UNIVERSIDAD POLITÉCNICA DE MADRID

E.T.S. DE INGENIERÍA DE SISTEMAS INFORMÁTICOS

PROYECTO FIN DE GRADO

GRADO EN INGENIERÍA DEL SOFTWARE

Segmentación Semántica en Cirugía Laparoscópica con Optimización para Clases Minoritarias

Desarrollado por: Leticia Martínez García

Dirigido por: Alberto Díaz Álvarez y Félix José Fuentes Hurtado

Madrid, 25 de junio de 2025

Segmentación Semántica en Cirugía Laparoscópica con Optimización para Clases Minoritarias
Desarrollado por: Leticia Martínez García
Dirigido por: Alberto Díaz Álvarez y Félix José Fuentes Hurtado
Proyecto Fin de Grado, 25 de junio de 2025

E.T.S. de Ingeniería de Sistemas Informáticos
Campus Sur UPM, Carretera de Valencia (A-3), km. 7
28031, Madrid, España

Si deseas citar este trabajo, la entrada completa en BibTeX es la siguiente:

```
@mastersthesis{citekey,
    title = {Segmentación Semántica en Cirugía Laparoscópica con Optimización
para Clases Minoritarias},
    type = {Bachelor's Thesis},
    author = {Martínez-García, L.},
    school = {E.T.S. de Ingeniería de Sistemas Informáticos},
    year = {2025},
    month = {6},
}
```

Digo que lo sé, y no que lo concibo, ni que lo comprendo, pues se puede saber que Dios es infinito y omnipotente, aunque nuestra alma, siendo finita, no lo pueda comprender ni concebir –de la misma manera que podemos tocar con las manos una montaña, pero no abrazarla como haríamos con un árbol o con cualquier otra cosa que no excediese el tamaño de nuestros brazos–. Pues comprender es abrazar con el pensamiento, pero para saber una cosa es suficiente con tocarla con el pensamiento.

— René Descartes

Resumen

Este trabajo se centra en la aplicación de técnicas de aprendizaje profundo, en concreto, el desarrollo de una red neuronal convolucional para su uso en segmentación semántica de imágenes laparoscópicas de colecistectomía.

El modelo ha sido entrenado a partir de un conjunto de imágenes reales de cirugías, incorporando técnicas de aumento de datos y estrategias específicas para optimizar la segmentación de clases minoritarias, con el fin de mejorar su capacidad de generalización frente a distintos contextos quirúrgicos y condiciones visuales.

El propósito principal del modelo es identificar y delimitar tanto los distintos elementos anatómicos como las herramientas quirúrgicas más comunes presentes en el campo operatorio, con la esperanza de que el modelo pueda ser útil en entornos clínicos, asistiendo al cirujano en tiempo real, de forma que contribuya a intervenciones más precisas y seguras. Además, puede servir como apoyo durante la formación médica de los estudiantes.

Finalmente, se analizan los resultados obtenidos por el algoritmo, evaluando su capacidad para segmentar con precisión las diferentes regiones de interés.

Palabras clave: aprendizaje profundo; segmentación semántica; imágenes laparoscópicas; red neuronal convolucional

Abstract

This work focuses on the application of deep learning techniques, specifically the development of a convolutional neural network (CNN) for semantic segmentation of laparoscopic images from cholecystectomy procedures.

The model has been trained using a dataset of real surgical images, incorporating data augmentation techniques and targeted strategies to improve the segmentation of minority classes, with the aim of enhancing its generalization capabilities across different surgical contexts and visual conditions.

The main goal of the model is to identify and delineate both anatomical structures and common surgical tools present in the operative field, with the expectation that it could prove useful in clinical settings by assisting the surgeon in real time, thus contributing to more precise and safer interventions. Additionally, it may serve as a valuable support tool in medical training for students.

Finally, the results obtained by the algorithm are analyzed, assessing its ability to accurately segment the different regions of interest.

Keywords: deep learning; semantic segmentation; laparoscopic images; convolutional neural network

Índice general

Índice de figuras

Índice de tablas

1. Introducción

La medicina ha sido una disciplina que se ha caracterizado por el uso necesario de herramientas; entre ellas, se pueden encontrar: el bisturí, el colonoscopio, el estetoscopio, etc.

La historia de la medicina ha estado acompañada hasta hace poco por herramientas metálicas, rudimentarias, no tan distintas de un hacha o una hoz.

En estos últimos años, la ingeniería ha desarrollado una herramienta novedosa que se aleja de la tangibilidad de los utensilios habituales, llamada inteligencia artificial (IA). La IA es una herramienta de supercálculo, almacenamiento y gestión de datos. Esa naturaleza ha propiciado un desarrollo explosivo en un gran conjunto de áreas heterogéneas, como, por ejemplo, la agricultura inteligente, la industria, entre otros.

Este desarrollo ha sido especialmente exponencial y fructífero en todas las ciencias y ramas de investigación. Y, habiendo estado ligada la medicina a la técnica y los útiles, era prácticamente imposible que no se hubiera visto afectada por el desarrollo de la IA.

En este contexto, la colecistectomía laparoscópica (término definido más abajo) es el objeto de estudio de la red neuronal convolucional (CNN) desarrollada en este proyecto de fin de grado (PFG).

Para comprender que es la colecistectomía laparoscópica, se debe definir la palabra laparoscopia: una técnica quirúrgica que se practica a través de pequeñas incisiones por las que se introduce una cámara de video que permite al equipo médico ver el campo quirúrgico dentro del paciente [1].

La laparoscopia ha revolucionado la cirugía al ofrecer múltiples ventajas sobre la cirugía abierta tradicional. Al ser una técnica mínimamente invasiva, entre sus beneficios se incluyen: una recuperación postoperatoria más rápida, menor dolor, reducción de complicaciones como infecciones y adherencias, y mejores resultados estéticos [2].

Además, la evolución de la laparoscopia ha permitido ampliar su aplicación a una variedad creciente de procedimientos quirúrgicos, incluyendo la colecistectomía.

La colecistectomía es un procedimiento quirúrgico que consiste en la extirpación de la vesícula biliar. Realizada por primera vez en 1985 e introducida en la práctica clínica general en 1990, se ha convertido rápidamente en el procedimiento predominante para la cirugía de vesícula biliar [3].

Este tipo de cirugías, aunque mínimamente invasivas y con notables ventajas para el paciente, plantea desafíos significativos para los cirujanos debido a: las condiciones variables de visibilidad, la visión túnel, la maniobrabilidad reducida en espacios estrechos y la falta de retroalimentación táctil [4]. La integración de la IA representa una herramienta útil para paliar parte de estos problemas.

1.1. Objetivos

El objetivo principal de este PFG es: **desarrollar una CNN para segmentación laparoscópica** (ver Figura 1.1)**, que cumpla con las siguientes características: incorporar técnicas de aumento de datos y funciones de pérdida adaptativas, con el fin de mejorar la generalización del modelo y su precisión en la segmentación de estructuras anatómicas minoritarias.**

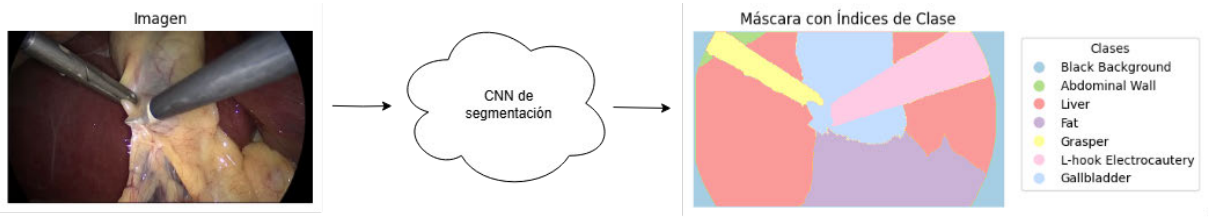

Figura 1.1. Resultado deseado del modelo de segmentación

Con este propósito, se han establecido los siguientes objetivos específicos:

- Investigar una arquitectura de CNN adecuada para la segmentación semántica de imágenes laparoscópicas.
- Preprocesar y adaptar el *dataset* CholecSeg8k, incluyendo el análisis de clases minoritarias y su distribución.
- Implementar técnicas de aumento de datos para mejorar la fuerza del modelo frente a la variabilidad intraoperatoria.

- Integrar funciones de pérdida especializadas para mejorar la segmentación de estructuras pequeñas y poco representadas.
- Evaluar el rendimiento del modelo utilizando métricas relevantes.
- Analizar los resultados obtenidos y discutir su posible aplicación en contextos clínicos o de formación médica.

1.2. Motivación

Desde el comienzo de mis estudios en software he deseado que los proyectos en los que participo tengan un propósito real, es decir, que puedan ser útiles y aporten valor a la sociedad. Este PFG representa una oportunidad para aplicar mis conocimientos en un ámbito de gran relevancia, como es la medicina.

En particular, mi mayor motivación personal ha sido el deseo de aportar, aunque sea un pequeño granito de arena, para que los pacientes puedan enfrentarse a una operación con mayor tranquilidad, sabiendo que existen herramientas que asisten al profesional y aumentan la seguridad del procedimiento.

2. Estado de la cuestión

2.1. Inteligencia artificial

Actualmente, no existe una única definición de IA, dependiendo del contexto social, esto puede abarcar desde definiciones epistemológicas hasta sacadas de la ciencia ficción. Por lo tanto, si se pretende comprender con rigor el objeto de estudio, es necesario ir a la raíz del concepto y formularse preguntas fundamentales: ¿cómo se construye un sistema de IA?, ¿de qué se compone?, ¿cómo funciona realmente?

Desde una perspectiva técnica, la IA puede definirse de manera sencilla a través de su objetivo: «la capacidad de los ordenadores para realizar tareas que normalmente requieren inteligencia humana» [5]. Aunque esta definición pueda parecer ambigua —pues remite a un concepto de inteligencia humana que tampoco está completamente definido—, permite entender que la IA se orienta hacia la resolución de problemas mediante la imitación de ciertos procesos cognitivos.

Esta «imitación» se debe al progreso en el campo de la neurociencia, que ha influido significativamente en el diseño y la evolución de los modelos de IA. En términos más concretos, se puede considerar que la IA permite a un sistema «aprender» a resolver problemas, desde tareas simples como clasificar correos electrónicos hasta otras más complejas, como la conducción autónoma. Esta capacidad se logra, sobre todo, mediante la estadística, el álgebra lineal y la optimización, que permiten modelar patrones a partir de grandes volúmenes de datos [6].

Dentro de los campos de aplicación de la IA, la visión artificial (CV) tiene como objetivo que los ordenadores «vean» imágenes y vídeos como lo haría un humano. En particular, la CV aplicada mediante redes neuronales profundas ha supuesto un gran avance en tareas como la segmentación de imágenes médicas.

Dentro del conjunto que representa la IA existen distintas ramas, cada una con enfoques específicos según el tipo de problema que se desea resolver:

El aprendizaje automático (ML) constituye una de las ramas principales de la IA. Su enfoque se centra en el desarrollo de algoritmos y modelos capaces de aprender y mejorar su desempeño mediante la experiencia, a partir del análisis de grandes volúmenes de datos. A diferencia de la programación tradicional, donde el comportamiento del sistema se define explícitamente mediante instrucciones, en ML se entrena a la máquina proporcionándole datos de los cuales extrae patrones, permitiéndole realizar predicciones o tomar decisiones de forma autónoma [7].

Dentro de esta rama se encuentra el aprendizaje profundo (DL), un subcampo especializado dentro del ML que se caracteriza por el uso de red neuronal artificial (ANN) de múltiples capas. Su objetivo es emular, de forma simplificada, el funcionamiento del cerebro humano en el procesamiento de datos complejos, como imágenes, texto o señales de audio. A través de estas arquitecturas profundas, el DL es capaz de aprender representaciones jerárquicas de los datos de manera automática, identificando patrones y relaciones con un nivel de abstracción superior al de otros métodos tradicionales [7].

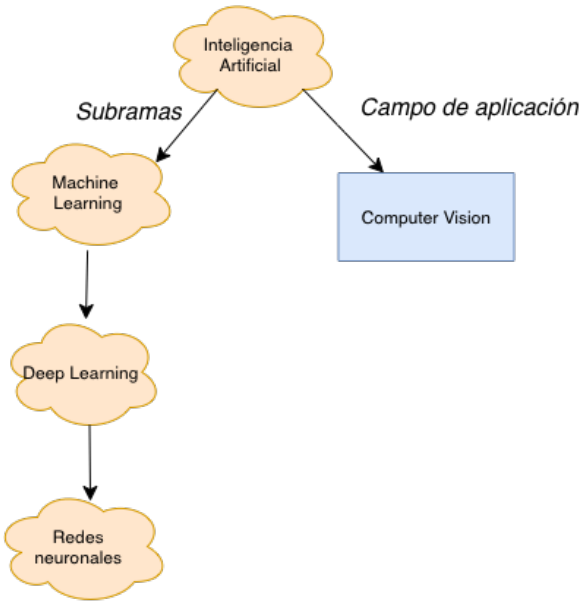

Figura 2.1. Representación estructurada de las ramas de la IA relevantes para este estudio

Las CNNs son una arquitectura específica dentro del campo del DL. Estas redes hacen uso de la operación de convolución aplicando filtros, esta operación se utiliza específicamente para procesar datos con estructura espacial o temporal, como imágenes o clips de sonido, ya que mantiene la topología de los datos (por ejemplo, filas y columnas en

imágenes) y aprovecha esa estructura.

La convolución discreta es una transformación lineal que preserva el orden de los datos, es esparcida (sólo unas pocas unidades de entrada afectan a una de salida) y reutiliza parámetros (los mismos pesos se aplican en múltiples ubicaciones del *input*) [8].

Gracias a su capacidad para detectar patrones locales —como bordes, texturas o formas— de forma jerárquica a lo largo de sus capas, las CNNs se emplean ampliamente en tareas de CV, incluyendo la segmentación semántica de imágenes médicas [9].

2.2. Visión por computador aplicada a la cirugía laparoscópica

La integración de técnicas de CV en la cirugía laparoscópica está permitiendo avances significativos en diversas áreas, mejorando tanto la precisión como la seguridad de los procedimientos quirúrgicos. Entre sus principales aplicaciones se encuentran: la detección de fases quirúrgicas, el reconocimiento de acciones, la segmentación de estructuras anatómicas e instrumentos quirúrgicos, así como la predicción de la visión crítica de seguridad (CVS), que evalúa si el campo quirúrgico cumple los criterios visuales mínimos para continuar la operación con seguridad.

En el ámbito de la CV aplicada a la cirugía, la colecistectomía laparoscópica se ha consolidado como un procedimiento de referencia para el entrenamiento y validación de modelos de segmentación [4].

Sin embargo, uno de los principales retos radica en que muchos de los algoritmos actuales no son adecuados para su aplicación directa en entornos quirúrgicos reales, ya que han sido entrenados con conjuntos de datos limitados que presentan condiciones homogéneas de iluminación, calidad de imagen y variabilidad anatómica [10].

Con el objetivo de mejorar la generalización de los modelos a nuevas situaciones clínicas, en los últimos años se ha impulsado la creación de conjuntos de datos más amplios. Estos nuevos *datasets*, formados por grandes volúmenes de imágenes extraídas de vídeos quirúrgicos en tiempo real, permiten entrenar modelos más adaptables a diferentes contextos clínicos. Algunos ejemplos destacados son `CholecTrack20` [11] y `Endoscapes2023` [12].

Paralelamente, también se están explorando métodos para aumentar la eficiencia de los modelos, puesto que el entorno quirúrgico es especialmente exigente en términos de fiabilidad, velocidad y precisión. Esto ha favorecido la adopción de arquitecturas más rápidas como YOLOv8 (*You Only Look Once*), un algoritmo ampliamente reconocido que utiliza un enfoque de detección *single-shot*, procesando cada imagen en una sola pasada. A diferencia de otros métodos que requieren múltiples análisis por imagen, YOLOv8 logra un equilibrio entre velocidad y eficiencia, aunque sus primeras versiones eran menos precisas en comparación con otros enfoques más complejos [10].

Asimismo, dado el alto coste temporal y humano que supone generar y anotar manualmente estos conjuntos de datos, se han propuesto nuevas estrategias para reducir la dependencia de datos etiquetados. Una de las más prometedoras es el aprendizaje semisupervisado, como en el caso del uso de *Mask Denoising Autoencoders*, aplicados recientemente en imágenes laparoscópicas con resultados destacados [13].

En los últimos años, los modelos basados en Transformers han comenzado a ganar protagonismo en la CV, incluyendo su aplicación al análisis de imágenes laparoscópicas. Su capacidad para capturar relaciones a largo alcance entre píxeles ha permitido mejorar el rendimiento en tareas complejas de segmentación anatómica. Sin embargo, su adopción práctica en el ámbito clínico sigue limitada debido a sus elevados requisitos de datos y computación, factores especialmente críticos en contextos quirúrgicos donde los conjuntos de datos anotados pueden ser escasos [14].

Los mencionados modelos híbridos combinan las capacidades de las CNNs y los Transformers para abordar las limitaciones inherentes a cada enfoque en tareas de segmentación médica. Estos modelos buscan aprovechar la capacidad de las CNNs para capturar detalles locales y la habilidad de los Transformers para modelar relaciones globales en los datos.

Uno de los modelos pioneros en esta línea es TransUNet, que integra un codificador basado en Transformers dentro de la arquitectura U-Net. En TransUNet, las características extraídas por una CNN se *tokenizan* y se procesan mediante capas Transformer para capturar contextos globales, mientras que el decodificador de tipo U-Net permite una reconstrucción precisa de los detalles espaciales [15].

Por otro lado, Swin-Unet propone una arquitectura completamente basada en Transformers, utilizando bloques Swin Transformer en lugar de convoluciones tradicionales. Esta arquitectura mantiene la estructura en forma de «U» y las conexiones de salto características de U-Net, permitiendo una fusión efectiva de características locales y glo-

bales. Swin-Unet ha demostrado un rendimiento superior en tareas de segmentación de múltiples órganos y cardíaca [16].

2.3. Arquitecturas de segmentación

2.3.1. Arquitecturas *encoder-decoder* para segmentación médica

Entre las distintas arquitecturas de red neuronal artificial, los modelos *encoder-decoder* destacan en segmentación médica [17], particularmente por su capacidad para equilibrar dos aspectos:

- El **contexto global**, que comprende la extracción de características semánticas profundas.
- El **detalle local**, que preserva información espacial de alta resolución [17].

2.3.2. U-Net clásica

La U-Net, introducida por Ronneberger [18], tuvo un gran impacto en la segmentación biomédica mediante una arquitectura simétrica en forma de «U» que combina la extracción de características contextuales con la preservación de detalles espaciales de la imagen. Su diseño aborda dos retos en imágenes médicas:

1. La **variabilidad morfológica** de estructuras anatómicas.
2. La **necesidad de precisión sub-píxel**, es decir, una precisión mayor que el tamaño del píxel individual.

Sus componentes son los siguientes:

Encoder

- Bloques convolucionales (3×3, ReLU) seguidos de operaciones de *max pooling* (2×2) para reducir la resolución espacial.

- Suele constar de 4 a 5 niveles, reduciendo progresivamente la imagen de entrada (por ejemplo: $512 \times 512 \rightarrow 32 \times 32$).
- Este módulo extrae características jerárquicas, desde bordes y texturas hasta estructuras más abstractas, condensando la información relevante para etapas posteriores de la red.
- Además, para reducir el riesgo de sobreajuste, algunas implementaciones de U-Net utilizan *dropout*, especialmente en las capas más profundas del encoder.

Cuello de botella

- Se sitúa entre el encoder y el decoder y representa el punto de mayor compresión de la red.
- Concentra la información extraída por el encoder en una representación compacta, que servirá para la reconstrucción posterior.

Decoder

- Emplea convoluciones transpuestas (2×2) para aumentar progresivamente la resolución espacial (por ejemplo: $32 \times 32 \rightarrow 512 \times 512$).
- En cada etapa se añade información del encoder mediante conexiones de salto, lo que ayuda a recuperar detalles espaciales finos perdidos durante la compresión.
- Este módulo reconstruye una representación de segmentación, combinando el contexto global aprendido con la información local preservada.

Conexiones de salto

- Enlazan capas simétricas del encoder y decoder (por ejemplo: nivel i con nivel $n-i$), permitiendo la recuperación de detalles espaciales.
- Transfieren directamente los mapas de características generados por el encoder, sin aplicar modificaciones, para combinarlos con los del decoder.
- Son esenciales para preservar bordes, texturas y otras características de bajo nivel que se pierden en la compresión del cuello de botella.

2.3.3. *Attention U-Net*: Evolución orientada a objetivos quirúrgicos

Mecanismo de Atención

La estructura general de *attention U-Net* mantiene el esquema encoder-decoder proveniente de U-Net, incluyendo las conexiones de salto. Sin embargo, introduce una modificación: antes de concatenar las activaciones del encoder al decoder, estas pasan por un módulo de atención denominado *Attention Gate* [19].

Este módulo recibe dos entradas:

- La activación generada en una capa del encoder, que conserva información local de bajo nivel.
- La activación correspondiente del decoder, que aporta información semántica de mayor abstracción.

La máscara de atención α_l se calcula mediante la siguiente operación:

$$\alpha_l = \sigma\left(\psi^T\left(\text{ReLU}(W_x x_l + W_g g + b_g)\right) + b_\psi\right) \tag{2.1}$$

donde x_l representa la activación del encoder en el nivel l, g es la señal del decoder, y W_x, W_g, b_g son parámetros aprendibles.

Efectos del mecanismo de atención:

- **Enfoque selectivo**:
  - Elimina regiones del fondo que no aportan información relevante, como grasa abdominal o artefactos como el humo quirúrgico.
  - Mejora la visibilidad de estructuras críticas, como nervios o márgenes tumorales.

- **Eficiencia computacional**:
  - Añade un sobrecoste reducido (∼15 % más parámetros que la U-Net clásica) [19].
  - Disminuye los falsos positivos en un ∼20–30 % [20].

2.4. Afrontando los retos en segmentación médica

2.4.1. Aumento de datos

La segmentación semántica de imágenes médicas presenta desafíos propios como: la escasez de datos médicos anotados suele deberse a los elevados costos asociados a la anotación manual por parte de especialistas, a los requisitos éticos y a la gran heterogeneidad presente en las imágenes obtenidas. Esta heterogeneidad se debe a:

- Variabilidad anatómica entre pacientes.
- Condiciones intraoperatorias (sangrado, vaporización, etc.).
- Cambios en la iluminación laparoscópica.
- Diferencias entre dispositivos de adquisición [21].

El aumento de datos es una estrategia desarrollada para mitigar estos problemas. A través de transformaciones controladas se generan imágenes sintéticas que:

- Amplían el volumen de datos disponible.
- Introducen mayor diversidad morfológica y contextual.
- Disminuyen el riesgo de sobreajuste.

Esto permite mejorar el desempeño del modelo sin comprometer la coherencia anatómica [22], [23]. Para ello, se busca evitar transformaciones agresivas como alteraciones cromáticas extremas o rotaciones anatómicamente imposibles, que puedan degradar la relevancia clínica de los datos [22].

Técnicas aplicables en cirugía laparoscópica

- **Transformaciones geométricas básicas**:
  - Rotaciones ($\pm 15°$) y volteos (horizontal/vertical $\pm 90°$), que simulan cambios en la perspectiva del laparoscopio.
  - Traslaciones y escalados ($\pm 10\,\%$–$20\,\%$), para emular variaciones en la distancia focal.
- **Ajustes fotométricos**:
  - Modulación de brillo/contraste ($\pm 15\,\%$), replicando cambios en la iluminación.
  - Adición de ruido gaussiano ($\sigma \leq 0{,}05$), para simular artefactos de sensor.
- **Técnicas avanzadas**:
  - Recorte aleatorio (recortes de 256×256 píxeles), que fomentan el aprendizaje de características locales.
  - Deformaciones elásticas ($\alpha \leq 50$, $\sigma \leq 5$), empleadas para simular la deformabilidad de los tejidos.

En laparoscopia, se recomienda priorizar:

- **Conservación topológica**: Las transformaciones deben respetar la estructura anatómica (por ejemplo: no invertir órganos asimétricos).
- **Realismo fotométrico**: Los ajustes de color deben mantenerse dentro de rangos fisiológicos (por ejemplo: tonos de sangre y tejidos).

2.4.2. Adaptación dinámica de la tasa de aprendizaje

En el entrenamiento de redes neuronales profundas, la tasa de aprendizaje controla la velocidad con la que el modelo actualiza sus pesos. Si es demasiado alta, puede impedir la convergencia del modelo; si es demasiado baja, puede ralentizar el aprendizaje y hacer que quede atrapado en mínimos locales subóptimos [24].

Para mejorar la eficiencia del entrenamiento, existen esquemas que ajustan dinámicamente este parámetro:

- **CosineAnnealingLR**: Propuesto por Loshchilov y Hutter [25], este planificador ajusta la tasa de aprendizaje siguiendo una curva coseno decreciente a lo largo del entrenamiento, desde un valor máximo hasta un mínimo definido. Favorece una convergencia estable, comenzando con una fase de mayor exploración y terminando con ajustes más finos.
- **ReduceLROnPlateau**: Este planificador reactivo reduce automáticamente la tasa de aprendizaje cuando la pérdida de validación deja de mejorar durante un número determinado de épocas, lo que permite superar estancamientos en la optimización [26].

2.4.3. Mejora del rendimiento: Funciones de pérdida adaptadas a clases minoritarias

La función de pérdida cuantifica el error entre las predicciones del modelo y las etiquetas reales. Su objetivo es proporcionar una medida que el optimizador pueda minimizar para mejorar la precisión del modelo.

En segmentación médica se puede apreciar cierto desbalance de clases: algunas estructuras anatómicas de interés ocupan áreas muy reducidas en comparación con el fondo o con otras regiones más extensas. Un ejemplo pueden ser los tumores, que en ocasiones representan menos del 1 % del área total de la imagen. Esta desproporción puede hacer que el modelo tienda a ignorar las clases minoritarias durante el entrenamiento, obteniendo segmentaciones imprecisas en las estructuras más críticas.

Para mitigar este problema, se han propuesto funciones de pérdida que asignan mayor peso a las clases minoritarias o penalizan más los errores sobre ellas [27].

En este contexto, destacan especialmente:

- ***Focal Loss***: Propuesta por Lin et al. [28], aumenta el peso de los ejemplos difíciles mediante un término de modulación.

$$\text{FL}(p_t) = -\alpha_t(1-p_t)^\gamma \log(p_t) \tag{2.2}$$

  donde p_t es la probabilidad asignada a la clase correcta, α_t un peso para compensar la frecuencia de clases, y γ un parámetro que regula cuánto se penalizan los errores.

- ***Dice loss***: Basada en el coeficiente de Dice, mide directamente la superposición entre predicción y máscara real. [29].

$$\text{Dice Loss} = 1 - \frac{2\sum_i p_i g_i}{\sum_i p_i + \sum_i g_i} \tag{2.3}$$

  donde p_i y g_i son, respectivamente, la predicción y la etiqueta verdadera del píxel i.

- ***Tversky loss***: Generaliza la Dice Loss introduciendo dos parámetros de ponderación que permiten controlar el equilibrio entre falsos positivos y falsos negativos. Es especialmente útil cuando los errores tienen distinto impacto clínico [30].

$$\text{Tversky index} = \frac{\sum_i p_i g_i}{\sum_i p_i g_i + \alpha \sum_i p_i(1-g_i) + \beta \sum_i (1-p_i) g_i} \tag{2.4}$$

$$\text{Tversky loss} = 1 - \text{Tversky Index} \tag{2.5}$$

2.4.4. Normalización por lotes

La técnica de normalización por lotes (BN) fue introducida por Ioffe y Szegedy [31] para abordar el problema del cambio en la distribución de las entradas durante el entrenamiento de redes neuronales profundas, conocido como *Covariate Shift*. Este fenómeno ocurre cuando la distribución de las activaciones de una capa cambia significativamente debido a las actualizaciones de los parámetros en capas anteriores, lo que ralentiza la convergencia al requerir tasas de aprendizaje pequeñas y una cuidadosa inicialización de los pesos.

BN normaliza las activaciones de cada capa para que sigan una distribución estable (media cercana a 0 y desviación estándar cercana a 1) en cada lote durante el entrenamiento.

Esto se logra mediante:

$$\hat{x}_i = \frac{x_i - \mu_{\mathcal{B}}}{\sqrt{\sigma_{\mathcal{B}}^2 + \epsilon}} \tag{2.6}$$

donde:

- $\hat{x}_i$ es el valor normalizado,
- $\mu_{\mathcal{B}}$ y $\sigma_{\mathcal{B}}^2$ son la media y varianza del lote,
- ϵ es una constante para evitar división por cero.

Posteriormente, se aplican parámetros aprendibles de escala (γ) y desplazamiento (β) para preservar la capacidad representativa de la red:

$$y_i = \gamma \hat{x}_i + \beta \tag{2.7}$$

Entre las principales ventajas del uso de BN se encuentran:

- **Reducción del *Covariate Shift***: Al estabilizar la distribución de las activaciones intermedias, BN mitiga el problema del *Covariate Shift* interno, acelerando significativamente la convergencia del entrenamiento [31].
- **Mayor eficiencia en la optimización**: BN permite el uso de tasas de aprendizaje más elevadas, reduce la sensibilidad a la inicialización de los pesos y facilita un entrenamiento más estable incluso en redes profundas [32].
- **Regularización implícita**: El proceso de normalización por lotes introduce una forma de ruido estocástico durante el entrenamiento, lo que actúa como un regularizador suave y ayuda a reducir el sobreajuste sin necesidad de técnicas adicionales. [33].

En el contexto de segmentación laparoscópica, donde, como se ha mencionado anteriormente, existen importantes variaciones en iluminación, texturas y condiciones intraoperatorias, la inclusión de BN contribuye a mejorar la capacidad de generalización del modelo frente a imágenes de distinta procedencia [34].

2.5. Métricas de evaluación

2.5.1. Matriz de confusión

La matriz de confusión consiste en una tabla cuadrada donde las filas representan las clases reales y las columnas las clases predichas por el modelo.

La diagonal principal contiene los píxeles correctamente clasificados (verdaderos positivos para cada clase), mientras que las celdas fuera de la diagonal muestran las confusiones del modelo, píxeles asignados erróneamente a otras clases. [35].

A partir de esta matriz se calculan métricas que permiten evaluar el desempeño del modelo para cada clase, tales como:

- **Sensibilidad:** Mide el porcentaje de píxeles correctamente clasificados de una clase en relación con todos los píxeles reales de esa clase. Un valor menor al 100 % indica que el modelo ha ignorado parte de esa clase al segmentar.
- **Tasa de falsos positivos (FPR):** Mide el porcentaje de píxeles que fueron asignados erróneamente a una clase, respecto al total de píxeles que no pertenecían a ella. Un FPR alto significa que el modelo está «viendo» esa clase donde no está, lo que puede deberse a similitudes visuales con otras estructuras o a predicciones demasiado optimistas.
- **Tasa de falsos negativos (FNR):** Es el complemento de la sensibilidad, e indica el porcentaje de píxeles de una clase que el modelo no logró identificar. Un FNR alto implica que el modelo está omitiendo estructuras importantes.

$$\text{FNR} = 1 - \text{Sensibilidad} = \frac{\text{FN}}{\text{FN} + \text{TP}}$$

donde FN son los falsos negativos y TP los verdaderos positivos.

2.5.2. Índice de Jaccard (IoU)

El IoU, también conocido como índice de Jaccard, es una métrica que mide la superposición entre la región predicha por el modelo y la región real [36].

Se define como la razón entre la intersección y la unión de las áreas predicha y real:

$$\text{IoU} = \frac{|P \cap G|}{|P \cup G|} \tag{2.8}$$

donde P es el conjunto de píxeles predichos como pertenecientes a una clase, y G es el conjunto de píxeles reales de dicha clase.

Un valor de IoU cercano a 1 indica una segmentación precisa, mientras que valores bajos indican discrepancias significativas entre la predicción y el *ground truth*.

2.5.3. Precisión por píxel

La precisión por píxel es una métrica que mide la proporción de píxeles correctamente clasificados respecto al total de píxeles en la imagen.

Matemáticamente, se expresa como:

$$\text{Precisión por píxel} = \frac{TP + TN}{TP + TN + FP + FN} \tag{2.9}$$

donde TP son los verdaderos positivos, TN los verdaderos negativos, FP los falsos positivos y FN los falsos negativos [37].

Se debe mencionar que esta métrica favorece más a las clases mayoritarias que a las minoritarias, ya que al solo medir la cantidad de pixeles, una clase que ocupe pocos pixeles en la imagen puede ser pasada por alto.

3. Metodología

3.1. Selección y preparación del *dataset*

3.1.1. Selección del *dataset*

Se ha elegido el *dataset* CholecSeg8k [38] por varias razones:

- **Representatividad clínica**: Las imágenes provienen de intervenciones reales de colecistectomía laparoscópica. Es decir, no son simples imitaciones o dibujos, reflejan la realidad del procedimiento.
- **Anotaciones de alta calidad**: Cada imagen cuenta con una máscara de segmentación detallada, donde se delimitan las estructuras anatómicas de interés y las herramientas quirúrgicas.
- **Volumen de datos**: El *dataset* contiene un número considerable de muestras (más de 8.000 imágenes), lo que facilita el entrenamiento del modelo minimizando el riesgo de sobreajuste.
- **Accesibilidad y uso académico**: Está disponible públicamente bajo licencia para investigación en la plataforma Kaggle.

Aunque en el estado de la cuestión se han mencionado conjuntos de datos más recientes y extensos, como `Endoscapes2023` o `CholecTrack20`, su uso planteaba ciertas limitaciones prácticas para un proyecto individual como este PFG. Muchos de ellos requieren solicitudes de acceso que restringen su utilización en contextos académicos. Además, en varios casos están orientados a tareas más complejas, como el seguimiento de instrumentos o la detección de fases quirúrgicas, y no siempre proporcionan máscaras segmentadas píxel a píxel con la calidad necesaria.

En cambio, CholecSeg8k, al estar disponible en Kaggle, resultó una opción accesible

que, por la calidad de sus anotaciones y su formato sencillo (que se detallará más adelante), se convirtió en una elección práctica.

3.1.2. Descripción y estructura del *dataset* seleccionado

CholecSeg8k contiene un total de 8.080 imágenes obtenidas a partir de 17 vídeos quirúrgicos distintos. Estas imágenes están distribuidas en 101 carpetas, cada una de las cuales contiene 80 imágenes (o fotogramas del vídeo original). Cada imagen se encuentra acompañada de sus correspondientes máscaras de segmentación, en las que se han etiquetado diferentes estructuras anatómicas.

Las imágenes presentan una resolución de 854×480 píxeles y están segmentadas en un total de 13 clases: Fondo negro, Pared abdominal, Hígado, Tracto gastrointestinal, Grasa, Pinzas, Tejido conjuntivo, Sangre, Conducto cístico, Electrocauterio de gancho en L, Vesícula biliar, Vena y Ligamento hepático, en ese orden de etiquetado. No obstante, no todas las clases están necesariamente presentes en cada imagen del *dataset*.

A diferencia de otros conjuntos más simplificados, CholecSeg8k presenta una organización jerárquica basada en vídeos (17 en total). Concretamente, el *dataset* contiene múltiples carpetas principales denominadas `video01`, `video02`, etc., cada una correspondiente a un procedimiento quirúrgico distinto. Dentro de cada una, se encuentran subcarpetas identificadas por el nombre del vídeo y el número de fotograma que indica el inicio de la secuencia extraída, como por ejemplo `video01_00080`, `video01_00160`, etc.

Cada subcarpeta contiene:

- Una imagen quirúrgica en formato *Portable Network Graphics* (PNG), extraída directamente de un vídeo real de cirugía.
- Varias máscaras de segmentación, también en formato PNG, que delimitan las distintas estructuras anatómicas e instrumentos presentes en la imagen. Estas máscaras se presentan en tres formatos distintos: una máscara en color, una máscara original dibujada a mano y una máscara denominada *watershed*.
    - La **máscara en color** está diseñada para mostrar visualmente las áreas segmentadas de forma clara. Cada clase anatómica o instrumental se representa mediante un color distinto, lo que facilita su interpretación visual.

- La **máscara *watershed*** contiene las mismas regiones segmentadas, pero representadas con valores numéricos uniformes en los tres canales modelo de color rojo, verde y azul (RGB), correspondientes al identificador asignado a cada clase. Esta codificación simplificada está optimizada para facilitar el procesamiento computacional, y por ello es la que se ha empleado para entrenar el modelo.
- La **máscara utilizada por la herramienta de anotación** es la versión original generada manualmente durante el proceso de etiquetado. A partir de ella se derivan tanto la máscara en color como la máscara *watershed*.

A continuación, se muestra la estructura jerárquica del *dataset*:

```
Dataset
|
+-- video01
|   +-- video01_00080
|   |   +-- frame_100_endo.png
|   |   +-- frame_100_endo_color_mask.png
|   |   +-- frame_100_endo_mask.png
|   |   +-- frame_100_endo_watershed_mask.png
|   |   +-- frame_101_endo.png
|   |   +-- frame_101_endo_color_mask.png
|   |   +-- frame_101_endo_mask.png
|   |   +-- frame_101_endo_watershed_mask.png
|   |   +-- ...
|   +-- ...
+-- ...
```

Este formato y la consistencia en los nombres permiten asociar cada imagen con su máscara correspondiente, manteniendo la trazabilidad respecto al vídeo quirúrgico original.

Al no contar con una partición predefinida para entrenamiento, validación y prueba, esta se realizó manualmente: un 80 % de los datos se destinó a entrenamiento, y el 20 % restante se dividió entre validación y prueba.

3.1.3. Preprocesamiento y aumento de datos

Recorte y redimensionado

Todas las imágenes y máscaras se han redimensionado a una resolución fija, reduciéndolas a una cuarta parte de su tamaño original, obteniendo una resolución de 360×640. Esto fue debido no solo a las limitaciones de memoria de la unidad de procesamiento gráfico (GPU) utilizada durante el entrenamiento, sino que también permitió optimizar el uso de recursos computacionales. No obstante, fue necesario asegurarse de que las máscaras de segmentación conservaran sus valores de píxel como etiquetas discretas durante el redimensionado. Para ello, se utilizó el parámetro de interpolación *nearest neighbors*, que permite mantener intactos los valores de clase sin introducir interpolaciones erróneas que distorsionaran la información semántica.

Además, se recortó al máximo el fondo negro presente en las imágenes con el objetivo de eliminar ruido y centrar la atención del modelo en las estructuras anatómicas relevantes, quedando mejor encuadradas.

Normalización

Se ha aplicado normalización de los valores de píxel en las imágenes, escalándolos al rango [0,1] y evitando así grandes saltos en los rangos de entrada. Esta técnica permitió estabilizar y acelerar el proceso de aprendizaje, garantizando que los datos se encuentren en una escala homogénea.

Las máscaras de segmentación no requieren ningún proceso de normalización, ya que no representan valores continuos de intensidad, sino etiquetas discretas correspondientes a las clases. Por tanto, se mantienen sin modificar para preservar la integridad de la codificación de cada clase.

Transformación de máscaras de clase

Las máscaras de segmentación, tanto la versión en color como la *watershed*, se encontraban originalmente en formato RGB. Se optó por utilizar la máscara *watershed*, ya que en ella cada píxel presenta el mismo valor en los tres canales RGB, equivalente a escala de

grises. Este valor numérico representa directamente el identificador de clase asignado por la herramienta de anotación.

Gracias a esta codificación simplificada, fue posible extraer la máscara de clases leyendo su identificador y mapeando ese valor a su clase correspondiente. Esta conversión fue necesaria ya que el formato RGB no es compatible con la función de pérdida utilizada en este PFG, que se detalla en su sección correspondiente, la cual requiere como entrada una máscara de clase en formato numérico, es decir, una matriz bidimensional donde cada valor entero indica directamente la clase del píxel.

Seguidamente, se asignaron colores pastel a cada clase para diferenciar visualmente las predicciones del modelo. La Tabla 3.1 muestra el mapeo entre los identificadores originales, sus índices de clase, el nombre semántico y el color pastel asociado.

Se seleccionó una paleta de colores pastel por razones estéticas y funcionales. Estos colores proporcionan un contraste suficiente para diferenciar las clases sin resultar visualmente agresivo, lo cual es importante en el contexto médico, donde una visualización clara puede facilitar la interpretación de las predicciones. Además, los tonos suaves reducen la fatiga visual durante análisis prolongados.

Tabla 3.1. Correspondencia entre clases, identificadores, índices y nombres de color asignados

ID (*watershed*)	Índice de clase	Nombre de clase	Color pastel asignado
50	0	Fondo negro	Azul pastel
11	1	Pared abdominal	Verde menta
21	2	Hígado	Rosa suave
13	3	Tracto gastrointestinal	Naranja claro
12	4	Grasa	Lila
31	5	Pinza	Amarillo pastel
23	6	Tejido conectivo	Rosa viejo
24	7	Sangre	Melocotón suave
25	8	Conducto cístico	Verde claro pastel
32	9	Electrocauterio	Rosa algodón
22	10	Vesícula biliar	Celeste tenue
33	11	Vena hepática	Kaki claro
5	12	Ligamento hepático	Verde seco pastel

Aumento de datos

Durante el entrenamiento se aplicaron técnicas de aumento de datos orientadas tanto a incrementar la variabilidad del conjunto como a contrarrestar el desbalance entre clases. En particular, se priorizó la generación de ejemplos con estructuras poco representadas (clases 7, 8, 11 y 12), dado que solo 1256 de las 8080 imágenes contienen alguna de estas clases.

Además, las clases 8 y 11 ocupan áreas muy reducidas en dichas imágenes, como se evidencia en la distribución de píxeles mostrada en la Figura 3.1.

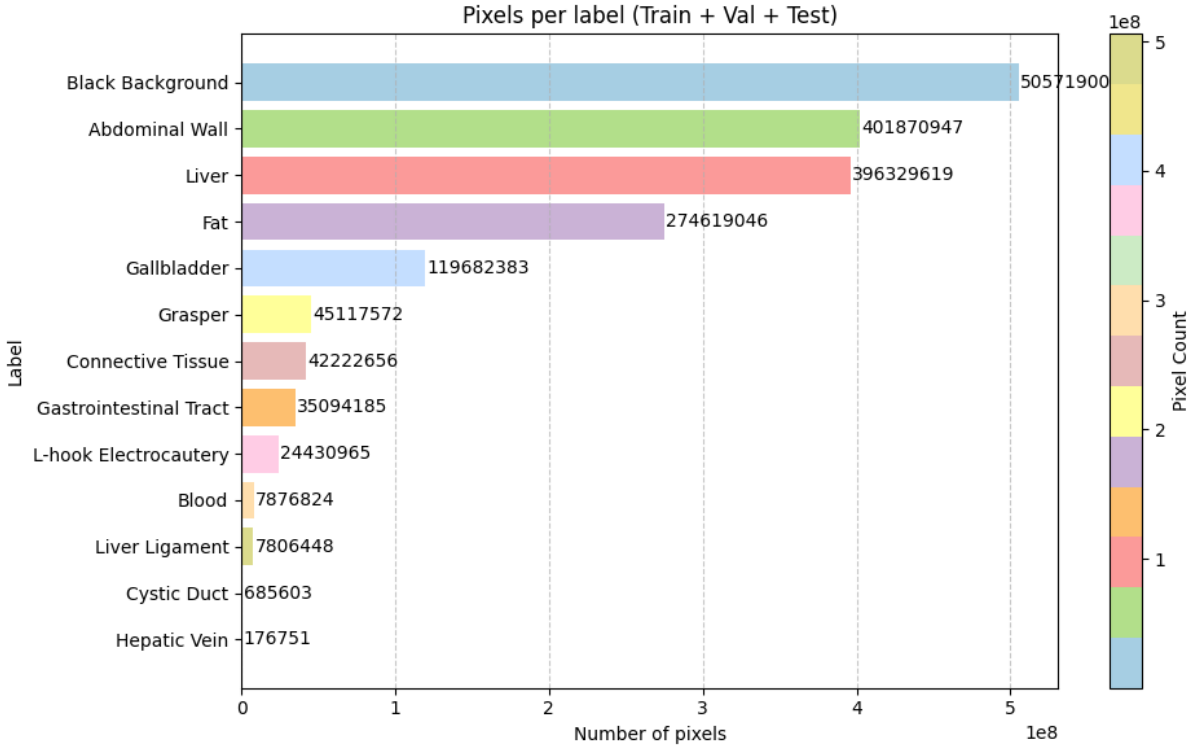

Figura 3.1. Distribución del número total de píxeles por clase en el conjunto de datos.

En términos porcentuales, tal como se muestra en la Tabla 3.2, estas dos clases tienen una representación extremadamente reducida. En particular, el conducto cístico (clase 8) y la vena hepática (clase 11) ocupan tan solo el $0{,}05\,\%$ y el $0{,}018\,\%$ de los píxeles anotados, respectivamente. Estos valores vuelven a confirmar su carácter minoritario dentro del conjunto de datos [38].

Tabla 3.2. Distribución de píxeles anotados por clase en el *dataset* CholecSeg8k [38]

Clase	**Proporción de píxeles (%)**
Pared abdominal	29,460
Hígado	29,393
Grasa	20,230
Vesícula biliar	8,893
Pinza	3,317
Tejido conectivo	3,125
Tracto gastrointestinal	2,557
Electrocauterio	1,824
Sangre	0,573
Ligamento hepático	0,561
Conducto cístico	0,050
Vena hepática	0,018

Se definieron dos conjuntos de transformaciones de datos: uno general, aplicado a todas las imágenes con distribución de clases relativamente equilibrada, y otro específico para aquellas que contienen clases minoritarias. Este segundo conjunto incluye transformaciones más agresivas, orientadas a mejorar la representación de estructuras poco frecuentes, cuidando especialmente de no comprometer la coherencia anatómica debido a la agresividad de las transformaciones.

Adicionalmente, y considerando la baja representación de algunas clases minoritarias, se implementó sobremuestreo con refuerzo progresivo. Las muestras que contienen al menos una de estas clases inicialmente se duplicaron, incrementando su probabilidad de aparición en el entrenamiento. Además, en función del grado de escasez, se aplicó un refuerzo adicional: las muestras con clase 8 fueron añadidas dos veces más, las de clase 11 tres veces más, y las de clase 12 una vez, de esta forma se consigue reducir el sesgo hacia las clases predominantes.

Tras aplicar la estrategia de sobremuestreo descrita, el tamaño final del conjunto de datos quedó repartido de la siguiente forma, siendo un total de 12279 imágenes:

- **Conjunto de entrenamiento:** 9 823 muestras
- **Conjunto de validación:** 1 227 muestras
- **Conjunto de test:** 1 229 muestras

Las transformaciones generales aplicadas incluyen:

- Rotaciones aleatorias dentro de un rango de ± 30 grados, para simular diferentes orientaciones de la cámara durante la intervención.
- Volteos horizontales y verticales con una probabilidad de $0{,}5$, que permiten aumentar la diversidad espacial de las imágenes.
- Ajustes aleatorios de brillo, contraste y saturación, que simulan variaciones en las condiciones de iluminación del quirófano.
- Aplicación de pequeños *zooms* o recortes aleatorios, que introducen deformaciones locales leves.
- Modificación de propiedades fotométricas de la imagen para simular condiciones de iluminación y color propias de entornos clínicos reales. En concreto, se aplican alteraciones aleatorias sobre tres atributos visuales:
  - **Brillo**: ajusta la intensidad global de la imagen.
  - **Contraste**: modifica la diferencia entre regiones claras y oscuras.
  - **Saturación**: altera la viveza y riqueza de los colores.

  Estas últimas transformaciones no afectan a las máscaras de segmentación, lo que permite mantener la coherencia anatómica mientras se introduce variabilidad visual útil para mejorar la generalización del modelo.

  Las demás transformaciones se aplicaron de manera consistente tanto a las imágenes como a sus respectivas máscaras. Esto garantiza que la correspondencia espacial entre la entrada y su etiqueta se mantenga intacta y evita cualquier desalineación que comprometiese la validez del proceso de entrenamiento.

Finalmente, las transformaciones especiales sobre las clases minoritarias incluyen:

- **Transformaciones afines** con *zooms in/out* y rotaciones aleatorias, que alteran la escala y orientación de las estructuras.
- **Cortes centrados** dentro de regiones cuadradas que contienen las clases minoritarias, para forzar su aparición en el campo visual del modelo.
- **Ajustes fotométricos intensificados**, como brillo o contraste más amplios que los aplicados al conjunto general.
- **Reflexiones horizontales y verticales** aplicadas con mayor probabilidad, con el objetivo de aumentar la diversidad posicional y simular distintas orientaciones anatómicas.
- **Transformaciones elásticas leves**: deformaciones no rígidas que simulan la flexibilidad y deformabilidad natural de los tejidos.

3.2. Entorno de desarrollo y tecnologías utilizadas

3.2.1. Frameworks de DL

El desarrollo del proyecto se ha realizado íntegramente en **Python**, debido a su amplia adopción en entornos de investigación y a su ecosistema maduro para tareas de visión por computador. Para la implementación y entrenamiento del modelo se ha utilizado el **framework PyTorch**, desarrollado por Facebook AI Research.

Inicialmente PyTorch fue recomendada por el tutor, sin embargo, su uso ha resultado propicio debido a que: ofrece aceleración por GPU, gráficos computacionales dinámicos y una interfaz intuitiva para investigadores y desarrolladores de aprendizaje profundo. PyTorch sigue un enfoque de *define-by-run*, lo que significa que sus gráficos computacionales se construyen sobre la marcha, lo que permite una mejor depuración y personalización de modelos [39].

Además, dispone de una comunidad activa y abundante documentación, con numerosos ejemplos específicos de segmentación.

3.2.2. Librerías empleadas

A lo largo del desarrollo se han utilizado diversas librerías auxiliares, agrupadas a continuación según su propósito:

- **Manejo de datos y procesamiento numérico**:
  - **NumPy**: manipulación eficiente de arreglos y operaciones matemáticas.
  - **Pandas**: gestión de anotaciones, métricas y estructuras tabulares.
  - **os**, **sys**, **math**, **random**, **collections**: utilidades del sistema, generación de aleatoriedad y estructuras de datos especializadas.
- **Transformaciones y carga de datos**:
  - **torchvision.transforms**, **torchvision.io**: transformaciones clásicas (redimensionado, normalización) y lectura de imágenes.
  - **Albumentations**: *augmentations* avanzadas específicas para visión por computador.
  - **Kornia**: transformaciones geométricas y fotométricas diferenciables compatibles con PyTorch.
- **Entrenamiento y optimización**:
  - **torch.nn**, **torch.nn.functional**, **torch.optim**: definición del modelo, funciones de activación, pérdidas y optimizadores.
  - **torch.utils.data**: gestión de *datasets* personalizados y carga eficiente mediante *dataloaders*.
  - **torch.optim.lr_scheduler**: implementación del planificador de *Cosine Annealing*.
- **Evaluación y visualización**:
  - **sklearn.metrics (confusion_matrix)**: evaluación de rendimiento del modelo.
  - **matplotlib.pyplot**, **matplotlib.colors**: visualización de imágenes, curvas de entrenamiento y mapas de clase.
  - **seaborn**: representación gráfica de métricas y matrices de confusión.

- **Librerías estándar**:
  - **time**: permite calcular el tiempo de ejecución del entrenamiento. Usada para detectar automáticamente cuándo se aproxima el límite de uso de recursos en Kaggle y guardar el modelo antes de que se interrumpa forzosamente.

3.2.3. Hardware utilizado

La naturaleza exigente, en términos de capacidad de cálculo del proyecto, ha forzado a buscar recursos en la nube para poder hacer uso de GPUs gratuitas, en esta búsqueda, se priorizó aquellos entornos que ofrecieran mayor flexibilidad para ejecuciones prolongadas.

- **Kaggle Notebooks**: fue el entorno principal utilizado, debido a su compatibilidad directa con el *dataset* CholecSeg8k, su integración con aceleradores GPU gratuitos y, sobretodo, por su capacidad para ejecutar *notebooks* incluso sin conexión, siendo una plataforma es idónea para entrenamientos prolongados.
- **Google Colab Pro**: se utilizó como entorno complementario al no permitir la ejecución sin conexión, usada especialmente en aquellos momentos en los que se agotaba el tiempo de uso de GPU en Kaggle. Permitió continuar con los entrenamientos y ajustes sin interrupciones, asegurando así la continuidad del desarrollo.
- **Equipo local**: también se utilizó de forma puntual para tareas de depuración y pruebas rápidas de código. Permitió agilizar el desarrollo mediante el análisis preliminar de los *scripts* antes de ejecutarlos en la nube.

Esta combinación de recursos facilitó un flujo de trabajo ágil, asegurando alta disponibilidad de cómputo sin infraestructura propia.

3.3. Diseño del modelo

3.3.1. Arquitectura base

El modelo empleado en este PFG se basa en la arquitectura *attention U-Net*, una extensión de la U-Net original que incorpora mecanismos de atención con el objetivo de mejorar el enfoque del modelo sobre regiones anatómicamente relevantes durante el proceso de decodificación.

La elección de la arquitectura *attention U-Net* responde a la necesidad de alcanzar un equilibrio entre precisión en la segmentación, robustez frente a la variabilidad de los datos clínicos y viabilidad computacional.

Si bien modelos híbridos como TransUNet, o arquitecturas completamente basadas en Transformers como Swin-Unet, han demostrado un rendimiento notable en tareas de segmentación médica, su aplicación práctica presenta limitaciones: estos enfoques requieren habitualmente grandes volúmenes de datos para evitar el sobreajuste, así como recursos computacionales elevados que pueden no estar disponibles en entornos clínicos o académicos, en este caso.

En contraposición, *attention U-Net* ofrece una mejora sustancial respecto a la U-Net clásica, incorporando mecanismos de atención que permiten al modelo enfocarse dinámicamente en las regiones más relevantes de la imagen. Esta capacidad resulta especialmente útil para segmentar estructuras pequeñas y poco diferenciadas, todo ello manteniendo un diseño arquitectónico relativamente sencillo y eficiente para su entrenamiento y despliegue [19].

Además, se ha incorporado BN tras las operaciones convolucionales en cada bloque del modelo. Esta técnica estabiliza la distribución de las activaciones internas durante el entrenamiento, acelera la convergencia y actúa como un regularizador adicional que contribuye a mejorar la generalización del modelo [31].

La red implementada consta de cinco bloques principales en la fase de codificación, seguidos por cuatro bloques de decodificación con compuertas de atención, y una capa final de salida.

A continuación, se describe en detalle la estructura:

Cada bloque de codificación está compuesto por dos capas convolucionales con BN y funciones de activación ReLU, seguidas opcionalmente *dropout* (probabilidad de 0,3) y *max pooling*. El cuello de la *U* consiste en un bloque convolucional profundo sin *pooling*. En la fase de decodificación, se emplean bloques de *upsampling* con convoluciones transpuestas y convoluciones adicionales. Además, las conexiones de salto se refinan mediante el mecanismo de atención *Attention Gate*. Finalmente, una capa convolucional 1×1 mapea las características extraídas a las 13 clases de segmentación.

Tabla 3.3. Estructura detallada de la arquitectura *attention U-Net* utilizada en el proyecto

Bloque	**Descripción**	**Salida**
`Input`	Imagen RGB	$3 \times 360 \times 640$
`Block 1`	Conv(3→64) + BN + ReLU $\times$ 2 + *max pooling*	$64 \times 180 \times 320$
`Block 2`	Conv(64→128) + BN + ReLU $\times$ 2 + *max pooling*	$128 \times 90 \times 160$
`Block 3`	Conv(128→256) + BN + ReLU $\times$ 2 + *max pooling*	$256 \times 45 \times 80$
`Block 4`	Conv(256→512) + BN + ReLU $\times$ 2 + *dropout* + *max pooling*	$512 \times 22 \times 40$
`Block 5`	Conv(512→1024) + BN + ReLU $\times$ 2 + *dropout*	$1024 \times 22 \times 40$
`UpBlock 6`	TransConv(1024→512) + **Conexión de salto refinada** + Conv $\times$ 2	$512 \times 45 \times 80$
`UpBlock 7`	TransConv(512→256) + **Conexión de salto refinada** + Conv $\times$ 2	$256 \times 90 \times 160$
`UpBlock 8`	TransConv(256→128) + **Conexión de salto refinada** + Conv $\times$ 2	$128 \times 180 \times 320$
`UpBlock 9`	TransConv(128→64) + **Conexión de salto refinada** + Conv $\times$ 2	$64 \times 360 \times 640$
`Output`	Conv(64→13), *softmax* implícito en pérdida	$13 \times 360 \times 640$

3.3.2. Configuración de hiperparámetros y función de pérdida

Los valores de los hiperparámetros se determinaron tras realizar numerosas pruebas empíricas, basándose tanto en la literatura como en la observación directa de los resultados durante y después del entrenamiento, analizando las gráficas obtenidas al finalizar cada sesión. A continuación, se detallan los valores seleccionados junto con la justificación de cada uno:

- **Tasa de aprendizaje:** se utilizó un valor inicial de 3×10^{-4}, el cual ofreció un equilibrio adecuado entre velocidad de convergencia y estabilidad.
- **Tamaño de lote:** se fijó en 4 debido a las limitaciones de memoria de los entornos utilizados. A pesar de su tamaño reducido, este valor permitió una estimación razonable del gradiente sin provocar errores por falta de memoria.

- **Número de épocas:** se definió un máximo inicial de 50 épocas, suficiente para que el modelo alcanzara la convergencia en la mayoría de los casos.
- **Optimizador:** se eligió Adam debido a su buena capacidad de adaptación durante las primeras fases del entrenamiento. Este optimizador combina los beneficios de AdaGrad y RMSProp, ajustando dinámicamente las tasas de aprendizaje por parámetro. Se añadió un término de *weight decay* (1×10^{-4}) para actuar como regularizador y prevenir el sobreajuste.

Respecto a la función de pérdida, se empleó una versión adaptada de la **Focal Loss**, con los siguientes componentes:

- **Pérdida de Focalización (Focal Loss):** La pérdida se basa en la *CrossEntropy*, introduciendo un *término de enfoque* controlado por el parámetro γ, que amplifica el peso de los errores cometidos sobre ejemplos difíciles. En lugar de fijar γ a un valor constante, se utilizó un esquema de ajuste dinámico. Este comportamiento progresivo permite al modelo centrarse primero en clases mayoritarias (con γ bajo) y después intensificar el aprendizaje de las clases minoritarias conforme aumenta γ.
- **Ponderación por clase (α):** Se introdujo una ponderación específica por clase mediante un vector α, asignando mayor peso a las clases minoritarias para compensar su escasa representación. A su vez, se redujo el peso asignado a la clase fondo, con el objetivo de evitar que esta clase, mayoritaria, dominara la función de pérdida.

Con esto se concluyen los componentes de la función de pérdida. A continuación, se describen los mecanismos restantes del entrenamiento:

- **CosineAnnealingLR**: Su funcionamiento se basa en una reducción progresiva y suave de la tasa de aprendizaje siguiendo una curva cosenoidal, que comienza en un valor máximo y decrece hasta un mínimo prefijado a lo largo de un número determinado de épocas, en este caso 50. El comportamiento oscilante de la curva puede favorecer la salida de mínimos locales durante las primeras etapas, facilitando una mejor exploración del espacio. A diferencia de otras técnicas como el reinicio del optimizador (que fue usado en algunas pruebas preliminares) esta estrategia ofrece un equilibrio entre estabilidad y capacidad de adaptación sin interrumpir el proceso de entrenamiento.

- ***Early Stopping***: Por último, se empleó *early stopping* con una paciencia de 8 épocas y un umbral mínimo de mejora de $0{,}001$ sobre la pérdida de validación. Esto permitió finalizar el entrenamiento prematuramente cuando el modelo dejaba de mejorar de forma sustancial, evitando el sobreajuste.

3.3.3. Configuración del entrenamiento

El proceso de entrenamiento se diseñó para adaptarse tanto a las características del modelo como a las limitaciones prácticas del entorno de ejecución. Desde el inicio, se fijaron semillas aleatorias para todas las librerías implicadas (`Python`, `NumPy`, `PyTorch` en unidad central de procesamiento (CPU) y GPU) con el objetivo de garantizar la reproducibilidad de los resultados, evitando variaciones entre ejecuciones debidas al comportamiento estocástico de los componentes del entrenamiento.

Dado que el entrenamiento se realizó mayoritariamente en la plataforma Kaggle, se impuso una restricción importante: cada sesión con GPU tiene un tiempo máximo de ejecución de aproximadamente 12 horas. Por ello, se implementó un sistema automático de control de tiempoq, con un pequeño margen, que interrumpe el entrenamiento a las 11 horas. En ese punto, se guarda un *checkpoint* completo, lo que permite reanudar el entrenamiento desde el último punto alcanzado, sin pérdida de información, este incluye:

- Estado del modelo.
- Estado del optimizador.
- Estado del planificador.
- Época actual.
- Historial completo de métricas: pérdida de entrenamiento, pérdida de validación, precisión por píxel, IoU medio por época, mejor perdida registrada y visualizaciones asociadas.
- Estado del early stopper.

Finalmente, aclarar que durante cada iteración se aplicaban las transformaciones de aumento de datos al conjunto de entrenamiento. Por el contrario, los conjuntos de validación y prueba no incluían ninguna transformación. Además, tras cada época se evaluaba el rendimiento del modelo usando el conjunto de validación.

3.3.4. Criterios de parada y validación

Durante el entrenamiento se aplicó un enfoque intensivo de depuración para monitorizar continuamente el comportamiento del modelo. Más allá de las métricas clásicas como la pérdida o la precisión, se incorporó la visualización periódica de imágenes reales junto con sus máscaras predichas y de referencia. Estas se seleccionaban cuando contenían clases minoritarias, permitiendo comprobar visualmente si el modelo era capaz de detectarlas correctamente.

Esta inspección cualitativa permitía identificar:

- Falsos positivos y falsos negativos visuales, especialmente en estructuras pequeñas o anatómicamente complejas.
- Si las predicciones eran triviales (por ejemplo: todo era fondo o una sola clase dominante).

Además, se mostraban imágenes intermedias durante el entrenamiento para verificar qué transformaciones se estaban aplicando en cada imagen y si estaban afectando negativamente a la coherencia entre imagen y máscara o, perturbaban excesivamente la imagen.

También se verificaba si los valores de pérdida eran razonables. Se estableció un umbral de alerta si la pérdida de un lote superaba el valor de $15{,}0$, lo cual activaba una interrupción en la ejecución.

Tras la quinta época, se visualizaban gráficas de evolución de las métricas de pérdida, precisión por píxel e IoU medio. Se esperaba observar una disminución progresiva de la pérdida de entrenamiento, junto con una estabilización o ligera mejora en la pérdida de validación. Fluctuaciones erráticas o aumentos repentinos indicaban posibles casos de sobreajuste.

No obstante, la decisión de continuar o detener el entrenamiento no se basaba exclusivamente en la pérdida o las visualizaciones. Se evaluaba también si las clases minoritarias presentaban mejoras efectivas; por ejemplo: si el IoU de estructuras como el conducto cístico (clase 8) o la vena hepática (clase 11) aumentaba con cada época, o si comenzaban a detectarse tras haber sido ignoradas anteriormente. En caso contrario, se consideraba detener el entrenamiento para revisar la estrategia de aumentos o función de pérdida.

El mecanismo de early stopping se activaba automáticamente si no había mejora sustancial en la pérdida de validación. Al hacerlo, se conservaba el modelo gracias al sistema de checkpoints.

3.3.5. Resumen de la configuración del sistema

En la Tabla 3.4 se presentan los principales parámetros y configuraciones utilizadas durante el desarrollo y entrenamiento del modelo.

Tabla 3.4. Resumen de configuración del sistema de entrenamiento

Componente	**Configuración**
Tasa de aprendizaje	3×10^{-4}
Tamaño de lote	4
Optimización	Adam + *weight decay* (10^{-4})
Planificador	CosineAnnealingLR ($T_{\max} = 50$, $\eta_{\min} = 10^{-6}$)
Pérdida	Focal Tversky Loss con γ dinámico
Pesos por clase ($\boldsymbol{\alpha}$)	[0,25, 1,0, 1,0, 1,15, 1,0, 1,0, 1,25, 3,0, 3,8, 1,5, 1,0, 4,0, 3,5]
Aumentos	Generales + específicas para clases minoritarias
Early Stopping	Paciencia de 8 épocas, $\Delta \geq 0{,}001$
Checkpoint	Guardado del mejor modelo según validación

3.3.6. Esquema general del proceso

La Figura 3.2 muestra el flujo general durante el desarrollo y entrenamiento del modelo:

- **Carga del *dataset***: se recorren las carpetas del conjunto CholecSeg8k para cargar imágenes RGB y sus máscaras *watershed*, identificando muestras con clases minoritarias para aumentos dirigidos.
- **Preprocesamiento**: redimensionado a 360×640 píxeles, normalización al rango [0,1] y conversión de máscaras RGB a etiquetas numéricas.
- **Aumento de datos**: aplicación de transformaciones específicas según presencia de clases minoritarias.
- **Forward pass**: las imágenes se procesan con el modelo *attention U-Net*.

- **Cálculo de pérdida**: se usa Focal Loss con ajuste dinámico de γ y ponderación de clases.
- ***Backpropagation* y optimización**: actualización de parámetros con Adam y ajuste de tasa de aprendizaje mediante CosineAnnealingLR.
- **Evaluación y visualización**: cálculo de métricas en validación, visualización de predicciones y matriz de confusión.
- **Checkpoint y finalización**: guardado del mejor modelo, terminación anticipada por early stopping o al alcanzar el límite de tiempo; y validación final en el conjunto de test.

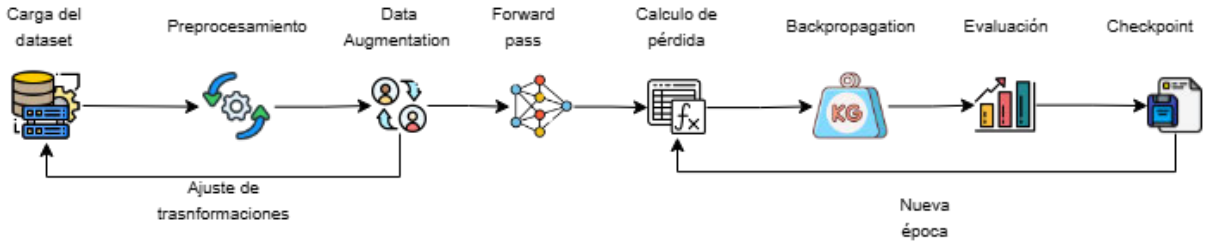

Figura 3.2. Esquema general del flujo de trabajo del modelo.

4. Resultados

4.1. Métricas de evaluación

4.1.1. Análisis de la función de pérdida

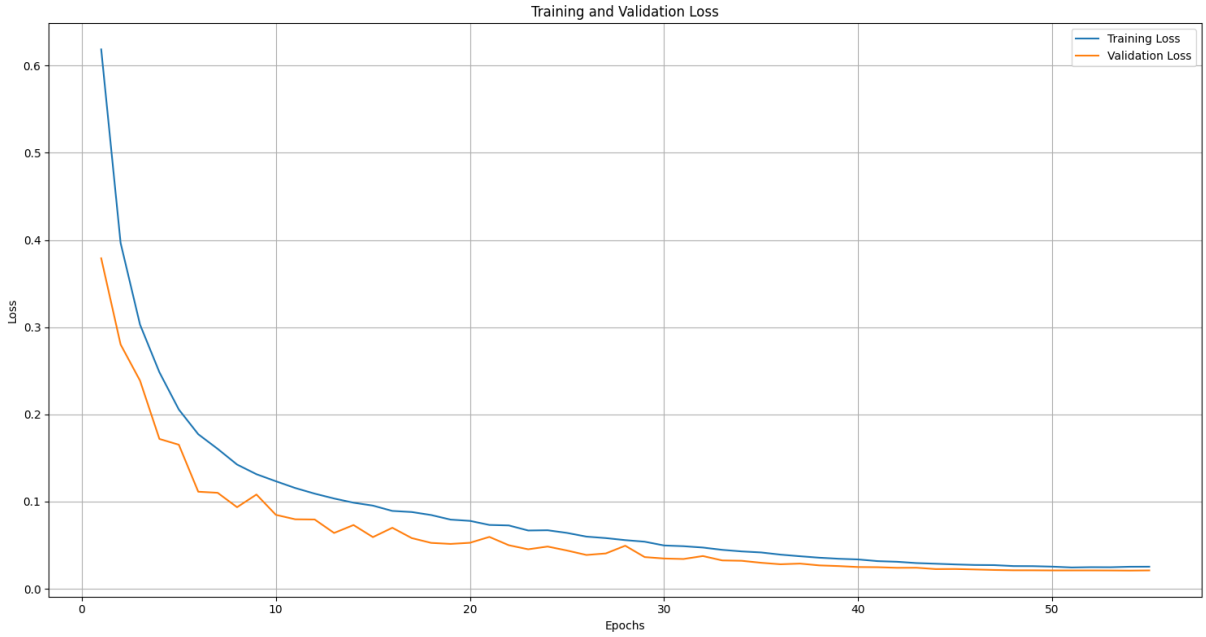

Figura 4.1. Evolución de la pérdida durante el entrenamiento y la validación del modelo

La Figura 4.1 muestra la evolución de la **pérdida de entrenamiento** (en azul) y la **pérdida de validación** (en naranja) a lo largo de 55 épocas. Inicialmente, se estableció un máximo de 50 épocas, pero el entrenamiento se prolongó hasta que el mecanismo de early stopping se activase, al no observar mejoras significativas en la validación.

Lo más destacable es que: a partir de la época 30, la pérdida de validación se estabiliza, indicando que el modelo alcanzó un punto en el que las mejoras son marginales.

Finalmente, se conservó el modelo correspondiente a la última época completada (55), ya que no se detectó un empeoramiento relevante que justificara conservar un modelo anterior.

4.1.2. Análisis de la precisión por píxel

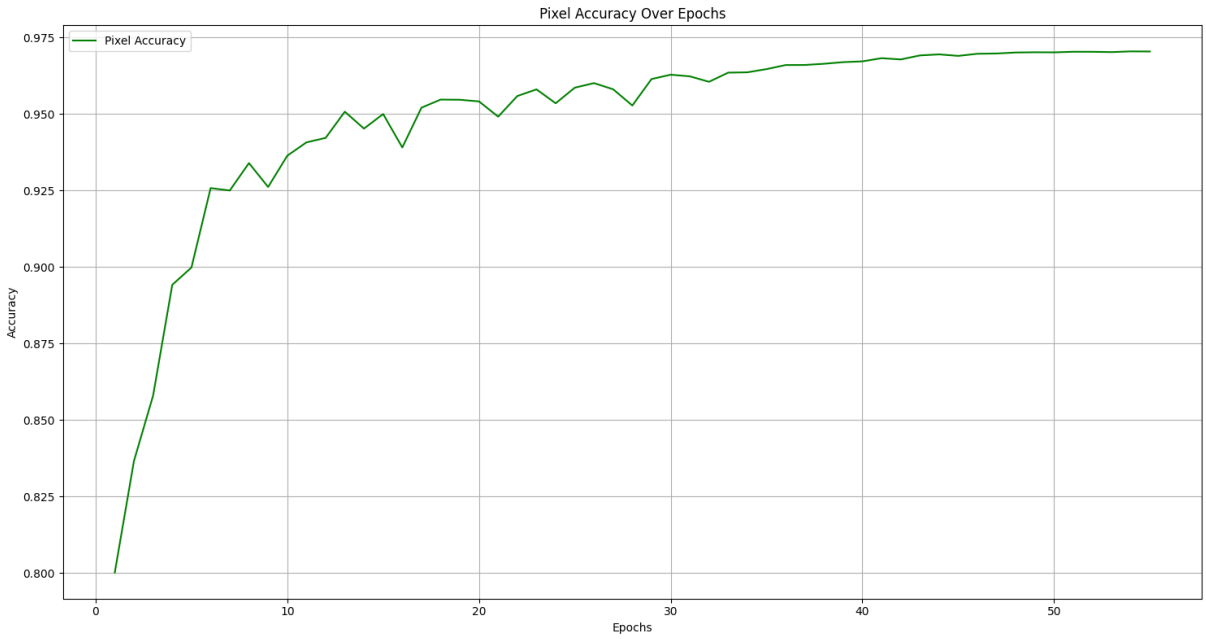

Figura 4.2. Evolución de la precisión por píxel durante las épocas de entrenamiento

La Figura 4.2 muestra la evolución de la **precisión por píxel** a lo largo del entrenamiento.

Se observa un incremento rápido de la precisión durante las primeras épocas, seguido de una estabilización progresiva a partir de la época 35-40, en consonancia con la evolución observada en la pérdida de validación.

4.1.3. Cálculo del IoU medio

El IoU medio se calculó mediante una función personalizada, en este caso, el cálculo del **IoU se realiza únicamente sobre las clases presentes en el *ground truth***, omitiendo por completo aquellas que no aparecen en una imagen determinada. Esta decisión evita

penalizar al modelo por clases que no tenía forma de predecir, pero también implica que el valor de IoU medio puede ser ligeramente más alto que si se consideraran todas las clases posibles.

Además, las métricas se agregan de manera acumulativa a lo largo de todos los lotes del conjunto de validación. Para cada clase presente, se acumulan sus IoU individuales y luego se calcula el promedio por clase. Para terminar, el IoU medio global se obtiene como la media aritmética de los valores válidos por clase.

Esta forma de cálculo resulta especialmente adecuada en este contexto de escasa aparición de ciertas clases, ya que permite una evaluación más justa del rendimiento real del modelo.

4.1.4. Análisis de la calidad en segmentación

La curva presenta una pendiente acentuada en las primeras 15 épocas aproximadamente, con un incremento aproximado de $+0{,}30$ en el IoU medio, seguido de una fase de estabilización progresiva. Esta dinámica sugiere que el modelo primero aprende a identificar formas globales y estructuras principales, para luego refinar detalles más complejos como los contornos vasculares o tejidos de bajo contraste.

La evolución del IoU medio mostrada en la Figura 4.3 revela que, al final del entrenamiento, el IoU medio alcanza valores superiores al **85 %**.

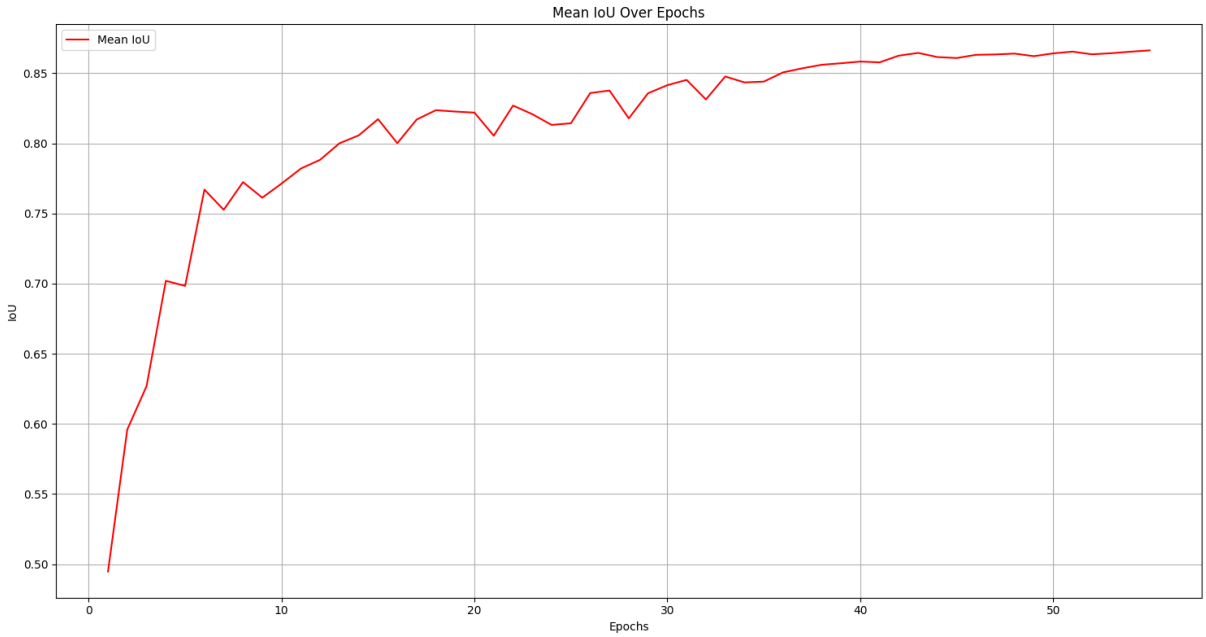

Figura 4.3. Evolución del IoU medio durante el entrenamiento

Este nivel de rendimiento se encuentra en la franja considerada de estado del arte para tareas de segmentación semántica. Por ejemplo, Yu et al. [40] alcanzaron un IoU medio de 86,2 % en el conjunto PASCAL VOC 2012, así como 80,3 % en Cityscapes.

En el ámbito médico, diferentes revisiones han reportado que modelos basados en U-Net suelen superar el 80 % de IoU en contextos clínicos reales:

Azad et al. [41] evalúan experimentalmente el rendimiento de U-Net en distintos escenarios clínicos, obteniendo valores de IoU de 0,8491 en el conjunto ISIC 2018, 0,8824 en SegPC 2021 y un promedio del 69,64 en *Synapse*. Asimismo, Caicedo et al.**<empty citation>**caicedo201 en un reto internacional de segmentación celular, informaron un IoU medio de 0,80 como referencia competitiva. En conjunto, estos trabajos respaldan que alcanzar o superar un IoU medio de 0,85 constituye un resultado notable dentro del estado actual de la técnica.

4.2. Evaluación final tras época 55

Tras completar el entrenamiento en la época 55, el modelo fue evaluado en el conjunto de *test*, obteniendo los resultados que se ven en la Tabla 4.1:

Tabla 4.1. Resultados del modelo tras el entrenamiento sobre el conjunto de test

Nombre de la clase	IoU
Loss	0,0211
Precisión por pixel	0,9702
IoU medio	0,8714

Los resultados cuantitativos obtenidos en el conjunto de test reflejan un rendimiento sobresaliente del modelo. El *loss* extremadamente bajo sugiere que las predicciones del modelo se ajustan con gran precisión a las máscaras de segmentación, incluso en condiciones visuales adversas de iluminación variable o bordes poco definidos.

La **precisión por píxel** confirma que el modelo clasifica correctamente la inmensa mayoría de los píxeles. Una segmentación fiable de las clases grandes también es de suma importancia debido a que estas clases ofrecen el marco de referencia en el que se sitúan las estructuras más pequeñas. Si estas regiones no están bien segmentadas, incluso una detección precisa de las clases pequeñas puede carecer de sentido clínico.

El IoU medio alcanzado posiciona al modelo dentro de un rango alto de rendimiento en segmentación semántica médica. No solo supera ampliamente la media esperada para arquitecturas como *attention U-Net*, sino que se aproxima al rendimiento de modelos más complejos como TransResUNet o U-Net++, que suelen situarse entre 0,85 y 0,89 en *dataset* clínicos como Kvasir-SEG o BKAI-IGH [42].

4.3. Análisis de resultados por categoría

Los resultados del IoU por clase muestran un desempeño alto en la mayoría de categorías, aunque persisten diferencias notables en clases con menor representación o mayor complejidad visual. La Tabla 4.2 resume el IoU por clase en el conjunto de test.

Tabla 4.2. IoU por clase obtenido en el conjunto de validación

Clase	Nombre de la clase	IoU	Clase	Nombre de la clase	IoU
0	Fondo	0,9566	7	Sangre	0,6766
1	Pared Abdominal	0,9639	8	Conducto Cístico	0,7861
2	Hígado	0,9500	9	Electrocauterio	0,9248
3	Vesícula Biliar	0,8578	10	Pinza	0,9132
4	Grasa	0,9446	11	Vena Hepática	0,6380
5	Estómago	0,8668	12	Ligamento Hepático	0,9653
6	Intestino Delgado	0,8851			

El modelo segmenta con **casi total precisión las clases más frecuentes**. El Fondo ($0,9566$), la Pared Abdominal ($0,9639$), el Hígado ($0,9500$) y la Grasa ($0,9446$) obtienen un IoU superior al 94 %, probablemente debido a su gran presencia en el *dataset* y a que sus texturas son homogéneas y sus bordes bien definidos.

En las estructuras anatómicas más complejas, el avance también es notable. La Vesícula Biliar ($0,8578$), el Estómago ($0,8668$) y el Intestino Delgado ($0,8851$) mejoran considerablemente respecto a versiones anteriores, consiguiendo segmentaciones detalladas pese a que morfológicamente son muy variables.

Sin embargo, la Vena Hepática ($0,6380$) y el Conducto Cístico ($0,7861$) siguen suponiendo un problema. Aunque han progresado, su baja presencia en el conjunto de datos y sus límites menos nítidos dificultan una buena segmentación.

Entre las clases menos frecuentes, el Ligamento Hepático destaca con un IoU de $0,9653$, el más alto de todas, un logro impresionante dada su morfología –muy membranosa–. La Sangre, con un IoU de $0,6766$, presenta una gran mejora frente a versiones pasadas, a pesar de que su aspecto variable –frecuentemente confuso con sombras o tejidos oscuros– sigue siendo difícil de diferenciar.

Destacar que funciona de forma **muy fiable con el instrumental quirúrgico**: la Pinza ($0,9132$) y el Electrocauterio ($0,9248$) superan el 91 % de IoU. Esta precisión es esencial en un proceso de cirugía real, ya que es imprescindible identificar bien las herramientas para ayudar a guiarse y a evitar errores.

4.3.1. Matriz de Confusión

Para seguir analizando el desempeño del modelo clase por clase, se calcularon métricas derivadas de la matriz de confusión, la cual se ilustra en la Figura 4.4.

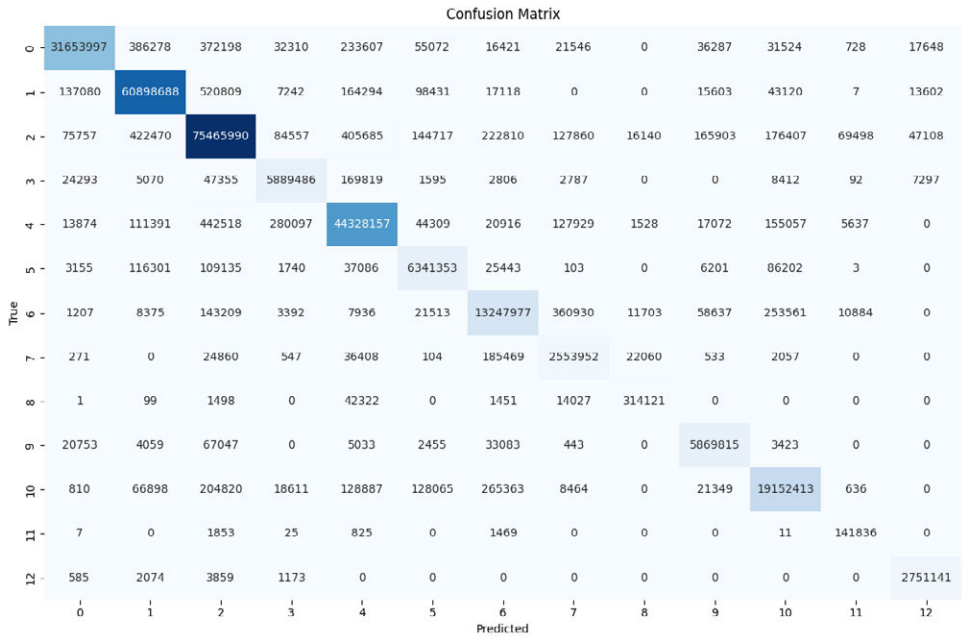

Figura 4.4. Matriz de confusión del modelo del conjunto de test

La Tabla 4.3 ofrece valores derivados de esta matriz.

Tabla 4.3. Métricas derivadas de la matriz de confusión para cada clase: sensibilidad, FPR y FNR

Clase	Nombre	Sensibilidad (%)	FPR (%)	FNR (%)
0	Fondo	96,57	0,12	3,43
1	Pared Abdominal	98,28	0,49	1,72
2	Hígado	97,34	0,90	2,66
3	Vesícula Biliar	94,84	0,14	5,16
4	Grasa	97,51	0,58	2,49
5	Estómago	94,31	0,19	5,69
6	Intestino Delgado	93,59	0,31	6,41
7	Sangre	90,91	0,26	9,09
8	Conducto Cístico	84,80	0,02	15,20
9	Electrocauterio	97,66	0,12	2,34
10	Pinza	96,02	0,32	3,98
11	Vena Hepática	95,47	0,03	4,53
12	Ligamento Hepático	99,77	0,03	0,23

Se observa que la clase Conducto Cístico presenta la mayor tasa de falsos negativos (15,20 %), seguido por Sangre con un 9,09 %.

Las métricas derivadas de la matriz de confusión no suman necesariamente 100 % por clase debido a que:

- Sensibilidad y FNR se calculan sobre el total de píxeles reales de una clase, y por tanto en cada clase se cumple que Sensibilidad + FNR = 100 %.
- La FPR se calcula sobre le conjunto de píxeles que **no pertenecen a esa clase**, por tanto no es complementaria de la sensibilidad.

4.4. Visualización de predicciones

Para evaluar visualmente el rendimiento del modelo, se presentan comparaciones de cuatro clases minoritarias entre las predicciones de la primera y última época.

La Figura 4.5 muestra la clase «conducto cístico» en verde pastel en la segmentación.

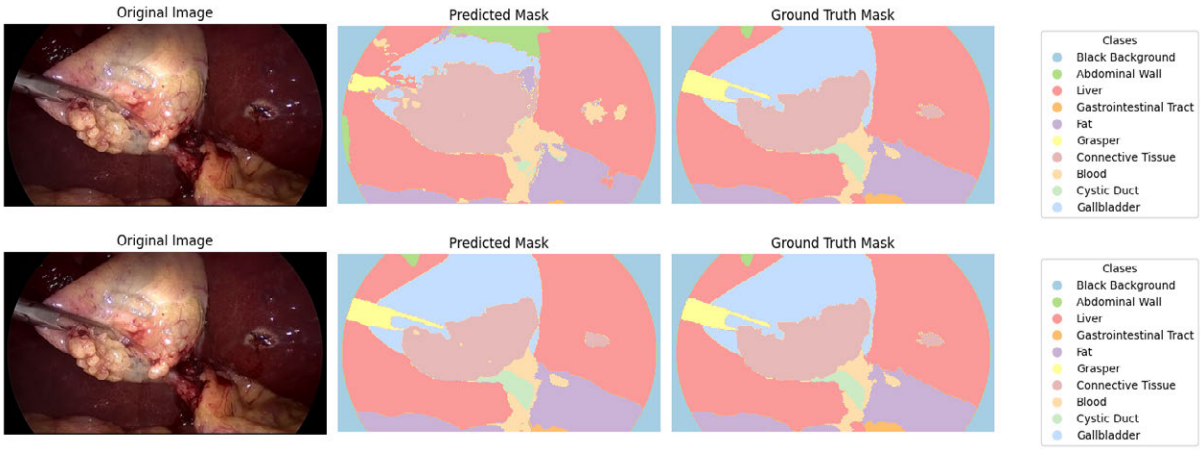

Figura 4.5. Ejemplo para la clase «conducto cístico» durante la primera (arriba) y última iteración (abajo)

La Figura 4.6 muestra la clase «vena hepática» marcada de kaki en la segmentación.

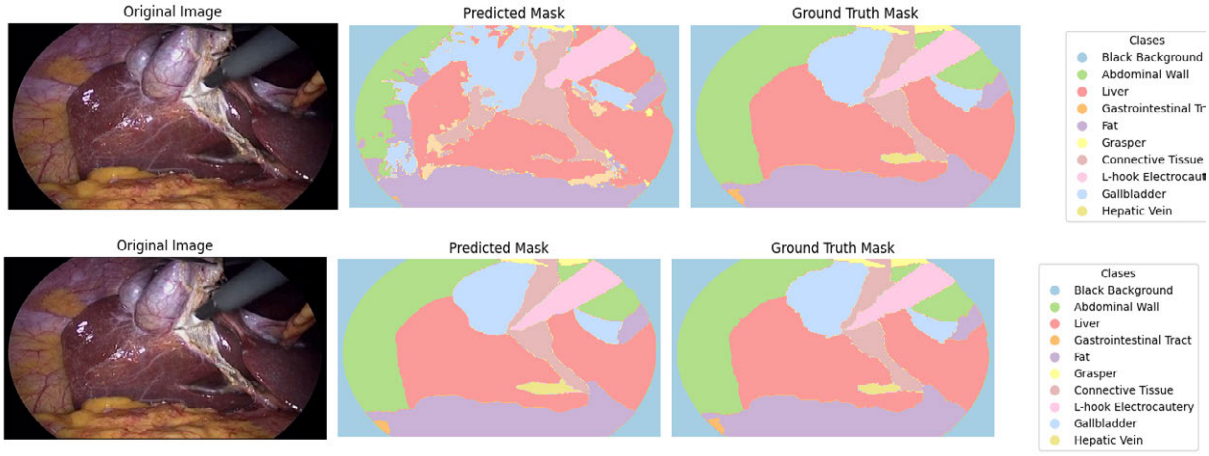

Figura 4.6. Ejemplo para la clase «vena hepática» durante la primera (arriba) y última iteración (abajo)

La Figura 4.7 muestra la clase «sangre» marcada de melocotón en la segmentación.

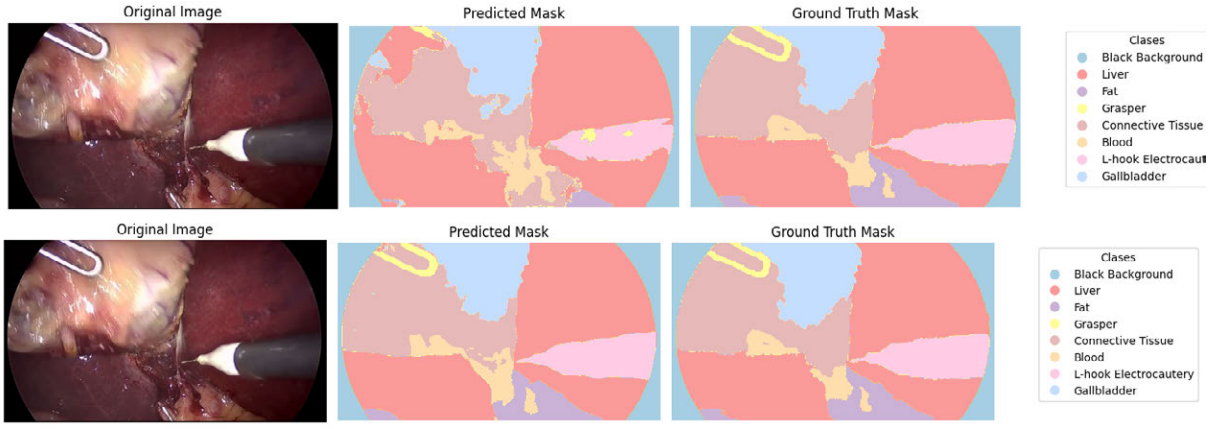

Figura 4.7. Ejemplo para la clase «sangre» durante la primera (arriba) y última iteración (abajo)

La Figura 4.8 muestra la clase «ligamento hepático», verde en la segmentación.

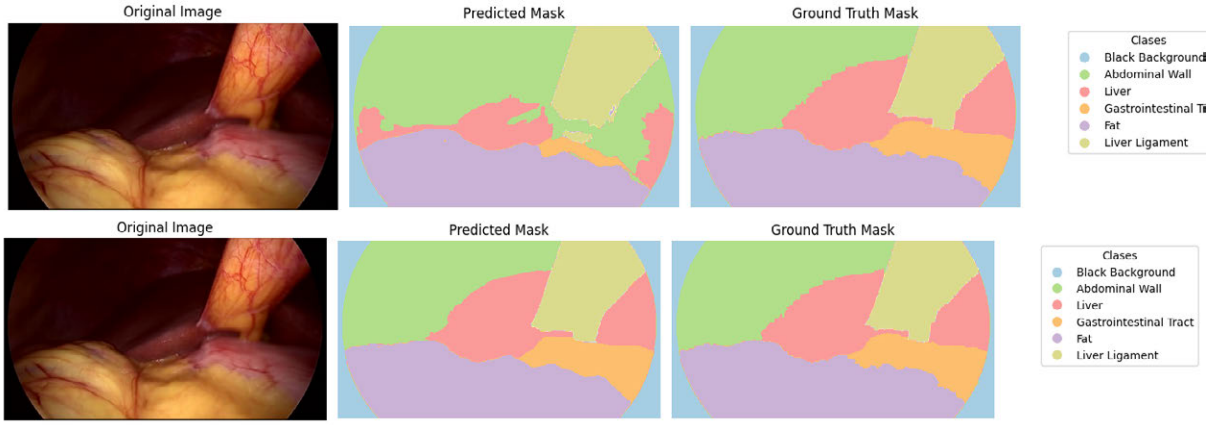

Figura 4.8. Ejemplo para la clase «ligamento hepático» durante la primera (arriba) y última iteración (abajo)

Aunque no es una evaluación formal, sí que nos permite una visualización de cómo mejora el modelo a lo largo del proceso de entrenamiento y su capacidad para manejar los casos más complejos del *dataset*.

5. Conclusiones

En este PFG se ha conseguido desarrollar una CNN (basada en la arquitectura *attention U-Net*) que segmenta de forma eficiente imágenes laparoscópicas, combinando técnicas de preprocesamiento, aumentos de datos exigentes y una función de pérdida especializada.

Los resultados del experimento realizado muestran una buena precisión y un IoU excelente, evidenciando la capacidad del modelo para observar tanto clases mayoritarias como minoritarias, lo que resulta fundamental para su posible aplicación práctica.

Es interesante, sin embargo, que estos resultados se hayan logrado sin depender de componentes más modernos como bloques Transformer o preentrenamiento masivo, lo cual puede ser una de las mayores contribuciones de este PFG. Esto indica que, incluso partiendo de un modelo ligero, es posible obtener buenos valores mediante una configuración cuidadosamente diseñada y un tratamiento adecuado de los datos.

Sin embargo, para validar su uso en contextos clínicos o de enseñanza, sería necesario evaluar su desempeño en conjuntos de datos adicionales y más diversos, que permitan verificar tanto sus fortalezas como sus debilidades cuando se enfrenta a diferentes situaciones y escenarios anatómicos.

En pocas palabras, el modelo desarrollado ha superado con creces las expectativas iniciales, dando esperanza para implementarse como sistema oficial en entornos reales más exigentes, a falta de más validación mediante estudios ulteriores.

5.1. Objetivos logrados

5.1.1. Objetivo general

El objetivo general consistió en implementar y entrenar una CNN capaz de segmentar imágenes laparoscópicas con alta precisión, poniendo especial énfasis en la mejora de

la segmentación de clases minoritarias.

Este objetivo fue alcanzado satisfactoriamente mediante la implementación de una arquitectura *attention U-Net*, entrenada con una función de pérdida Focal Loss adaptativa y un sistema de aumentos de datos condicionales.

El modelo superó ampliamente las expectativas iniciales, logrando mejoras significativas en la segmentación de estructuras poco representadas, como el Conducto Cístico (IoU$= 0{,}7861$) y la Sangre (IoU$= 0{,}6766$), manteniendo un rendimiento excepcional en clases mayoritarias (con IoU superior al $95\,\%$).

5.1.2. Objetivos específicos

- **Seleccionar una arquitectura adecuada para la segmentación semántica laparoscópica.** Se eligió *attention U-Net*, por su capacidad para enfocar la atención en regiones anatómicamente relevantes. Esta decisión fue acertada, pues el modelo mostró gran eficacia tanto en clases mayoritarias como en estructuras complejas y de bajo contraste.
- **Preprocesar y adaptar el dataset CholecSeg8k, con especial atención a las clases minoritarias.** Se realizó un análisis exhaustivo de la distribución de clases, aplicando estrategias de sobremuestreo y normalización específicas para mejorar la representación de clases poco frecuentes.
- **Aplicar técnicas de aumento de datos condicionales para mejorar la generalización.** Se implementaron aumentos diferenciados para imágenes con clases minoritarias, incluyendo recortes centrados, rotaciones y transformaciones más agresivas en color y geometría.
- **Integrar una función de pérdida adaptativa para reforzar la segmentación de clases complejas.** La Focal Loss con parámetro gamma dinámico permitió al modelo ajustar progresivamente su foco hacia clases más difíciles durante el entrenamiento, equilibrando precisión en clases mayoritarias y sensibilidad en minoritarias.
- **Evaluar el rendimiento del modelo con métricas robustas y análisis por clase.** Fue evaluado mediante la función de pérdida, precisión por píxel, matriz de confusión e IoU por clase, evidenciando un rendimiento sobresaliente tanto globalmente como en estructuras pequeñas.
- **Analizar los resultados y su aplicabilidad en contextos clínicos o formativos.** El alto rendimiento alcanzado sugiere el potencial del modelo como herramienta

de apoyo en cirugía laparoscópica y simulación médica, mejorando la seguridad quirúrgica y la comprensión anatómica en entornos formativos.

Sin embargo, dado que la evaluación se realizó con un único conjunto de datos, sería conveniente validar el modelo en escenarios clínicos más variados. Asimismo, factores como variabilidad anatómica, diferencias en calidad de imagen o condiciones intraoperatorias podrían afectar su desempeño.

5.2. Impacto social y medioambiental

Clínica y socialmente hablando, el sistema presentado puede ayudar a que los pacientes estén más seguros durante un tratamiento quirúrgico, porque es capaz de evitar algunos errores humanos al identificar estructuras en una operación y ayudar a los profesionales de la salud mientras realizan acciones de precisión en poco tiempo durante una cirugía. Además, el hecho de que sea una arquitectura relativamente liviana podría permitir democratizar el acceso a tecnologías avanzadas, haciendo posible que centros médicos con pocos recursos computacionales lo utilicen, una hipótesis que debería confirmarse en escenarios reales futuros.

En cuanto al impacto medioambiental, el trabajo se desarrolló íntegramente en un entorno digital, sin consumo de materiales físicos (más allá del uso de un ordenador para el desarrollo). El entrenamiento y evaluación se llevaron a cabo en dos plataformas en la nube: Kaggle y Google Colab, cuyo impacto energético está asociado al uso de GPUs en centros de datos. No obstante, este consumo es reducido en comparación con el ciclo de vida de soluciones industriales, sin requerir hardware especializado ni infraestructuras locales de alto consumo.

Dado que es una solución basada en software con una arquitectura eficiente, su huella energética durante el uso debería ser moderada, especialmente si se integra en infraestructuras existentes optimizadas.

En términos generales, este proyecto presenta un impacto social positivo con posibles aplicaciones en contextos médicos reales, y de bajo impacto para el medio ambiente, convirtiéndolo en una solución tecnológica sostenible y con aplicaciones sociales que se pueden explotar.

5.3. Líneas futuras

Debido al tiempo limitado disponible, las restricciones de hardware y considerando el alcance del proyecto, hay varias formas en que este sistema podría llevarse al siguiente nivel:

- **Incorporación de técnicas de *Hard Example Mining***: permitiría centrar el aprendizaje en aquellas muestras que el modelo tiende a clasificar erróneamente, especialmente en estructuras pequeñas o visualmente ambiguas.
- **Aplicación de técnicas avanzadas de mezcla de imágenes**, como CutMix o MixUp, adaptadas al contexto de segmentación. Estas estrategias podrían enriquecer la diversidad del conjunto de entrenamiento y mejorar la capacidad de generalización del modelo.
- **Evaluación de arquitecturas más complejas**, como TransUNet o variantes de Swin-Unet, que podrían mejorar la captura de contexto global y las relaciones espaciales, aunque con un coste computacional más elevado.
- **Uso sistemático de aumento del tiempo de prueba (TTA)** durante la inferencia, lo cual permitiría combinar predicciones sobre múltiples versiones transformadas de la imagen para obtener resultados más precisos y estables.
- **Optimización para despliegue clínico**: evaluar y, si fuese necesario, adaptar el modelo a versiones más ligeras y eficientes que puedan ejecutarse en tiempo real, o integrarse en herramientas de apoyo quirúrgico con recursos computacionales limitados.

Estas líneas de trabajo representarían una continuación natural del PFG, que dado el tiempo y los recursos necesarios, puede llegar a convertirse en una herramienta de gran valor en el ámbito médico.

Referencias

[1] R. G. M. Vélez, M. G. M. Córdova, W. A. P. Zambrano y D. R. Z. Vera, «Comparación de técnicas quirúrgicas de cirugía abierta y cirugía por laparoscopia,» *RECIMUNDO: Revista Científica de la Investigación y el Conocimiento*, vol. 2, n.º 3, págs. 648-657, 2018.

[2] E. López e Y. Quijano, «Aspectos generales de cirugía laparoscópica,» *Servicio de Cirugía General y Digestiva, Hospital Madrid Norte Sanchinarro*, págs. 4-7, 2007.

[3] R. Mishra, «Laparoscopic cholecystectomy,» *nd Textbook of Practical Laparoscopic Surgery*, vol. 2, págs. 161-180, 2023.

[4] K. Guo, H. Tao, Y. Zhu et al., «Current applications of artificial intelligence-based computer vision in laparoscopic surgery,» *Laparoscopic, Endoscopic and Robotic Surgery*, vol. 6, n.º 3, págs. 91-96, 2023.

[5] L. Rouhiainen, «Inteligencia artificial,» *Madrid: Alienta Editorial*, págs. 20-21, 2018.

[6] C. F. Caiafa y S. E. Lew, «¿ Qué es la Inteligencia Artificial?,» 2020.

[7] SmartMind. «¿Cuáles son las ramas de la Inteligencia Artificial?» (2022), dirección: https://www.smartmind.net/blog/ramas-ia/.

[8] V. Dumoulin y F. Visin, «A guide to convolution arithmetic for deep learning,» *arXiv preprint arXiv:1603.07285*, 2016.

[9] I. D. Mienye, T. G. Swart, G. Obaido, M. Jordan y P. Ilono, «Deep Convolutional Neural Networks in Medical Image Analysis: A Review,» *Information*, vol. 16, n.º 3, pág. 195, 2025. DOI: 10.3390/info16030195.

[10] A. Tashtoush, Y. Wang, M. T. Khasawneh, A. Hader, M. S. Shazeeb y C. G. Lindsay, «Real-time object segmentation for laparoscopic cholecystectomy using YOLOv8,» *Neural Computing and Applications*, vol. 37, n.º 4, págs. 2697-2710, 2025.

[11] C. I. Nwoye, K. Elgohary, A. Srinivas, F. Zaid, J. L. Lavanchy y N. Padoy, *CholecTrack20: A Multi-Perspective Tracking Dataset for Surgical Tools*, 2025. arXiv: 2312.07352 [cs.CV]. dirección: https://arxiv.org/abs/2312.07352.

[12] P. Mascagni, D. Alapatt, A. Murali et al., «Endoscapes, a critical view of safety and surgical scene segmentation dataset for laparoscopic cholecystectomy,» *Scientific Data*, vol. 12, pág. 331, 2025. DOI: 10.1038/s41597-025-04642-4. dirección: https://doi.org/10.1038/s41597-025-04642-4.

[13] Y. Zhou, H. Badgery, M. Read, J. Bailey y C. E. Davey, «SurgicalSemiSeg: A Semi-Supervised Framework for Laparoscopic Image Segmentation,» en *Medical Imaging with Deep Learning*, 2025. dirección: https://openreview.net/forum?id=ozFfz2PCtT.

[14] A. Khan, Z. Rauf, A. R. Khan et al., *A Recent Survey of Vision Transformers for Medical Image Segmentation*, 2023. arXiv: 2312.00634 [eess.IV]. dirección: https://arxiv.org/abs/2312.00634.

[15] J. Chen, Y. Lu, Q. Yu et al., *TransUNet: Transformers Make Strong Encoders for Medical Image Segmentation*, 2021. arXiv: 2102.04306 [cs.CV]. dirección: https://arxiv.org/abs/2102.04306.

[16] H. Cao, Y. Wang, J. Chen et al., *Swin-Unet: Unet-like Pure Transformer for Medical Image Segmentation*, 2021. arXiv: 2105.05537 [eess.IV]. dirección: https://arxiv.org/abs/2105.05537.

[17] G. Litjens, T. Kooi, B. E. Bejnordi et al., «A survey on deep learning in medical image analysis,» *Medical image analysis*, vol. 42, págs. 60-88, 2017.

[18] O. Ronneberger, P. Fischer y T. Brox, «U-net: Convolutional networks for biomedical image segmentation,» en *Medical image computing and computer-assisted intervention–MICCAI 2015: 18th international conference, Munich, Germany, October 5-9, 2015, proceedings, part III 18*, Springer, 2015, págs. 234-241.

[19] O. Oktay, J. Schlemper, L. L. Folgoc et al., «Attention u-net: Learning where to look for the pancreas,» *arXiv preprint arXiv:1804.03999*, 2018.

[20] M. AlJabri, M. Alghamdi, F. Collado-Mesa y M. Abdel-Mottaleb, «Recurrent attention U-Net for segmentation and quantification of breast arterial calcifications on synthesized 2D mammograms,» *PeerJ Computer Science*, vol. 10, e2076, 2024. DOI: 10.7717/peerj-cs.2076. dirección: https://peerj.com/articles/cs-2076/.

[21] Y. Gao, Y. Jiang, Y. Peng, F. Yuan, X. Zhang y J. Wang, «Medical Image Segmentation: A Comprehensive Review of Deep Learning-Based Methods,» *Tomography*, vol. 11, n.º 5, 2025, ISSN: 2379-139X. DOI: 10.3390/tomography11050052. dirección: https://www.mdpi.com/2379-139X/11/5/52.

[22] C. Shorten y T. M. Khoshgoftaar, «A survey on Image Data Augmentation for Deep Learning,» *Journal of Big Data*, vol. 6, n.º 1, págs. 1-48, 2019. DOI: 10.1186/s40537-019-0197-0.

[23] L. Perez y J. Wang, «The Effectiveness of Data Augmentation in Image Classification using Deep Learning,» *arXiv preprint*, 2017. arXiv: 1712.04621 [cs.CV]. dirección: https://arxiv.org/abs/1712.04621.

[24] Y. Bengio, «Practical recommendations for gradient-based training of deep architectures,» *Neural networks: Tricks of the trade*, págs. 437-478, 2012.

[25] I. Loshchilov y F. Hutter, «SGDR: Stochastic Gradient Descent with Warm Restarts,» *arXiv preprint arXiv:1608.03983*, 2016. dirección: https://arxiv.org/abs/1608.03983.

[26] L. N. Smith, «Cyclical Learning Rates for Training Neural Networks,» en *2017 IEEE Winter Conference on Applications of Computer Vision (WACV)*, IEEE, 2017, págs. 464-472.

[27] M. Cabezas e Y. Diez, «An Analysis of Loss Functions for Heavily Imbalanced Lesion Segmentation,» *Sensors*, vol. 24, n.º 6, pág. 1981, 2024.

[28] T.-Y. Lin, P. Goyal, R. Girshick, K. He y P. Dollár, «Focal Loss for Dense Object Detection,» en *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2017, págs. 2980-2988.

[29] F. Milletari, N. Navab y S.-A. Ahmadi, «V-Net: Fully Convolutional Neural Networks for Volumetric Medical Image Segmentation,» en *Proceedings of the Fourth International Conference on 3D Vision (3DV)*, IEEE, 2016, págs. 565-571. DOI: 10.1109/3DV.2016.79.

[30] S. S. M. Salehi, D. Erdogmus y A. Gholipour, «Tversky loss function for image segmentation using 3D fully convolutional deep networks,» en *International workshop on machine learning in medical imaging*, Springer, 2017, págs. 379-387.

[31] S. Ioffe y C. Szegedy, «Batch normalization: Accelerating deep network training by reducing internal covariate shift,» en *International conference on machine learning*, pmlr, 2015, págs. 448-456.

[32] N. Bjork et al., «Understanding Batch Normalization,» en *NeurIPS Workshop on Deep Learning*, 2018.

[33] X. Li, S. Wang, J. Zhu y B. Tang, «Understanding the Disharmony Between Dropout and Batch Normalization by Variance Shift,» *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, págs. 2682-2690, 2019.

[34] F. Isensee, J. Petersen, S. A. Kohl, P. F. Jäger y K. H. Maier-Hein, «No New-Net,» *arXiv preprint arXiv:1809.10483*, 2018.

[35] Devopedia, *Confusion Matrix*, https://devopedia.org/confusion-matrix, ver. 6, ago. de 2019.

[36] M. A. Rahman e Y. Wang, «Optimizing Intersection-Over-Union in Deep Neural Networks for Image Segmentation,» en *Advances in Visual Computing*, G. Bebis, R. Boyle, B. Parvin et al., eds., Cham: Springer International Publishing, 2016, págs. 234-244, ISBN: 978-3-319-50835-1.

[37] A. A. Taha y A. Hanbury, «Metrics for evaluating 3D medical image segmentation: analysis, selection, and tool,» *BMC medical imaging*, vol. 15, págs. 1-28, 2015.

[38] W.-Y. Hong, C.-L. Kao, Y.-H. Kuo, J.-R. Wang, W.-L. Chang y C.-S. Shih, «Cholecseg8k: a semantic segmentation dataset for laparoscopic cholecystectomy based on cholec80,» *arXiv preprint arXiv:2012.12453*, 2020.

[39] GeeksforGeeks, *Getting Started with PyTorch*, https://www.geeksforgeeks.org/getting-started-with-pytorch/, 2025.

[40] C. Yu, J. Wang, C. Peng, C. Gao, G. Yu y N. Sang, «Learning a Discriminative Feature Network for Semantic Segmentation,» en *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018, págs. 1857-1866.

[41] R. Azad, E. K. Aghdam, A. Rauland et al., *Medical Image Segmentation Review: The success of U-Net*, 2022. arXiv: 2211.14830 [eess.IV]. dirección: https://arxiv.org/abs/2211.14830.

[42] N. K. Tomar, A. Shergill, B. Rieders, U. Bagci y D. Jha, *TransResU-Net: Transformer based ResU-Net for Real-Time Colonoscopy Polyp Segmentation*, 2022. arXiv: 2206.08985 [eess.IV]. dirección: https://arxiv.org/abs/2206.08985.

Índice de términos

Glosario

Attention Gate Mecanismo de atención que filtra selectivamente las activaciones del encoder antes de combinarlas con el decoder en arquitecturas tipo U-Net, permitiendo al modelo enfocarse en regiones relevantes. 11, 31

Backpropagation Algoritmo de entrenamiento que ajusta los pesos de una red neuronal propagando el error desde la salida hacia las capas anteriores. 36

Covariate Shift Fenómeno donde la distribución de las entradas de una red neuronal cambia durante el entrenamiento, afectando la convergencia y estabilidad del aprendizaje. 15, 16

CrossEntropy Función de pérdida ampliamente utilizada en clasificación, que mide la diferencia entre la distribución real y la distribución predicha por el modelo. 32

Hard Example Mining Técnica de entrenamiento que se enfoca en ejemplos difíciles o mal clasificados para mejorar el rendimiento del modelo, priorizando el aprendizaje en casos problemáticos. 51

Mask Denoising Autoencoders Arquitectura de red neuronal usada para aprendizaje semisupervisado que reconstruye datos originales a partir de entradas ruidosas o parcialmente ocultas, aplicados recientemente en imágenes laparoscópicas. 8

Synapse Conjunto de datos utilizado en segmentación multi-órgano a partir de imágenes de tomografía computarizada (TC), que incluye estructuras anatómicas como hígado, aorta, riñones o páncreas. Es ampliamente empleado como *benchmark* en tareas de segmentación médica tridimensional. 40

attention U-Net Variante de la arquitectura U-Net que incorpora mecanismos de atención para mejorar la segmentación de estructuras pequeñas o difíciles de distinguir. 11, 30, 31, 35, 41, 47, 49, IV

dataset Conjunto de datos utilizado para entrenar y evaluar modelos de aprendizaje automático. 2, 7, 19–21, 25, 28, 29, 35, 41, 42, 46, IV

dropout Técnica de regularización en redes neuronales que consiste en desactivar aleatoriamente algunas unidades durante el entrenamiento para reducir el sobreajuste y mejorar la generalización del modelo. 10, 31

ground truth Conjunto de datos de referencia que representa la etiqueta real contra la cual se comparan las predicciones del modelo. 18, 38

max pooling Operación de reducción espacial en redes neuronales convolucionales que selecciona el valor máximo dentro de una región específica, ayudando a extraer características relevantes y reducir dimensionalidad. 9, 31

softmax Función matemática que convierte un vector de valores en una distribución de probabilidad, donde la suma de todos los valores es igual a 1. Se usa habitualmente en la capa de salida de redes neuronales para tareas de clasificación multiclase. 31

watershed Máscara de segmentación generada mediante el algoritmo *watershed*, que separa regiones de una imagen imitando el flujo del agua sobre una superficie topográfica. Se utiliza para delimitar con precisión estructuras adyacentes o superpuestas. 20–23, 35

weight decay Técnica de regularización que penaliza la magnitud de los pesos en una red neuronal, añadiendo un término de decaimiento en la función de pérdida para prevenir el sobreajuste. 32, 35

`CholecTrack20` Conjunto de datos de seguimiento multiperspectiva para instrumentos quirúrgicos, compuesto por 20 vídeos quirúrgicos con más de $35{,}000$ fotogramas y $65{,}000$ anotaciones de herramientas. 7, 19

`Endoscapes2023` Conjunto de datos que incluye 201 vídeos de cirugías reales y más de $11{,}000$ imágenes anotadas, representando uno de los mayores esfuerzos de anotación manual en este ámbito. 7, 19

AdaGrad Algoritmo de optimización adaptativo que ajusta la tasa de aprendizaje para cada parámetro según el historial de gradientes, favoreciendo parámetros con gradientes pequeños y acelerando la convergencia. 32

Adam Algoritmo de optimización basado en el método de gradiente descendente estocástico, que combina las ventajas de AdaGrad y RMSProp mediante el cálculo

adaptativo de tasas de aprendizaje para cada parámetro, mejorando la velocidad y estabilidad del entrenamiento. 32, 35, 36

BKAI-IGH Dataset clínico para segmentación médica con imágenes anotadas, empleado para validar modelos de segmentación en contextos hospitalarios. 41

CholecSeg8k Conjunto de datos de más de 8.000 imágenes laparoscópicas con segmentaciones manuales de órganos y estructuras relevantes durante colecistectomías. 2, 19, 20, 25, 29, 35, 49, IV

Cityscapes Dataset de imágenes urbanas para tareas de segmentación semántica, diseñado para evaluar métodos en entornos de conducción autónoma y escenas urbanas complejas. 40

CutMix Método de aumento de datos que mezcla dos imágenes y sus etiquetas cortando y combinando regiones rectangulares, para mejorar la generalización y robustez del modelo. 51

ISIC 2018 Conjunto de datos de imágenes dermatológicas proporcionado por la *International Skin Imaging Collaboration*, utilizado para la segmentación de lesiones cutáneas en fotografías clínicas. 40

Kvasir-SEG Conjunto de datos público para segmentación de imágenes endoscópicas, utilizado en investigación médica para evaluar modelos de segmentación. 41

MixUp Técnica de aumento de datos que genera nuevas muestras mediante combinaciones lineales de pares de imágenes y sus etiquetas, promoviendo regularización y mejor aprendizaje. 51

PASCAL VOC 2012 Conjunto de datos público utilizado para la evaluación de algoritmos de segmentación y reconocimiento de objetos en visión por computador. 40

ReLU Función de activación lineal rectificada, definida como $\text{ReLU}(x) = \text{máx}(0, x)$, que introduce no linealidad en redes neuronales y ayuda a evitar el problema de desaparición del gradiente. 9, 31

RMSProp Algoritmo de optimización adaptativo que mantiene una media móvil de los cuadrados de los gradientes para ajustar dinámicamente la tasa de aprendizaje, mejorando la estabilidad y convergencia en problemas no estacionarios. 32

SegPC 2021 Conjunto de datos del desafío *Segmentation of Nuclei and Cytoplasm in Overlapping Cervical Cells*, centrado en la segmentación de núcleos y citoplasmas en imágenes microscópicas de células cervicales. 40

sobreajuste Fenómeno en el que un modelo de aprendizaje automático aprende en exceso los datos de entrenamiento, incluyendo el ruido o las particularidades de esos datos, lo que reduce su capacidad para generalizar a datos nuevos (*overfitting*). 10, 12, 16, 19, 30, 32–34

Transformer Arquitectura de red neuronal basada en mecanismos de atención que permite capturar relaciones de largo alcance en los datos. 8, 47

visión túnel Un síntoma oftalmológico que se manifiesta como una reducción progresiva del campo visual periférico, manteniendo la visión central intacta.. 2

Siglas

ANN Del inglés *artificial neural network*. 6, 9

BN Del inglés *batch normalization*. 15–17, 30, 31

CNN Del inglés *convolutional neural network*. 1, 2, 4, 6–8, 47

CPU Del inglés *central processing unit*, unidad central de procesamiento, es el componente principal de un ordenador que realiza las instrucciones de los programas. 33

CV Del inglés *computer vision*. 5, 7, 8

CVS Del inglés *critical view of safety*. 7

DL Del inglés *deep learning*. 4, 6, 27

FNR Del inglés *false negative rate*, tasa de falsos negativos. 17, 43, 44, IV

FPR Del inglés *false positive rate*, tasa de falsos positivos. 17, 43, 44, IV

GPU Del inglés *graphics processing unit*, unidad de procesamiento gráfico especializada en cálculos paralelos para acelerar renderizado y cómputo. 22, 27, 29, 33, 50

IA inteligencia artificial. 1, 2, 5, 6, III

IoU Del inglés *Jaccard index*, más conocido como *intersection over union*, es una métrica que evalúa la superposición entre la predicción del modelo y la anotación real, especialmente usada en tareas de segmentación.. 18, 33, 34, 38–42, 47, 49, III, IV

ML Del inglés *machine learning*. 6

PFG proyecto de fin de grado. 1, 2, 4, 19, 23, 30, 47, 51

PNG Formato de imagen conocido como *Portable Network Graphics*, que soporta compresión sin pérdida, usado para imágenes digitales. 20

RGB Del inglés *Red, Green, Blue*, modelo de color basado en la combinación aditiva de los colores rojo, verde y azul. 21–23, 31, 35

TTA Del inglés *test time augmentation*. 51