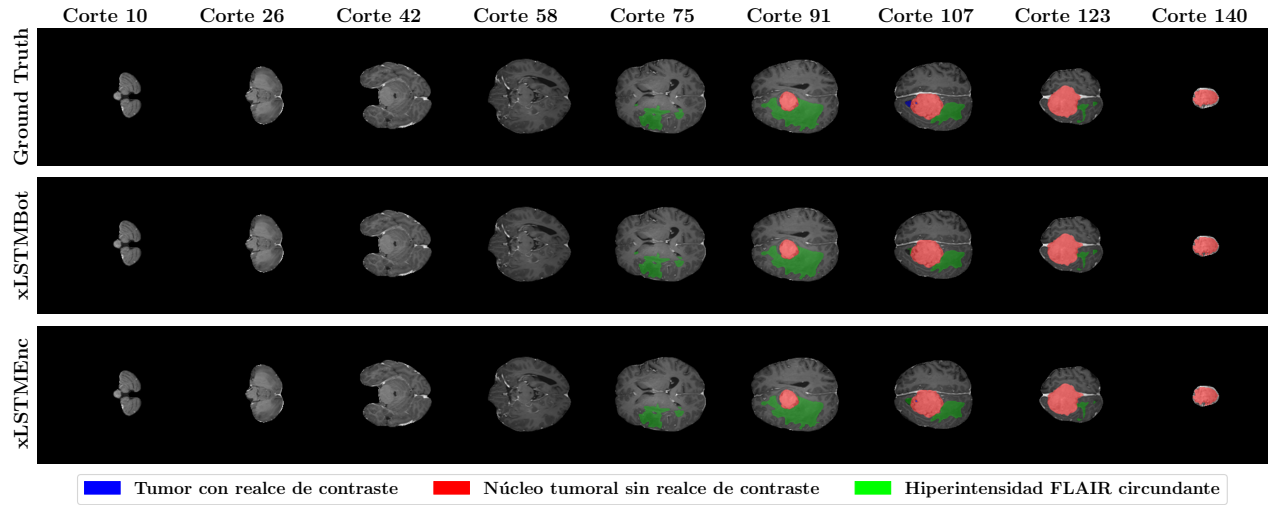


Figura 6.14: Ejemplo tridimensional de segmentación tumoral en modalidad T1 mediante nueve cortes seleccionados.

Una vez expuesto todo lo anterior, se finaliza la presentación de los resultados obtenidos en este trabajo. En ellos se abarca tanto los análisis cuantitativos como cualitativos para cada uno de los conjuntos de datos utilizados. Se ha llevado a cabo una comparación de los modelos propuestos con métodos del estado del arte y con los resultados del artículo de referencia. En el siguiente capítulo, se abordará una discusión más amplia sobre ellos y sus implicaciones.

Capítulo 7

Conclusiones

7.1. Conclusiones

En términos generales, en este trabajo se entrenaron y se evaluaron tres arquitecturas híbridas de U-Net con bloques xLSTM. La variante con bloques xLSTM en la etapa de codificación, en general, fue la más eficaz. Además, el enfoque iterativo del trabajo permitió optimizar el rendimiento de los modelos: la iteración hace que sea más fácil evaluar y validar progresivamente en cada mejora. Esto favoreció la implementación y ajuste continuo de las arquitecturas, avanzando hacia los objetivos planteados.

La capacidad de generalización y la robustez de los modelos se evaluaron a través de múltiples conjuntos de datos médicos: Abdomen MRI (órganos abdominales en 2D y 3D), Endovis (instrumentos quirúrgicos), NeurIPS (segmentación celular) y BraTS2023 (tumores cerebrales). Estos conjuntos de datos heterogéneos permitieron exponer las arquitecturas a condiciones variadas y complejas.

En el caso del conjunto de datos NeurIPS, centrado en la segmentación celular, los resultados obtenidos muestran que las variantes xLSTM-UNet_Bot y xLSTM-UNet_Enc alcanzaron valores de F1-score inferiores a los reportados tanto por el artículo original como por algunos métodos del estado del arte. A pesar de ello, se observó una desviación estándar notablemente menor, lo que sugiere una mayor estabilidad y consistencia en las predicciones. Esto indica que, si bien el rendimiento medio fue más bajo, los modelos se comportaron de forma más regular a lo largo de los distintos *folds*.

En cuanto al conjunto de segmentación de órganos abdominales (Abdomen MRI), se observó que el rendimiento de los modelos es ligeramente inferior al del artículo de referencia en el que se basó la comparación; sin embargo, se obtuvo una desviación estándar menor en las métricas de evaluación. Esta reducción de la variabilidad significa que el modelo fue capaz de hacer las proyecciones más estables en este dominio.

Respecto al conjunto Endovis, centrado en la segmentación de instrumentos quirúrgicos, el modelo fue capaz de detectar la presencia de estos elementos. A pesar de ello, el modelo no logró clasificarlos según las categorías definidas. Este resultado pone en evidencia una dificultad en la segmentación semántica de instrumentos quirúrgicos: diferenciar con precisión entre objetos similares no es una tarea sencilla. Por lo tanto, la arquitectura propuesta logró la detección de instrumentos en general, aunque mostró dificultades para distinguir sus tipos específicos en la escena.

Los experimentos con el conjunto BraTS2023, destinado a la segmentación de tumores cerebrales en 3D, revelaron varias limitaciones desde el punto de vista de recursos compu-

tacionales. En particular, el tamaño y complejidad de las imágenes volumétricas multicanal implican un uso intensivo de memoria GPU, lo que restringe la posibilidad de entrenar modelos complejos en entornos con capacidades limitadas. Las variantes xLSTM-UNet_Bot y xLSTM-UNet_Enc se entrenaron y mostraron un rendimiento relativamente consistente entre sí. Sin embargo, sus métricas globales resultaron significativamente más bajas en comparación con otros conjuntos de datos.

Por otro lado, este trabajo incluye una documentación detallada y en él se realiza un análisis cualitativo y visual comprensivo de los resultados. Estas herramientas de análisis utilizadas ayudan a comprender mejor el comportamiento de los modelos en distintos casos de prueba y validar las conclusiones cuantitativas obtenidas. Además, la documentación detallada de las arquitecturas de los modelos, el desarrollo de las fases y los resultados obtenidos en cada una de ellas facilita la interpretación y comprensión de la investigación.

Es importante señalar que el artículo de referencia original no era completamente reproducible con la información proporcionada, por lo que se adoptó una estrategia empírica basada en ajustes iterativos justificados por resultados observables. Gracias a este enfoque, se resolvieron las dudas que se presentaban y se ajustaron los hiperparámetros poco a poco, validando cada cambio con pruebas específicas. Así, aunque al principio hubo algunos desafíos para reproducir los resultados, la metodología aplicada permitió llevar a cabo una implementación clara y alineada con los objetivos definidos.

En conjunto, se cumplieron los objetivos propuestos: se implementaron y evaluaron las variantes arquitectónicas, y se demostró que la combinación de U-Net con bloques xLSTM puede superar a esta última en tareas de segmentación médica compleja. Se identificaron las fortalezas y debilidades de cada enfoque en los diferentes escenarios evaluados, lo que permite afirmar que las metas establecidas inicialmente se lograron correctamente dentro de los alcances definidos.

7.2. Principales aportaciones

Este trabajo contribuyó al ámbito de la segmentación de imágenes médicas mediante la implementación, evaluación y despliegue de arquitecturas híbridas que combinan la red U-Net con bloques xLSTM. Una de las aportaciones más destacadas de este trabajo es la propuesta y validación de una nueva variante arquitectónica: xLSTM-UNet_EncDec. Esta propuesta, inspirada en el modelo ViLU-Net, se adaptó para funcionar de manera eficiente dentro de las limitaciones computacionales del entorno. A diferencia de las variantes xLSTM-UNet_Bot y xLSTM-UNet_Enc, que ya se encontraban disponibles en los repositorios oficiales, la arquitectura xLSTM-UNet_EncDec se desarrolló en este proyecto, incorporando bloques xLSTM tanto en la etapa de codificación como en la de decodificación.

Otro componente clave del trabajo es su enfoque de desarrollo iterativo. A lo largo del proyecto, se desarrollaron tres ciclos de experimentación que abarcan desde el ajuste de hiperparámetros no especificados en el artículo de referencia y la evaluación periódica de los modelos cada 20 épocas, hasta la realización de experimentos de larga duración empleando validación cruzada. Todo ello ha permitido reducir la variabilidad de los resultados entre *folds*, lo que le da al estudio una mayor solidez en comparación con investigaciones anteriores.

Además, se documentó todo el proceso de forma detallada, lo que garantiza la reproducibilidad de los experimentos.

Finalmente, el estudio incorpora un análisis crítico y cualitativo de los resultados, utilizando visualizaciones en 3D, segmentaciones a color y análisis de errores en casos específicos. En algunos casos, como en la segmentación de instrumentos quirúrgicos, se identificó un fallo en la clasificación semántica pese a un correcto reconocimiento, lo que abre líneas de mejora futuras en la función de pérdida o en el tipo de segmentación empleada. Adicionalmente, como parte del trabajo, se elaboraron de forma propia las figuras que documentan las distintas arquitecturas presentadas en esta memoria, con el fin de proporcionar una idea clara y visual de cada una de las propuestas implementadas.

Más allá de su valor académico, este trabajo tiene el potencial de contribuir de forma significativa al entorno técnico y científico relacionado con la salud y el procesamiento de imágenes médicas. La propuesta de arquitecturas eficientes que mejoran el rendimiento sin requerir grandes recursos computacionales responde a una necesidad real en hospitales y centros de salud con infraestructuras limitadas. Esto puede facilitar que más hospitales pequeños o centros de investigación con presupuestos ajustados puedan usar herramientas de inteligencia artificial. Además, al enfocarse en que los resultados sean confiables y reproducibles, este estudio sienta una base sólida para futuros desarrollos y aplicaciones clínicas. En el plano científico, las variantes que se proponen amplían las posibilidades para experimentar con redes híbridas y pueden inspirar nuevas investigaciones en tareas similares. Por último, mejorar la precisión en la segmentación de las imágenes médicas puede ayudar a la detección y tratamiento más temprano de diversas patologías, lo cual beneficia el bienestar social y mejora la calidad de vida.

En definitiva, este trabajo ofrece una evaluación extensa, detallada y aplicada de arquitecturas híbridas U-Net y xLSTM, incorporando desarrollos propios, análisis comparativo y un enfoque orientado a la implementación práctica en contextos médicos con recursos computacionales restringidos. Todo ello se ve reforzado por un análisis crítico de los resultados y una cuidada documentación del proceso, que no solo garantiza la reproducibilidad, sino que también abre nuevas líneas de investigación y mejora.

7.3. Trabajo futuro

Como continuación del trabajo realizado, se plantea una serie de líneas de investigación orientadas a profundizar en las capacidades del modelo propuesto y ampliar su aplicabilidad clínica. En primer lugar, se propone una exploración más exhaustiva de las variantes arquitectónicas basadas en xLSTM. Esta línea de investigación contempla la evaluación sistemática del impacto de incorporar diferentes cantidades de bloques xLSTM en el codificador, así como su distribución en distintas combinaciones codificador-decodificador. El objetivo es identificar configuraciones que maximicen el rendimiento para cada modalidad de datos (2D y 3D), adaptando la arquitectura a las características particulares de cada conjunto de imágenes médicas.

Otro aspecto clave que se considera como mejora futura es la evaluación de funciones de pérdida alternativas. Aunque en este proyecto se empleó una función compuesta basada en

la entropía cruzada y el coeficiente de Dice, sería de gran interés analizar el comportamiento del modelo utilizando otras alternativas, especialmente en contextos de alto desbalanceo de clases o estructuras anatómicas complejas.

En el caso del conjunto Endovis, donde el modelo ha mostrado mayor dificultad para diferenciar herramientas quirúrgicas visualmente similares, se propone realizar un *fine-tuning* específico por clase. Este ajuste fino, orientado a refinar la discriminación semántica entre instrumentos, permitiría al modelo aprender representaciones más especializadas, con el objetivo de mejorar su precisión en entornos quirúrgicos reales.

Además, se contempla la extensión del tiempo de entrenamiento en los conjuntos de datos BraTS2023, AbdomenMRI y NeurIPS, tanto en sus modalidades 2D como 3D. En la presente investigación, los recursos computacionales limitaron el número de épocas; sin embargo, un entrenamiento más prolongado podría conducir a una convergencia más estable, especialmente en arquitecturas de alta complejidad como las evaluadas. Ampliar el ciclo de entrenamiento permitiría obtener un modelo con un comportamiento más robusto y generalizable.

Finalmente, si los modelos desarrollados alcanzan niveles de precisión adecuados para su uso en entornos clínicos, se propone como objetivo la validación externa e integración en sistemas de apoyo al diagnóstico. En este sentido, se plantea el desarrollo de una interfaz de usuario interactiva que permita al personal médico cargar imágenes y visualizar segmentaciones de forma intuitiva, con herramientas de edición y exportación. Este paso sería clave para facilitar la transferencia tecnológica desde el laboratorio hacia su implementación práctica en centros hospitalarios.

Bibliografía

- [1] Allan, M., Shvets, A., Kurmann, T., Zhang, Z., Duggal, R., Su, Y.-H., Rieke, N., Laina, I., Kalavakonda, N., Bodenstedt, S., Herrera, L., Li, W., Iglovikov, V., Luo, H., Yang, J., Stoyanov, D., Maier-Hein, L., Speidel, S., and Azizian, M. (2019). 2017 robotic instrument segmentation challenge.
- [2] Applied Computer Vision Lab, H. I. and Division of Medical Image Computing, G. C. R. C. (2025). acvl-utils: Super cool utilities that we just love to use. <https://pypi.org/project/acvl-utils/>. Version 0.2.5, License: Apache 2.0, Accessed: 2025-04-02.
- [3] Awais, M., Naseer, M., Khan, S., Anwer, R. M., Cholakkal, H., Shah, M., Yang, M.-H., and Khan, F. S. (2023). Foundational models defining a new era in vision: A survey and outlook.
- [4] Azad, R., Kazerouni, A., Heidari, M., Aghdam, E. K., Molaei, A., Jia, Y., Jose, A., Roy, R., and Merhof, D. (2023). Advances in medical image analysis with vision transformers: A comprehensive review.
- [5] Beck, M., Pöppel, K., Spanring, M., Auer, A., Prudnikova, O., Kopp, M., Klambauer, G., Brandstetter, J., and Hochreiter, S. (2024). xlstm: Extended long short-term memory.
- [6] Bradski, G. (2000). The OpenCV Library. *Dr. Dobb's Journal of Software Tools*.
- [7] Brett, M., Markiewicz, C., Hanke, M., Côté, M.-A., Cipollini, B., McCarthy, P., Cheng, C., et al. (2024). nibabel: Access a cascade of neuroimaging file formats. <https://nipy.org/nibabel/>. Accessed: 2025-04-02.
- [8] Chen, F.-L., Zhang, D.-Z., Han, M.-L., Chen, X.-Y., Shi, J., Xu, S., and Xu, B. (2023). Vlp: A survey on vision-language pre-training. *Machine Intelligence Research*, 20(1):38–56.
- [9] Chen, T. (2024). xlstm-unet-pytorch: An effective 2d & 3d medical image segmentation backbone. <https://github.com/tianrun-chen/xLSTM-UNet-Pytorch>. GitHub repository, Accessed: 2025-04-02.
- [10] Chen, T., Ding, C., Zhu, L., Xu, T., Ji, D., Wang, Y., Zang, Y., and Li, Z. (2024). xlstm-unet can be an effective 2d & 3d medical image segmentation backbone with vision-lstm (vil) better than its mamba counterpart.
- [11] Gobierno de España (2002). Ley 41/2002, de 14 de noviembre, básica reguladora de la autonomía del paciente y de derechos y obligaciones en materia de información y documentación clínica. Boletín Oficial del Estado, núm. 274, pp. 40126–40132, 15 de noviembre de 2002.
- [12] Gobierno de España (2018). Ley orgánica 3/2018, de 5 de diciembre, de protección de datos personales y garantía de los derechos digitales. Boletín Oficial del Estado, núm. 294, pp. 119788–120043, 6 de diciembre de 2018.

- [13] Gohlke, C. (2022). TiffFile: A python library to read and write image data in tiff format. *Scientific Data*, 9:874.
- [14] Gohlke, C. (2024). imagecodecs: A collection of image compression codecs for use in scientific computing. <https://github.com/cgohlke/imagecodecs>. Accessed: 2025-04-02.
- [15] Harris, C. R., Millman, K. J., van der Walt, S. J., Gommers, R., Virtanen, P., Cournapeau, D., Wieser, E., Taylor, J., Berg, S., Smith, N. J., Kern, R., Picus, M., Hoyer, S., van Kerkwijk, M. H., Brett, M., Haldane, A., del Río, J. F., Wiebe, M., Peterson, P., Gérard-Marchant, P., Sheppard, K., Reddy, T., Weckesser, W., Abbasi, H., Gohlke, C., and Oliphant, T. E. (2020). Array programming with NumPy. *Nature*, 585(7825):357–362.
- [16] Heidari, M., Aghdam, E. K., Manzella, A., Hsu, D., Scalabrino, R., Chen, W., Foran, D. J., and Hacıhaliloglu, I. (2025). A study on the performance of u-net modifications in retroperitoneal tumor segmentation.
- [17] Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, 9(8):1735–1780.
- [18] Huang, S.-C., Pareek, A., Jensen, M., Venkatesh, B., Lungren, M. P., and Ball, R. L. (2023). Self-supervised learning for medical image classification: a systematic review and implementation guidelines. *npj Digital Medicine*, 6(1):74.
- [19] Hunter, J. D. (2007). Matplotlib: A 2d graphics environment. *Computing in Science & Engineering*, 9(3):90–95.
- [20] Isensee, F., Jaeger, P. F., Petersen, J., and the DKFZ team (2024a). batchgenerators: Data augmentation in deep learning pipelines. <https://github.com/MIC-DKFZ/batchgenerators>. Accessed: 2025-04-02.
- [21] Isensee, F., Wald, T., Ziegler, S., and Cohen, E. (2024b). Dynamic network architectures: Flexible pytorch implementations of resnet, vgg and u-net for 1d, 2d and 3d data. <https://github.com/MIC-DKFZ/dynamic-network-architectures>. Developed by the Applied Computer Vision Lab (ACVL), Helmholtz Imaging. Accessed: 2025-04-02.
- [22] Jupyter, P. (2024). Jupyter notebook. <https://jupyter.org>. Accessed: 2025-04-02.
- [23] Kazerooni, A. F., Khalili, N., Liu, X., Gandhi, D., Jiang, Z., Anwar, S. M., Albrecht, J., Adewole, M., Anazodo, U., Anderson, H., Baid, U., Bergquist, T., Borja, A. J., Calabrese, E., Chung, V., Conte, G.-M., Dako, F., Eddy, J., Ezhov, I., Familiar, A., Farahani, K., Franson, A., Gottipati, A., Haldar, S., Iglesias, J. E., Janas, A., Johansen, E., Jones, B. V., Khalili, N., Kofler, F., LaBella, D., Lai, H. A., Leemput, K. V., Li, H. B., Maleki, N., McAllister, A. S., Meier, Z., Menze, B., Moawad, A. W., Nandolia, K. K., Pavaine, J., Piraud, M., Poussaint, T., Prabhu, S. P., Reitman, Z., Rudie, J. D., Sanchez-Montano, M., Shaikh, I. S., Sheth, N., Tu, W., Wang, C., Ware, J. B., Wiestler, B., Zapaishchykova, A., Bornhorst, M., Deutsch, M., Fouladi, M., Lazow, M., Mikael, L., Hummel, T., Kann, B., de Blank, P., Hoffman, L., Aboian, M., Nabavizadeh, A., Packer, R., Bakas, S., Resnick, A., Rood, B., Vossough, A., and Linguraru, M. G. (2024). The brain tumor segmentation

- in pediatrics (brats-peds) challenge: Focus on pediatrics (cbtbn-connect-dipgr-asnr-miccai brats-peds).
- [24] Kazerouni, A., Aghdam, E. K., Heidari, M., Azad, R., Fayyaz, M., Hacihaliloglu, I., and Merhof, D. (2023). Diffusion models for medical image analysis: A comprehensive survey.
 - [25] Kirillov, A., He, K., Girshick, R. B., Rother, C., and Dollár, P. (2018). Panoptic segmentation. *CoRR*, abs/1801.00868.
 - [26] Larman, C. and Basili, V. (2003). Iterative and incremental developments. a brief history. *Computer*, 36(6):47–56.
 - [27] Ma, J., Xie, R., Ayyadhury, S., Ge, C., Gupta, A., Gupta, R., Gu, S., Zhang, Y., Lee, G., Kim, J., Lou, W., Li, H., Upschulte, E., Dickscheid, T., de Almeida, J. G., Wang, Y., Han, L., Yang, X., Labagnara, M., Gligorovski, V., Scheder, M., Rahi, S. J., Kempster, C., Pollitt, A., Espinosa, L., Mignot, T., Middeke, J. M., Eckardt, J.-N., Li, W., Li, Z., Cai, X., Bai, B., Greenwald, N. F., Van Valen, D., Weisbart, E., Cimini, B. A., Cheung, T., Brück, O., Bader, G. D., and Wang, B. (2024). The multimodality cell segmentation challenge: toward universal solutions. *Nature Methods*, 21(6):1103–1113.
 - [28] Ma, J., Zhang, Y., Gu, S., Ge, C., Ma, S., Young, A., Zhu, C., Meng, K., Yang, X., Huang, Z., Zhang, F., Liu, W., Pan, Y., Huang, S., Wang, J., Sun, M., Xu, W., Jia, D., Choi, J. W., Alves, N., de Wilde, B., Koehler, G., Wu, Y., Wiesenfarth, M., Zhu, Q., Dong, G., He, J., the FLARE Challenge Consortium, and Wang, B. (2023). Unleashing the strengths of unlabeled data in pan-cancer abdominal organ quantification: the flare22 challenge.
 - [29] McKinney, W. et al. (2010). Data structures for statistical computing in python. In *Proceedings of the 9th Python in Science Conference*, volume 445, pages 51–56. Austin, TX.
 - [30] Menze, B. H., Jakab, A., Bauer, S., Kalpathy-Cramer, J., Farahani, K., Kirby, J., Burren, Y., Porz, N., Slotboom, J., Wiest, R., Lanczi, L., Gerstner, E., Weber, M.-A., Arbel, T., Avants, B. B., Ayache, N., Buendia, P., Collins, D. L., Cordier, N., Corso, J. J., Criminisi, A., Das, T., Delingette, H., Demiralp, , Durst, C. R., Dojat, M., Doyle, S., Festa, J., Forbes, F., Geremia, E., Glocker, B., Golland, P., Guo, X., Hamamci, A., Iftexharuddin, K. M., Jena, R., John, N. M., Konukoglu, E., Lashkari, D., Mariz, J. A., Meier, R., Pereira, S., Precup, D., Price, S. J., Raviv, T. R., Reza, S. M. S., Ryan, M., Sarikaya, D., Schwartz, L., Shin, H.-C., Shotton, J., Silva, C. A., Sousa, N., Subbanna, N. K., Szekely, G., Taylor, T. J., Thomas, O. M., Tustison, N. J., Unal, G., Vasseur, F., Wintermark, M., Ye, D. H., Zhao, L., Zhao, B., Zikic, D., Prastawa, M., Reyes, M., and Van Leemput, K. (2015). The multimodal brain tumor image segmentation benchmark (brats). *IEEE Transactions on Medical Imaging*, 34(10):1993–2024.
 - [31] Microsoft Corporation (2025). Visual studio code. <https://code.visualstudio.com/>. Version 1.89.0, Accessed: 2025-04-02.

- [32] Minaee, S., Boykov, Y., Porikli, F., Plaza, A., Kehtarnavaz, N., and Terzopoulos, D. (2020). Image segmentation using deep learning: A survey.
- [33] MONAI, P. (2024). Monai: Medical open network for ai. <https://monai.io>. Version 1.3.0, Accessed: 2025-04-02.
- [34] Naciones Unidas (2018). ¿sabes cuáles son los 17 objetivos de desarrollo sostenible? <https://www.un.org/sustainabledevelopment/es/2018/08/sabes-cuales-son-los-17-objetivos-de-desarrollo-sostenible/>. Accessed: 2025-04-03.
- [35] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., et al. (2011). Scikit-learn: Machine learning in python. *Journal of machine learning research*, 12(Oct):2825–2830.
- [36] Ronneberger, O., Fischer, P., and Brox, T. (2015). U-net: Convolutional networks for biomedical image segmentation. *CoRR*, abs/1505.04597.
- [37] Saraei, M., Lalinia, M., and Lee, E.-J. (2025). Deep learning-based medical object detection: A survey. *IEEE Access*, 13:53019–53038.
- [38] Team, J. D. (2024a). ipywidgets: Interactive html widgets for jupyter notebooks and the ipython kernel. <https://ipywidgets.readthedocs.io/>. Version 8.1.2, Accessed: 2025-04-02.
- [39] Team, P. (2024b). Pytorch: Deep learning framework. <https://pytorch.org>. Version 2.0.0, Accessed: 2025-04-02.
- [40] Team, S. D. (2024c). Simpleitk: Simplified image processing toolkit. <https://simpleitk.org/>. Accessed: 2025-04-02.
- [41] Unión Europea (2016). Reglamento (ue) 2016/679 del parlamento europeo y del consejo, de 27 de abril de 2016, relativo a la protección de las personas físicas en lo que respecta al tratamiento de datos personales y a la libre circulación de estos datos. Diario Oficial de la Unión Europea, L 119, pp. 1–88, 4 de mayo de 2016.
- [42] van der Walt, S., Schönberger, J. L., Nunez-Iglesias, J., Boulogne, F., Warner, J. D., Yager, N., Gouillart, E., Yu, T., and the scikit-image contributors (2014). scikit-image: Image processing in python. *PeerJ*, 2:e453.
- [43] van Rikxoort, R. R. (2024). dicom2nifti: Convert dicom files to nifti format. <https://github.com/icometrix/dicom2nifti>. GitHub repository, Accessed: 2025-04-02.
- [44] Virtanen, P., Gommers, R., Oliphant, T. E., Haberland, M., Reddy, T., Cournapeau, D., Burovski, E., Peterson, P., Weckesser, W., Bright, J., van der Walt, S. J., Brett, M., Wilson, J., Millman, K. J., Mayorov, N., Nelson, A. R. J., Jones, E., Kern, R., Larson, E., Carey, C. J., Polat, İ., Feng, Y., Moore, E. W., VanderPlas, J., Laxalde, D., Perktold, J., Cimrman, R., Henriksen, I., Quintero, E. A., Harris, C. R., Archibald, A. M., Ribeiro, A. H., Pedregosa, F., van Mulbregt, P., and SciPy 1.0 Contributors (2020). SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*, 17:261–272.

- [45] Walczak, E. and contributors (2024). graphviz: Simple python interface for graphviz. <https://github.com/xflr6/graphviz>. Version 0.20.3, Accessed: 2025-04-02.
- [46] Wasim, M. (2013). A review of image segmentation techniques. *International Journal of Future Computer and Communication*, 2:89–93.
- [47] Waskom, M. L. (2021). seaborn: statistical data visualization. *Journal of Open Source Software*, 6(60):3021.
- [48] Weights and Team, B. (2025). wandb: Experiment tracking, model management, and collaboration for machine learning projects. <https://wandb.ai>. Version 0.16.4, Accessed: 2025-04-02.
- [49] Zhou, T., Xia, W., Zhang, F., Chang, B., Wang, W., Yuan, Y., Konukoglu, E., and Cremers, D. (2024). Image segmentation in foundation model era: A survey.