



UNIVERSIDAD NACIONAL DEL CENTRO DE LA PROVINCIA DE BUENOS AIRES

FACULTAD DE CIENCIAS EXACTAS

PROYECTO INTEGRADOR - INGENIERÍA DE SISTEMAS

**VISIÓN COMPUTACIONAL EN MEDICINA:
DESARROLLO DE UN MODELO DE SEGMENTACIÓN
BÁSICO (BASELINE) DE DISPOSITIVOS PARA
TRATAMIENTO DE ANEURISMAS CEREBRALES A
PARTIR DE IMÁGENES MÉDICAS 3D**

**David Burckhardt
Martín Vazquez Arispe**

**Director: Ignacio Larrabide
Co-directora: Romina Luciana Muñoz**

TANDIL - BS. AS. - ARGENTINA

2024

RESUMEN

En este trabajo se desarrolló un modelo de segmentación para dispositivos endosaculares luego del tratamiento de aneurismas cerebrales a partir de imágenes médicas 3D. Se empleó la arquitectura UNet, una red neuronal convolucional, para crear un modelo supervisado capaz de identificar y localizar estos dispositivos en las imágenes. Se realizó una revisión exhaustiva de literatura y antecedentes en visión computacional en medicina, utilizando herramientas como PyTorch, NumPy y Pandas para el desarrollo del modelo. Los datos utilizados fueron seleccionados manualmente por expertos en este tipo de imágenes. Este trabajo presenta una primera alternativa para el desarrollo de métodos de segmentación para este tipo de dispositivos, y busca mejorar la segmentación de imágenes para facilitar el diagnóstico y tratamiento de aneurismas cerebrales, capitalizando recursos disponibles y cultivando conocimientos en visión computacional médica.

ÍNDICE

RESUMEN.....	1
ÍNDICE.....	2
PREFACIO.....	4
INTRODUCCIÓN Y ANTECEDENTES.....	5
ANEURISMAS CEREBRALES.....	5
TRATAMIENTO.....	5
IMÁGENES MÉDICAS 3D.....	6
OBJETIVO Y CONTRIBUCIÓN.....	8
APRENDIZAJE PROFUNDO Y SU APLICACIÓN.....	9
TECNOLOGÍA.....	9
RED NEURONAL.....	9
FUNCIONAMIENTO DE UNA RED UNET.....	10
SUPERVISIÓN PROFUNDA.....	13
FUNCIONES DE ACTIVACIÓN.....	13
LEAKY RELU.....	13
SOFTMAX.....	14
NORMALIZACIÓN.....	14
Z-SCORE.....	15
HERRAMIENTAS COMPLEMENTARIAS.....	15
SERVIDORES DE PROCESAMIENTO.....	16
CÓDIGO.....	17
PARÁMETROS Y AJUSTES.....	17
TAMAÑO DE LOTE.....	17
NÚMERO DE ÉPOCAS.....	18
TAMAÑO DEL KERNEL.....	19
ITERACIONES POR ÉPOCA.....	19
TASA DE APRENDIZAJE INICIAL.....	19
PORCENTAJE DE SOBREMUESTREO DEL PRIMER PLANO.....	20
DECAIMIENTO DEL PESO.....	20
MULTIETIQUETA.....	20
AUMENTO DE DATOS.....	20
DATOS DE ENTRENAMIENTO.....	22
OBTENCIÓN Y ORIGEN.....	22
ORGANIZACIÓN.....	24
PLANIFICACIÓN Y PREPROCESAMIENTO.....	27
CONFIGURACIÓN DE DIRECTORIOS Y VARIABLES DE ENTORNO.....	27
PREPROCESAMIENTO Y MANEJO DE LA SALIDA.....	28
ACERCA DE LAS INFERENCIAS.....	30
MÉTRICAS CUANTITATIVAS Y SU IMPORTANCIA.....	30
ENTRENAMIENTOS.....	33
GRÁFICOS DE REFERENCIA.....	33

FUNCIONES DE PÉRDIDA Y PRECISIÓN DEL MODELO.....	33
DURACIÓN DE LAS ÉPOCAS.....	36
TASA DE APRENDIZAJE.....	36
ESTRATEGIA DE AJUSTE DE LA TASA DE APRENDIZAJE.....	36
VENTAJAS.....	36
MODELO 0.....	38
SOBREAJUSTE.....	40
MODELO 1.....	43
MODELO 2.....	44
RESUMEN ENTRENAMIENTOS 1.....	47
MODELO 5.....	48
REVISIÓN DE LO APRENDIDO.....	50
MODELO 10.....	51
MODELO 11.....	53
RESUMEN ENTRENAMIENTOS 2.....	56
MODELO 15.....	58
RESUMEN ENTRENAMIENTOS 3.....	63
 MODELO FINAL.....	65
 SERVICIO.....	68
SECCIONES.....	68
GUÍA DE USO.....	70
 APLICACIÓN DE CONOCIMIENTOS ACADÉMICOS.....	72
FUNDAMENTOS DE PROGRAMACIÓN Y ALGORITMOS.....	72
MATEMATICAS Y ESTADISTICA.....	72
PROCESAMIENTO DE IMÁGENES Y MANEJO DE DATOS.....	72
DESARROLLO DE SOFTWARE Y GESTIÓN DE PROYECTOS.....	72
INTELIGENCIA ARTIFICIAL.....	73
COMUNICACION Y GESTION DE SERVIDORES.....	73
 CONCLUSIONES Y TRABAJO FUTURO.....	74
 AGRADECIMIENTOS.....	75
 REFERENCIAS.....	76

PREFACIO

Este informe se presenta como parte integral de los requisitos para la realización de las Prácticas Profesionales Supervisadas (PPS) y Proyecto Integrador (PI), en cumplimiento con los lineamientos establecidos por la Universidad Nacional del Centro de la Provincia de Buenos Aires (UNICEN). El contenido aquí expuesto refleja los avances y logros obtenidos en el área de visión computacional en medicina, específicamente en el desarrollo de un modelo de segmentación de dispositivos para tratamiento de aneurismas cerebrales a partir de imágenes médicas 3D.

La investigación se desarrolló en el periodo comprendido entre el 13 de octubre de 2023 y el 25 de julio de 2024, en colaboración con el Instituto Pladema y el grupo Yatiris, un equipo de investigación dentro del instituto especializado en Medicina Asistida por Computadora. Durante este tiempo, el proyecto fue supervisado por el Dr. Ignacio Larrabide y la Dra. Romina Muñoz, quienes proporcionaron orientación y apoyo crítico para el éxito de este trabajo.

Cabe destacar que este informe no ha sido presentado con anterioridad para la obtención de ningún otro título en la UNICEN ni en ninguna otra institución académica.

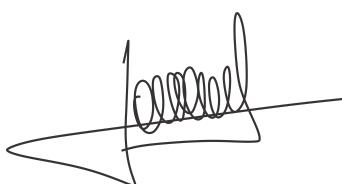
La estructura de este documento sigue las pautas establecidas para un PI, abarcando desde la introducción hasta las conclusiones, y se organiza de manera coherente para proporcionar una visión clara y completa de los esfuerzos realizados durante las PPS.

Se agradece a todas las personas e instituciones que contribuyeron de diversas maneras al desarrollo de este proyecto, brindando su conocimiento y experiencia. En particular, se reconoce la colaboración y asesoramiento del Dr. Ignacio Larrabide, la Dra. Romina Luciana Muñoz y la Dra. Camila García, cuya dedicación fue fundamental para el éxito de esta investigación.

Este informe representa el resultado del compromiso y la dedicación hacia la exploración y aplicación de técnicas avanzadas en el campo de la visión computacional en medicina, y se espera que contribuya de manera significativa al conocimiento y avance en esta área específica.



Firma de los estudiantes



Firma del Director



Firma de la Codirectora

INTRODUCCIÓN Y ANTECEDENTES

ANEURISMAS CEREBRALES

Los aneurismas cerebrales son anomalías caracterizadas por la dilatación anormal de una arteria en el cerebro, que puede llevar a un riesgo significativo de ruptura y hemorragia cerebral [3]. Estas condiciones pueden ser congénitas o desarrollarse a lo largo de la vida debido a factores como la hipertensión arterial, el tabaquismo o condiciones genéticas.

Un aneurisma cerebral presenta graves problemas clínicos, ya que su ruptura puede ocasionar una hemorragia subaracnoidea potencialmente mortal o daño neurológico permanente [9].

Históricamente, los aneurismas cerebrales han sido objeto de estudio y tratamiento. A lo largo de los años, diversas técnicas quirúrgicas y terapias menos invasivas han sido desarrolladas para abordar este desafío médico [14].

En la actualidad, la visión computacional y la inteligencia artificial ofrecen nuevas oportunidades para mejorar el diagnóstico y tratamiento de esta problemática. La aplicación de técnicas como la segmentación automática de imágenes médicas puede contribuir significativamente a la identificación temprana y precisa de estas condiciones, permitiendo una intervención más efectiva y segura [5].

TRATAMIENTO

El tratamiento de los aneurismas cerebrales ha experimentado una transformación notable gracias al desarrollo de tratamientos endovasculares haciendo uso de dispositivos intravasculares especializados. Actualmente, estas anomalías pueden abordarse de manera más segura y efectiva con dispositivos diseñados específicamente para reparar o proteger el aneurisma.

Uno de los enfoques principales involucra el uso de dispositivos especializados implantados de forma intravascular, como stents, bobinas de embolización e intrasaculares (como los WEBs, Woven EndoBridge), diseñados para fortalecer las arterias y ocluir el cuello del aneurisma con el objetivo de prevenir su ruptura. Los stents o endoprótesis tubulares de malla se implantan en la arteria afectada ocluyendo el cuello del aneurisma para así reforzar las paredes debilitadas del aneurisma y reducir este riesgo de ruptura. Por otro lado, las bobinas de embolización se introducen en el aneurisma para bloquear el flujo sanguíneo hacia él, lo que promueve la coagulación y la curación progresiva [6]. Estas intervenciones representan estrategias avanzadas y efectivas para el tratamiento de los aneurismas cerebrales, ofreciendo protección contra sus potenciales complicaciones.

El desarrollo y fabricación de estos dispositivos intrasaculares requiere una estrecha colaboración entre ingenieros biomédicos y profesionales de la salud. Los ingenieros aplican conocimientos médicos y habilidades de ingeniería para diseñar dispositivos seguros, duraderos y compatibles con el sistema vascular, contribuyendo así a mejorar los tratamientos disponibles.

El desarrollo del dispositivo intrasacular WEB, es un ejemplo del trabajo colaborativo entre profesionales de diferentes áreas para dar solución al tratamiento de aneurismas de

cuello ancho ubicados en bifurcaciones arteriales. Estos aneurismas representaban un desafío para ser tratados endosacularmente con stents o bobinas de embolización, debido al riesgo de migración del dispositivo o la oclusión parcial de una de las bifurcaciones [15].

El dispositivo WEB surgió como una opción de tratamiento más seguro para los aneurismas ubicados en bifurcaciones. Está disponible en diferentes tamaños y la elección del tamaño apropiado depende de las características morfológicas del aneurisma, las cuales se cuantifican usando imágenes 2D con una técnica que puede presentar fallas [16].

Con el objetivo de ofrecer una solución a la elección del tamaño de WEB apropiado para cada aneurisma se ha desarrollado una herramienta para simular computacionalmente la implantación de dispositivos intra-saculares (ID) utilizados en el tratamiento de este tipo de aneurismas intracraneales complejos. Esta herramienta permite calcular antes del tratamiento la altura final, la expansión radial y la porosidad del dispositivo dentro de la anatomía del paciente, utilizando imágenes angiográficas 3D, ofreciendo así una solución precisa y avanzada para mejorar la selección del tamaño del dispositivo antes del tratamiento [12].

El artículo [13] presenta una metodología avanzada para analizar los cambios morfológicos y de posición sufridos por el dispositivo WEB después del tratamiento de aneurismas intracraneales. Utilizando imágenes de angiografía rotacional tridimensional (3DRA), los investigadores estudiaron casos tratados con dispositivos WEB, comparando imágenes post-tratamiento y de seguimiento. Se segmentaron manualmente los dispositivos para obtener modelos 3D y se calcularon varias métricas, incluyendo volumen, área superficial, altura y diámetro máximo, así como cambios en la posición y dirección del dispositivo. Los resultados revelaron modificaciones significativas en la forma y posición de los dispositivos WEB, proporcionando nueva información crucial para entender y mejorar el tratamiento de los aneurismas intracraneales utilizando estos dispositivos intravasculares.

Estos avances han permitido tratamientos menos invasivos, reduciendo el tiempo de recuperación y mejorando los resultados para los pacientes afectados. La continua innovación en ingeniería biomédica cumple una función esencial en la mejora y desarrollo de dispositivos que salvan vidas y mejoran la calidad de vida de quienes padecen esta condición.

IMÁGENES MÉDICAS 3D

Las imágenes médicas 3D juegan un rol crucial en el tratamiento al proporcionar una representación tridimensional detallada de las estructuras cerebrales y los vasos sanguíneos [17]. Estas imágenes permiten una visualización completa del cerebro y sus vasos desde múltiples ángulos, lo que facilita a los médicos la evaluación precisa del tamaño, forma y ubicación de los aneurismas y otras lesiones cerebrovasculares. Esta información es de suma utilidad para planificar intervenciones quirúrgicas o procedimientos endovasculares, guiando con precisión el tratamiento y la colocación de dispositivos utilizados para reparar los vasos sanguíneos dañados.

Además, estas imágenes ofrecen una valiosa oportunidad para analizar la efectividad y la precisión de los dispositivos utilizados en el tratamiento. La capacidad de evaluar visualmente cómo interactúan los dispositivos con los vasos sanguíneos, su correcta posición y su efecto terapéutico es crítico para garantizar el éxito del tratamiento y prevenir complicaciones.

Las modalidades de imágenes médicas utilizadas para visualizar dispositivos en el cerebro incluyen la angiografía digital, la tomografía computarizada (TC) y la resonancia magnética (RM). La angiografía digital permite una representación dinámica en tiempo real de la interacción entre los dispositivos intravasculares y los vasos sanguíneos, mientras que la TC y la RM ofrecen imágenes detalladas en 3D para evaluar la posición y la integridad de los dispositivos [18].

Al observar cómo esta tecnología se comporta en el contexto anatómico específico del cerebro, los ingenieros pueden optimizar el diseño de los dispositivos y mejorar su eficacia y seguridad en futuras aplicaciones médicas.

OBJETIVO Y CONTRIBUCIÓN

En el contexto de los avances en el diagnóstico y tratamiento de aneurismas cerebrales mediante imágenes médicas y dispositivos especializados, surge la necesidad de automatizar y optimizar los procesos de segmentación para una intervención más rápida y efectiva. Este proyecto se enfoca en la creación, adaptación y ajuste de un modelo de línea base utilizando una red neuronal convolucional preexistente, específicamente una UNet.

El objetivo principal es el desarrollo de un modelo de segmentación automática de dispositivos para el tratamiento de aneurismas cerebrales a partir de imágenes médicas 3D de pacientes tratados por esta anomalía. Indagando en la visión computacional aplicada a la medicina, se busca automatizar un proceso que, hasta la fecha, se sigue realizando de manera manual por radiólogos y especialistas.

La contribución más innovadora de este trabajo radica en que no existe actualmente una herramienta específica similar capaz de segmentar dispositivos como los stents o dispositivos WEB. Pese a los avances en inteligencia artificial, la segmentación manual de estos dispositivos continúa siendo un proceso relativamente reciente y utilizado en la práctica clínica. Sin embargo, esta metodología manual, que se realiza de manera visual y subjetiva por parte de los especialistas, es propensa a la variabilidad, consume mucho tiempo y requiere un alto grado de precisión, lo que subraya la necesidad de soluciones más eficientes.

Este proyecto marca un punto de inflexión al ofrecer una alternativa automatizada inédita, que no solo reemplaza la labor manual, sino que promete reducir significativamente el tiempo de análisis y mejorar la precisión en la detección de estos dispositivos. Al desarrollar este modelo de segmentación automática, se busca acelerar y estandarizar el proceso de evaluación, mejorando así la planificación quirúrgica y permitiendo intervenciones más rápidas y seguras. La originalidad y relevancia del enfoque residen en que, hasta la fecha, no se ha desarrollado nada similar, lo que convierte a este trabajo en una contribución valiosa e innovadora en el campo del tratamiento de aneurismas cerebrales.

APRENDIZAJE PROFUNDO Y SU APLICACIÓN

El aprendizaje profundo (*deep learning*), es un subcampo de la inteligencia artificial (IA) y el aprendizaje automático (*machine learning*) que se centra en el uso de redes neuronales profundas para resolver problemas complejos. Estas redes, compuestas por múltiples capas de neuronas artificiales, aprenden representaciones jerárquicas de datos, capturando características cada vez más abstractas y complejas a medida que se avanza por las capas.

Dentro del ámbito de la IA, el aprendizaje profundo se clasifica como una técnica avanzada de aprendizaje automático. Mientras que la IA abarca una amplia gama de enfoques, desde sistemas expertos hasta procesamiento de lenguaje natural y visión por computadora, el aprendizaje automático se enfoca en algoritmos que pueden aprender y mejorar a partir de datos. El deep learning se distingue por su capacidad para manejar grandes volúmenes de datos y aprender características complejas de forma automática.

Se puede implementar en dos modalidades principales: supervisado y no supervisado. En el aprendizaje supervisado, los modelos se entrena con conjuntos de datos etiquetados, donde cada entrada tiene una salida asociada. Este enfoque es óptimo para tareas como clasificación de imágenes, reconocimiento de voz y predicción de valores, ya que la red neuronal aprende a realizar un mapeo preciso entre entradas y salidas mediante el ajuste de sus parámetros para minimizar el error entre las predicciones y las etiquetas verdaderas.

Por otro lado, el aprendizaje no supervisado prescinde de datos etiquetados y se enfoca en descubrir patrones y estructuras ocultas en conjuntos de datos sin guías explícitas. Este enfoque es útil en tareas como clustering (agrupamiento de datos similares), reducción de dimensionalidad y detección de anomalías. Las redes neuronales en aprendizaje no supervisado tienen la capacidad de identificar relaciones y agrupaciones inherentes en los datos, lo que permite obtener información valiosa y realizar descubrimientos significativos sin la necesidad de intervención humana previa.

En este trabajo, se utilizó aprendizaje automático supervisado con datos etiquetados para entrenar modelos de segmentación eficaces. Estos datos fueron fundamentales para guiar el entrenamiento y optimización de los modelos, logrando resultados precisos.

TECNOLOGÍA

RED NEURONAL

Para alcanzar la meta propuesta en este proyecto, se empleó la red neuronal nnUNet, una poderosa herramienta diseñada específicamente para la segmentación automática de imágenes médicas en 3D. Desarrollada y mantenida por el *Applied Computer Vision Lab* (ACVL) de la organización *Helmholtz Imaging* y la *Division of Medical Image Computing at the German Cancer Research Center* (DKFZ), nnUNet se basa en la arquitectura de redes neuronales convolucionales profundas y está implementada utilizando el framework de aprendizaje profundo PyTorch [1].

La arquitectura de nnUNet se caracteriza por ser altamente modular y adaptable, permitiendo la integración de diferentes componentes y estrategias de entrenamiento según

las necesidades específicas del problema. Incorpora capas convolucionales profundas junto con técnicas de aumento de datos y regularización para mejorar la generalización y la robustez del modelo. Ofrece una amplia gama de opciones predefinidas y personalizables para la configuración de las diferentes etapas, incluyendo la elección de funciones de pérdida, optimizadores y estrategias de entrenamiento. La herramienta genera varias configuraciones de UNet específicamente adaptadas a cada conjunto de datos:

- **2d**: Una UNet en 2D, aplicable a conjuntos de datos 2D.
- **3d_fullres**: Una UNet en 3D diseñada para operar a alta resolución de imagen.
- **3d_lowres → 3d_cascade_fullres**: Una UNet en 3D en cascada, donde una primera UNet opera en imágenes de baja resolución seguida de una segunda UNet en alta resolución que refina las predicciones de la primera. Esta configuración es particularmente útil para conjuntos de datos 3D con imágenes de gran tamaño.

Durante el desarrollo de este trabajo, se seleccionó la configuración 3D fullres de nnUNet para garantizar la preservación de detalles importantes en las imágenes médicas. Esta elección fue especialmente beneficiosa gracias al acceso a la tecnología avanzada del servidor de cálculo de Pladema, que proporcionó la capacidad de procesamiento necesaria para trabajar con imágenes de alta resolución sin comprometer la calidad ni la precisión de los resultados. En conjunto, esta decisión permitió aprovechar al máximo la capacidad de nnUNet para segmentar imágenes médicas detalladas y complejas, demostrando así su eficacia en aplicaciones clínicas y de investigación.

FUNCIONAMIENTO DE UNA RED UNET

La arquitectura UNet es una red neuronal convolucional diseñada específicamente para tareas de segmentación de imágenes. Esta red se caracteriza por su estructura en forma de "U" que combina las etapas de codificación (*encoder*) y decodificación (*decoder*) para lograr una segmentación precisa y detallada.

En la fase de codificación de la arquitectura, la imagen de entrada se somete a un proceso de extracción progresiva de características mediante capas convolucionales y de pooling. Inicialmente, las capas convolucionales aplican filtros para detectar patrones locales como bordes, texturas o formas en diferentes partes de la imagen. Conforme avanzamos en las capas convolucionales, estos filtros aprenden a representar características más abstractas y complejas debido a la estructura jerárquica de la red neuronal. Cada operación de convolución es seguida por una función de activación no lineal, como ReLU, que introduce no linealidad y permite que la red aprenda relaciones más sofisticadas entre las características extraídas.

Posteriormente, las capas de pooling entran en juego para reducir la resolución espacial de los mapas de características, conservando las más relevantes y robustas. Esta reducción de dimensionalidad controla el sobreajuste y mejora la eficiencia computacional de la red al eliminar detalles menos importantes. Además, el uso de capas de pooling hace que la

red sea invarianta a pequeñas traslaciones y distorsiones en las características detectadas, proporcionando una representación más generalizada de la imagen.

En el contexto del entrenamiento de redes neuronales, es crucial comprender el concepto de gradiente y su papel en el proceso de retropropagación (backpropagation). El gradiente indica la dirección y magnitud del cambio en la función de pérdida con respecto a los parámetros de la red. La función de pérdida mide la discrepancia entre las predicciones de la red y los valores reales, guiando el proceso de aprendizaje. Durante la retropropagación, este gradiente se calcula utilizando la regla de la cadena y se utiliza para ajustar los pesos de la red con el objetivo de minimizar la pérdida. Sin embargo, en redes profundas, puede surgir el desafío del desvanecimiento del gradiente, donde los gradientes se vuelven demasiado pequeños a medida que se propagan hacia atrás a través de múltiples capas, dificultando el aprendizaje efectivo. Para abordar este problema, UNet utiliza conexiones residuales que facilitan un flujo de información más estable a lo largo de la red durante la retropropagación. Estas conexiones permiten que los gradientes se propaguen de manera más eficiente, manteniendo su magnitud y permitiendo que las capas más profundas contribuyan de manera significativa al proceso de optimización.

En la fase de decodificación, se emplean capas de muestreo ascendente (*upsampling*) para reconstruir las características codificadas en una representación detallada de la imagen segmentada. Estas capas de decodificación aumentan la resolución espacial de las características previamente extraídas durante la fase de codificación. Utilizando técnicas como la convolución transpuesta o la interpolación bilineal, el muestreo ascendente amplía la información para reconstruir una versión más detallada de la imagen. De esta manera se recuperan detalles finos y estructuras importantes que podrían haberse perdido durante el muestreo descendente (*downsampling*) en la fase de codificación. Al aumentar la resolución espacial de las características, se logra una representación más precisa y fiel a la imagen original.

Además en esta fase se realiza la concatenación de características. Esto implica combinar las características de las capas de codificación con las de las capas de decodificación. La concatenación permite fusionar información de alto nivel (contextual y semántica) proveniente de las capas de codificación, con detalles de bajo nivel (texturas y estructuras) de las capas de decodificación.

La salida final de la red es una máscara de segmentación que indica la pertenencia de cada píxel a una determinada clase o categoría. La capa de salida utiliza una función de activación apropiada según el tipo de problema de segmentación. A continuación se adjunta una imagen que muestra la arquitectura de la red utilizada (ver figura 1).

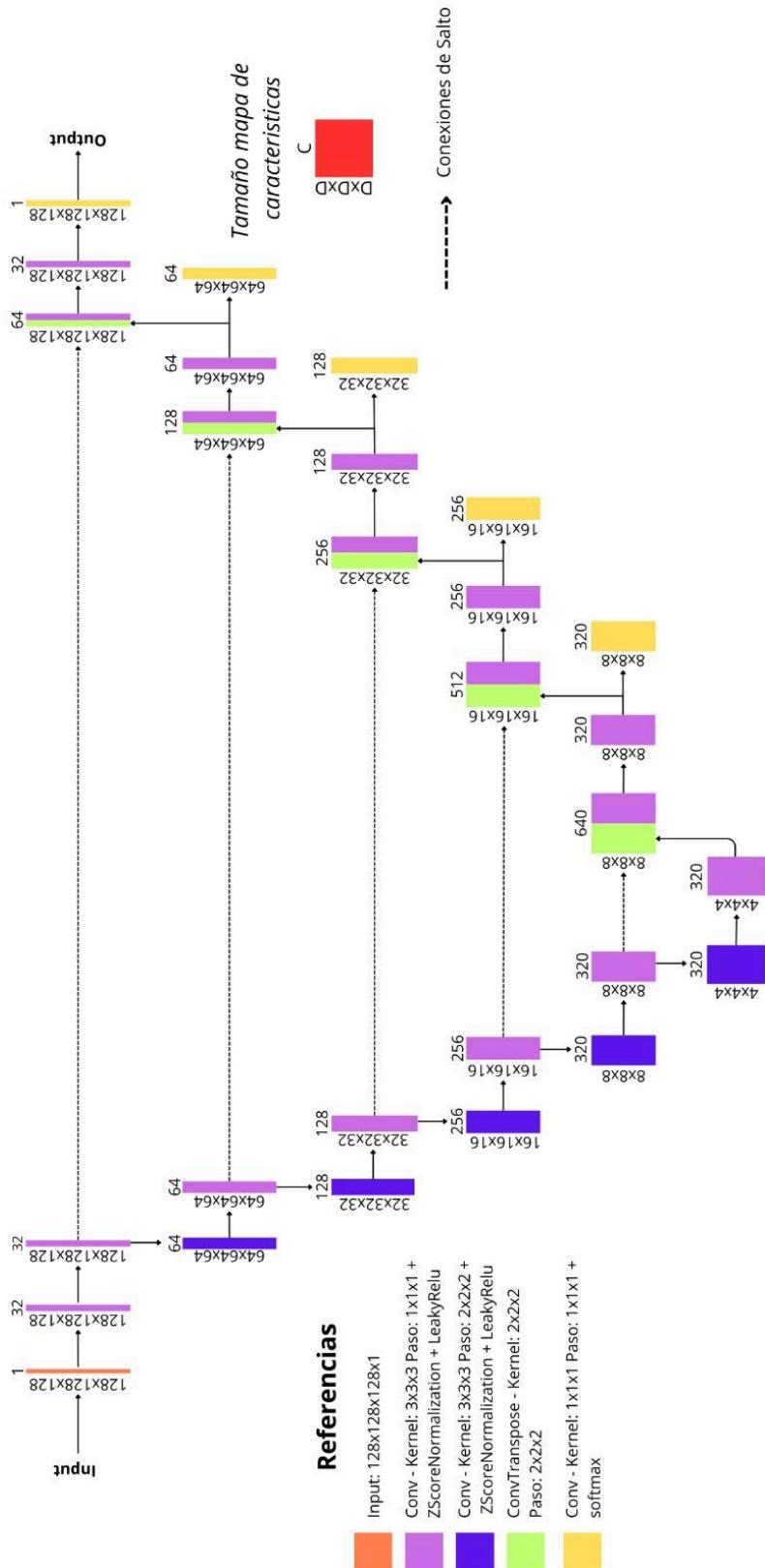


Figura 1: Gráfico de la arquitectura de la red utilizada.

Esta implementación se compone de 6 etapas de codificación y 5 etapas de decodificación. La primera etapa de codificación está comprendida de 2 convoluciones, ambas con un kernel de 3x3x3 y un paso (*stride*) de 1x1x1 donde la normalización utilizada es Z-score y la función de activación es Leaky Relu (ambas conceptos se abordan más adelante). Luego, en las 5 etapas posteriores del codificador se realizan nuevamente 2 capas de convolución, la primera de igual modo aplica un kernel de 3x3x3 pero con un paso de 2x2x2, y la segunda, a diferencia de la primera emplea un paso de 1x1x1.

En la fase de decodificación cada etapa se compone de una convolución traspuesta con un kernel y paso de 2x2x2, más la concatenación correspondiente con la salida del mismo nivel del codificador, a esto se le suma dos convoluciones con un filtro de 3x3x3 y un paso 1x1x1 con la misma configuración que antes (LeakyRelu y Z-score). Finalmente se emplea la técnica supervisión profunda, para que se pueda ejecutar dicho método se anexa una nueva capa de convolución con un kernel de 1x1x1, un paso de 1x1x1 y una función de activación Softmax. Es importante resaltar que la entrada de la siguiente fase de la decodificación es la salida de la segunda convolución y no es de la última capa que solamente es utilizada para la supervisión profunda. Seguidamente se hará una descripción de las técnicas anteriormente mencionadas.

SUPERVISIÓN PROFUNDA

La supervisión profunda (*deep supervision*) es una técnica utilizada en redes neuronales profundas para mejorar el entrenamiento y la convergencia. En lugar de aplicar la adición función de pérdida solo en la salida final de la red, esta técnica introduce funciones de pérdida adicionales en capas intermedias. Esto proporciona gradientes más útiles y evita el problema del desvanecimiento del gradiente.

En una UNet, se implementa supervisión profunda añadiendo salidas auxiliares en las capas de resolución intermedia. Cada salida auxiliar tiene su propia función de pérdida y se combina con la pérdida final. Esto guía el aprendizaje en diferentes niveles de la red, mejorando la precisión y la estabilidad del entrenamiento.

FUNCIONES DE ACTIVACIÓN

Las funciones de activación son funciones matemáticas que se aplican a la salida de cada neurona en una capa de la red. Su propósito es introducir no linealidad en la red, lo que permite a la red aprender y modelar relaciones complejas en los datos.

LEAKY RELU

La función Leaky Relu es una variante de ReLU que permite un pequeño gradiente cuando la entrada es negativa (ver figura 2). Esto ayuda a mitigar el problema de las "neuronas muertas" en las redes neuronales. La ecuación se detalla abajo.

$$\text{LeakyRelu}(x) = \max(kx, x), \text{ donde } 0 \leq k \leq 1$$

En el trabajo se utilizó un $k = 0.01$, dado que es un buen compromiso para tener un pequeño gradiente y mantener la no linealidad. Sin embargo, no es el único posible; otros valores cercanos también pueden funcionar, pero 0.01 es una elección comúnmente aceptada y probada empíricamente.

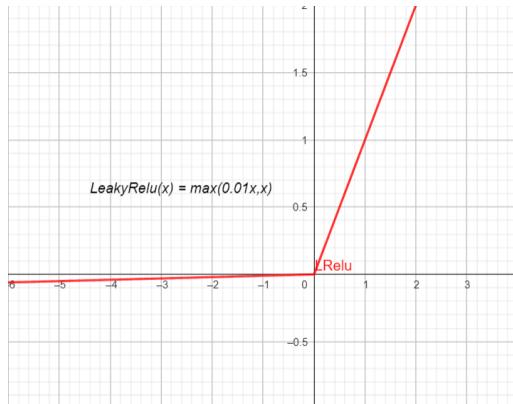


Figura 2: Gráfica de la función de activación Leaky Relu utilizada en el trabajo.

SOTMAX

La función Softmax convierte un vector de valores en probabilidades, son útiles en clasificación multiclas. Asigna a cada clase una probabilidad entre 0 y 1, que suma 1. La ecuación de la misma se ve a continuación.

$$\text{Softmax}(x_i) = \frac{e^{x_i}}{\sum_{j=1}^K e^{x_j}}$$

Parámetros:

- $\text{Softmax}(x_i)$: Es la i -ésima componente del vector de salida después de aplicar la función softmax
- x : Es el vector de entrada de puntuaciones
- x_i : Es la i -ésima componente del vector de entrada x
- K : Es el número de clases

NORMALIZACIÓN

La normalización de datos es un proceso utilizado en el preprocesamiento de datos para ajustar la escala de las características dentro de un rango específico o distribución. Su objetivo principal es hacer que los datos sean más uniformes y comparables entre sí, lo que puede facilitar el análisis y mejorar el rendimiento de los algoritmos de aprendizaje automático.

Z-SCORE

La técnica Z-score transforma los datos para que tengan una media de 0 y una desviación estándar de 1. Esto facilita la comparación entre diferentes datasets y mejora el rendimiento de algoritmos de aprendizaje automático. Al centrar los datos y escalarlos según su variabilidad, se eliminan las unidades, permitiendo que distintas características contribuyan equitativamente al análisis. La ecuación del Z-score de un valor x se traduce en la siguiente ecuación.

$$z = \frac{x - \mu}{\sigma}$$

Donde:

- x : Es el valor a normalizar.
- μ : Es la media del conjunto de datos.
- σ : Es la desviación estándar del conjunto de datos.

HERRAMIENTAS COMPLEMENTARIAS

Se han empleado diversas herramientas tecnológicas para gestionar eficazmente el desarrollo y la implementación del enfoque de segmentación automática de dispositivos endosaculares. Se utilizó GitHub como plataforma central para el control de versiones del código, permitiendo mantener un registro organizado y colaborativo de los cambios realizados. Además, se creó un repositorio propio que contiene una copia de la nnUNet, y sirve para mantener los cambios, parámetros y ajustes de la red de manera específica para este proyecto en concreto [2]. También se desarrolló otro repositorio privado para mantener un backup de los modelos entrenados, asegurando la disponibilidad y la integridad de los resultados obtenidos.

Python fue el lenguaje principal utilizado, aprovechando librerías como PyTorch, NumPy y Pandas. PyTorch facilitó la definición y entrenamiento de redes neuronales, mientras que NumPy y Pandas fueron determinantes para el análisis y procesamiento de datos. Para la visualización y procesamiento de imágenes médicas, se emplearon herramientas como Slicer y ParaView, que ofrecen capacidades avanzadas en análisis visual de datos en 3D. Se adjunta como ejemplo de uso de las herramientas, una imagen que representa una segmentación del árbol arterial obtenida de la imagen original de uno de los pacientes junto a la segmentación del dispositivo endosacular (amarillo) usado para tratar el aneurisma (ver figuras 3 y 4). Por último, se utilizó Visual Studio Code como entorno de desarrollo integrado (IDE) principal, proporcionando funcionalidades avanzadas de edición de código y depuración.

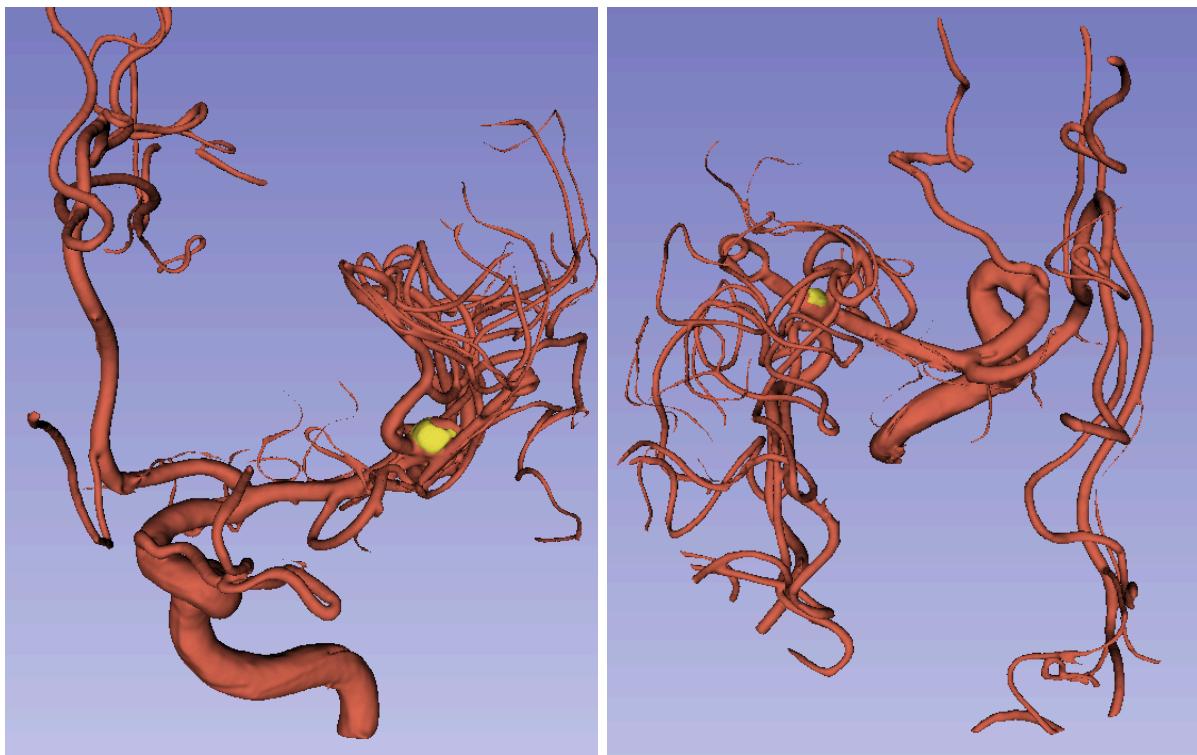


Figura 3: Vista anterior del árbol arterial con segmentación resaltada en amarillo del paciente 19 (P019).

Figura 4: Vista superior del árbol arterial con segmentación resaltada en amarillo del paciente 19 (P019).

En conjunto, estas herramientas complementaron el desarrollo, implementación y validación del enfoque de segmentación automática de dispositivos endosaculares, aprovechando el potencial del aprendizaje automático y la visión computacional en medicina.

SERVIDORES DE PROCESAMIENTO

Para realizar trabajos de esta magnitud, que involucran entrenamientos de alto procesamiento de datos, se requiere una infraestructura sólida y de gran capacidad. En Pladema, cuentan con un total de 6 GPUs (Unidades de Procesamiento Gráfico), las cuales desempeñan un papel crucial en tareas intensivas como el preprocesamiento, entrenamiento de modelos de aprendizaje automático y realización de inferencias en imágenes médicas. Esta capacidad de procesamiento es esencial para impulsar la investigación médica computacional, permitiendo el análisis eficiente de grandes volúmenes de datos médicos. La presencia de estas GPUs en la infraestructura tecnológica demuestra el compromiso de Pladema en proporcionar recursos avanzados para facilitar el avance en la ciencia y la medicina basada en datos.

En el entorno de Pladema, la gestión eficiente de los servidores se realiza utilizando SLURM [8] (Simple Linux Utility for Resource Management), un sistema altamente utilizado en entornos de cómputo de alto rendimiento. SLURM permite coordinar y distribuir recursos computacionales entre múltiples usuarios de manera efectiva. Este sistema es esencial para optimizar la utilización de los recursos de hardware, como CPUs y GPUs, garantizando un acceso equitativo entre los investigadores de la institución. Los usuarios interactúan con

SLURM a través de comandos como *sbatch* para enviar trabajos al sistema de colas, *squeue* para monitorear el estado de las tareas en ejecución, *scancel* para cancelar trabajos en cola, entre otros. Este acceso basado en comandos facilita una organización eficaz y equitativa de los recursos informáticos, fomentando la colaboración y el intercambio entre los investigadores de Pladema. La implementación de SLURM como sistema de gestión de recursos es fundamental para maximizar la eficiencia y el rendimiento en entornos de cómputo de alto rendimiento como el utilizado en esta investigación.

CÓDIGO

Para utilizar y ajustar los parámetros de la red neuronal convolucional (CNN) en este proyecto, se empleó una clase proporcionada por la misma red llamada *nnUNetTrainer*. Este es el entrenador por defecto para la nnU-Net, diseñado para proporcionar todo lo necesario para su funcionamiento básico y efectivo. Incluye la configuración estándar y los componentes esenciales para entrenar la red de manera eficiente. Al heredar de esta clase, se tuvo acceso a la capacidad de editar diversos parámetros que se ajustaron durante el proceso de entrenamiento.

La clase *nnUNetTrainer* facilitó la configuración y optimización de varios aspectos de la red, incluidos los hiper parámetros como la tasa de aprendizaje, el tamaño del lote (*batch size*), el número de épocas de entrenamiento, las funciones de pérdida (*loss functions*), y los métodos de regularización, entre otros. Estos parámetros se modificaron iterativamente durante el entrenamiento para optimizar el desempeño del modelo en términos de precisión y otras métricas de evaluación. Además, se incorporaron otros ajustes inspirados en el curso "Redes Neuronales Convolucionales" de Stanford University en Coursera, que proporcionaron perspectivas valiosas sobre cómo optimizar y ajustar modelos de redes neuronales para aplicaciones médicas. [4]

PARÁMETROS Y AJUSTES

Además, la capacidad de heredar y modificar esta clase permitió experimentar con diferentes configuraciones y estrategias de entrenamiento para adaptarse mejor a las características específicas del conjunto de datos y del problema médico en cuestión. Esto fue determinante para lograr mejoras significativas en la capacidad predictiva y el rendimiento general del modelo. A continuación se desarrollan algunos parámetros ajustados en los entrenamientos.

TAMAÑO DE LOTE

El tamaño del lote en el contexto del entrenamiento de modelos de aprendizaje automático es un parámetro que influye directamente en la eficiencia y la capacidad de generalización del modelo. Este tamaño se refiere al número de ejemplos de entrenamiento utilizados en cada iteración durante el proceso de entrenamiento.

El ajuste del tamaño del lote depende de varios factores, incluido el tamaño del conjunto de datos, la arquitectura de la red neuronal, la potencia de la GPU y la naturaleza específica de la tarea de aprendizaje automático. Utilizar un tamaño de lote grande puede

resultar en un entrenamiento más eficiente, ya que la red neuronal puede aprender de más datos simultáneamente, lo que puede llevar a una mejora en la precisión del modelo. Sin embargo, un tamaño de lote grande también puede requerir más memoria y potencia de procesamiento, y puede aumentar el riesgo de sobreajuste, donde el modelo se adapta demasiado a los detalles específicos de los datos de entrenamiento y no generaliza bien a nuevos datos.

Por otro lado, un tamaño de lote pequeño puede ayudar a prevenir el sobreajuste al obligar a la red neuronal a aprender de menos datos a la vez. Aunque esto puede ser beneficioso, un tamaño de lote pequeño también puede resultar menos eficiente en términos de tiempo de entrenamiento, ya que la red neuronal necesita aprender de los mismos datos repetidamente.

En términos prácticos, se recomienda comenzar con un tamaño de lote grande y luego ajustarlo según sea necesario. Si se observa sobreajuste o problemas de generalización, se reduce el tamaño del lote y se vuelve a experimentar. Además, probar diferentes tamaños de lote durante el proceso puede ayudar a determinar cuál funciona mejor para el conjunto de datos y la tarea específica que se está abordando.

En el caso específico del proyecto, se inició el entrenamiento con un tamaño de lote pequeño, utilizando solo 2 parches por lote. Los parches tienen un tamaño de 128x128x128 voxels, al manejar porciones de esta dimensión, en lugar de imágenes completas, se reduce el uso de memoria, pero el tamaño del lote debe mantenerse pequeño para evitar sobrecargar la GPU. Este enfoque permite actualizaciones más frecuentes del modelo, lo que puede mejorar su ajuste a los gradientes y favorecer una convergencia más rápida. De todas formas, es importante tener cuidado con el sobreajuste, dado el tamaño reducido del lote. Un tamaño de lote mayor podría reducir este riesgo al proporcionar más variabilidad en los datos de entrenamiento, lo que permitiría a la red neuronal generalizar mejor.

NÚMERO DE ÉPOCAS

Una época (*epoch*) en el contexto del entrenamiento de modelos de aprendizaje automático representa una iteración completa sobre el conjunto de datos de entrenamiento. Durante una época, el modelo procesa todos los ejemplos en el conjunto de datos de entrenamiento una vez y ajusta sus parámetros en función de la función de pérdida calculada. Cada iteración dentro de una época implica el procesamiento secuencial de cada ejemplo individual en el conjunto de datos, seguido de la actualización de los pesos del modelo utilizando técnicas como el descenso del gradiente.

El ajuste de la cantidad de épocas durante el entrenamiento de un modelo es una consideración clave que afecta tanto la eficiencia temporal como la convergencia del modelo. Reducir el número de épocas puede acelerar significativamente el proceso de entrenamiento, lo cual es beneficioso cuando se realizan iteraciones rápidas para ajustar hiperparámetros y explorar diferentes configuraciones. Esta eficiencia temporal es especialmente valiosa en entornos de investigación y desarrollo donde se requiere una rápida experimentación.

Por otro lado, es fundamental evaluar si el modelo ha convergido adecuadamente en el número de épocas seleccionado. La convergencia del modelo se refiere a la estabilización

de las métricas de entrenamiento y validación, indicando que el modelo ha aprendido lo suficiente de los datos disponibles. En algunos casos, un número menor de épocas puede resultar en un modelo subentrenado que no captura todas las características relevantes de los datos. Por lo tanto, es importante monitorear de cerca las métricas de rendimiento durante el entrenamiento para determinar si el modelo ha alcanzado su mejor rendimiento o si se requieren más épocas para lograr una convergencia adecuada.

TAMAÑO DEL KERNEL

El tamaño de kernel en el contexto del procesamiento de imágenes se refiere al tamaño de la región de la imagen que se toma como muestra durante el proceso de entrenamiento o inferencia de un modelo de aprendizaje automático.

El ajuste del tamaño de kernel es fundamental ya que influye directamente en cómo se capturan y representan las características locales de la imagen. Un tamaño de kernel más pequeño permite capturar detalles finos y características locales precisas en la imagen, mientras que un tamaño de kernel más grande puede tomar mayores regiones de la imagen, capturando características más globales pero perdiendo detalles finos.

El tamaño del kernel puede influir en la capacidad del modelo para aprender y generalizar patrones. Un tamaño de kernel más pequeño puede llevar a modelos con una mayor capacidad de representación de detalles finos, lo que puede ser beneficioso para tareas donde las características locales son importantes, como la detección de bordes o texturas. Por otro lado, un kernel más grande puede ser útil para capturar características más globales y estructurales de la imagen, como la forma general de un objeto o la distribución de ciertas características.

En el contexto del proyecto, se optó por un tamaño de kernel más pequeño debido a la necesidad de detectar estructuras relativamente pequeñas y detalladas en las imágenes médicas. Esta elección facilitó que el modelo capturara con mayor precisión características locales importantes, como bordes, texturas o detalles anatómicos referentes a los aneurismas y dispositivos de tratamiento, que eran cruciales para la tarea de segmentación de imágenes médicas.

ITERACIONES POR ÉPOCA

El número de iteraciones por época determina cuántas veces se procesa el conjunto de datos de entrenamiento completo durante una única época. Esta configuración afecta directamente la cantidad de actualizaciones de peso que ocurren en cada época, influyendo en la convergencia del modelo. En este proyecto, se estableció un valor de 250 iteraciones por época para controlar la cantidad de procesamiento y optimizar la eficiencia del entrenamiento.

TASA DE APRENDIZAJE INICIAL

La tasa de aprendizaje inicial, definida como la magnitud de los ajustes aplicados a los pesos del modelo durante el entrenamiento, es un factor crítico que influye en la convergencia y estabilidad del modelo de deep learning. Una tasa de aprendizaje más alta

puede acelerar el proceso de aprendizaje, pero también puede llevar a una convergencia inestable o a oscilaciones en la optimización del modelo.

En este proyecto, se seleccionó una tasa de aprendizaje inicial de 0.01 como punto de partida para controlar la velocidad de aprendizaje y mejorar la convergencia del modelo sobre el conjunto de datos.

PORCENTAJE DE SOBREMUESTREO DEL PRIMER PLANO

Este parámetro controla la proporción de muestras del primer plano (*foreground*) en relación con el fondo (*background*) durante el entrenamiento. Por ejemplo, un valor de 0.33 indica que aproximadamente el 33% de las muestras son del primer plano. Este ajuste es crucial para equilibrar la representación de diferentes clases y mejorar la capacidad del modelo para segmentar correctamente las áreas de interés en las imágenes médicas.

DECAIMIENTO DEL PESO

El decaimiento del peso (*weight decay*) es una técnica de regularización que penaliza los valores grandes de los pesos en la función de pérdida. Esto ayuda a prevenir el sobreajuste al desalentar la creación de modelos demasiado complejos que se ajusten demasiado a los datos de entrenamiento. En este proyecto, se utilizó un valor de decaimiento de 3e-05 para controlar la complejidad del modelo y mejorar su capacidad de generalización.

MULTIETIQUETA

Este parámetro indica si el problema de segmentación es de tipo multietiqueta, lo que significa que una entrada puede estar asociada con múltiples etiquetas. En este caso, se estableció que el problema no será abordado como multietiqueta, lo que significa que cada entrada se asocia con una sola posible salida. Esta configuración es importante para definir cómo se interpreta y procesa la salida de la red neuronal durante el entrenamiento y la inferencia.

AUMENTO DE DATOS

El aumento de datos (*data augmentation*), es una técnica utilizada en el entrenamiento de modelos de aprendizaje automático, especialmente en el campo del procesamiento de imágenes, para aumentar la diversidad y cantidad de datos disponibles para el entrenamiento. Consiste en aplicar transformaciones aleatorias y controladas a las imágenes originales para crear nuevas versiones de las mismas. Estas transformaciones incluyen rotaciones, traslaciones, cambios en la escala, modificaciones en el brillo y contraste, entre otros.

El objetivo principal del aumento de datos es mejorar la generalización del modelo y reducir el sobreajuste al proporcionar variaciones realistas de los datos originales. Al exponer el modelo a diferentes variaciones de los datos durante el entrenamiento, se espera que aprenda patrones más robustos y adaptables, lo que puede mejorar su rendimiento en datos de prueba no vistos.

En el contexto de entrenamientos con un número limitado de muestras, como en este estudio que involucra solamente 15 muestras, el uso del aumento de datos resulta

fundamental. Dado que la cantidad de datos es insuficiente para entrenar efectivamente un modelo de manera convencional sin riesgo de sobreajuste, se recurre a esta técnica para ampliar artificialmente el conjunto de entrenamiento. Al aplicar transformaciones como las mencionadas se generan nuevas instancias de datos que ayudan a enriquecer y diversificar el conjunto de entrenamiento.

DATOS DE ENTRENAMIENTO

OBTENCIÓN Y ORIGEN

El grupo Yatiris facilitó un conjunto de datos compuesto por 16 imágenes de pacientes tratados, las cuales fueron recopiladas y empleadas como dataset para el entrenamiento de los modelos. Cada imagen representa visualmente los resultados del tratamiento aplicado a un paciente específico. Esta representación tridimensional del cerebro del paciente, se obtuvo mediante técnicas avanzadas de imagenología médica, específicamente las imágenes utilizadas en este trabajo son tomografías computarizadas 3D con contraste (ver figura 5). La CT permite capturar imágenes detalladas del cerebro en múltiples planos, que luego son reconstruidas en un modelo 3D, proporcionando una visión completa y precisa de la estructura cerebral y los efectos del tratamiento. El contraste en las CT usadas en este trabajo es una sustancia radio-opaca que se inyecta en el torrente sanguíneo para visualizar la vasculatura cerebral.

Cada imagen está asociada con su respectiva segmentación, la cual indica el dispositivo endosacular WEB usado durante el procedimiento para tratar el aneurisma. Como se mencionó antes, las segmentaciones son el resultado de una labor manual realizada por especialistas utilizando el software Slicer 3D, donde al visualizar la imagen se trabaja por encima delimitando la sección que abarca el dispositivo endosacular. El propósito fundamental de este conjunto de datos radica en el desarrollo y entrenamiento de modelos de inteligencia artificial destinados a reconocer y comprender los efectos del tratamiento en estas imágenes.

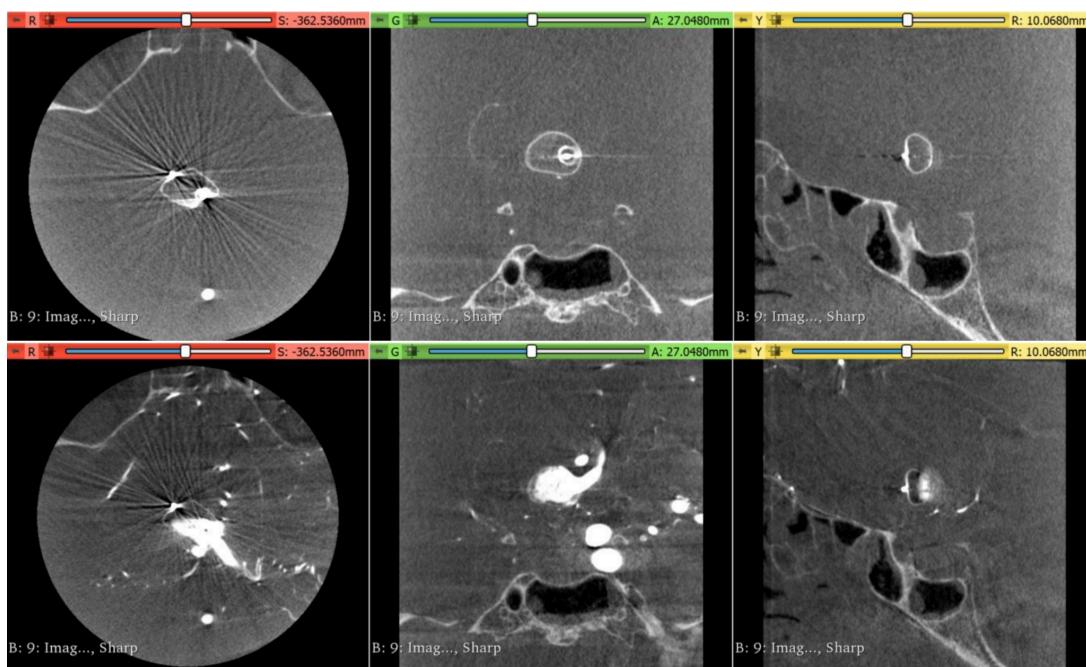


Figura 5: Arriba la máscara de la CT 3D con contraste que se observa abajo. En la máscara se observa la estructura ósea de la región y el dispositivo WEB. En la CT con contraste además de éstas estructuras se visualiza la vasculatura. En el cuadro rojo se muestra el plano axial, en el cuadro verde el plano coronal y en el cuadro amarillo el plano sagital.

El dispositivo fue segmentado manualmente usando el módulo Segment Editor del software Slicer 3D. La imagen utilizada para esta tarea fue la máscara de la CT 3D con contraste de cada caso. La máscara de una CT 3D es la imagen de la misma región obtenida antes de la inyección de contraste y por lo tanto, es una imagen en la que no se observa la vasculatura pero sí se observan las estructuras óseas y cualquier elemento metálico implantado en el paciente, en este caso un dispositivo WEB. (ver figura 5).

El plano anatómico axial, coronal o sagital (imagen 5) con la mejor visibilidad del contorno del dispositivo fue seleccionado para dibujar manualmente, slice por slice, el borde del WEB y luego generar el modelo de superficie 3D (ver figura 6). En algunos *slices*, el contorno del WEB no podía distinguirse con claridad, en esos casos los *slices* adyacentes (anterior y posterior) fueron usados como referencias para dibujar el contorno no observable. Además, la subsiguiente visualización tridimensional (Paraview) de la superficie del WEB obtenida, junto con la imagen usada para su segmentación permitió detectar cualquier error cometido en la delimitación del contorno del dispositivo.

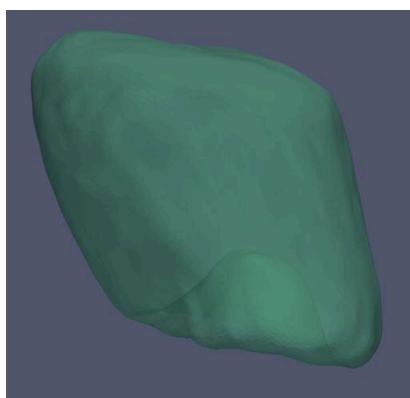


Figura 6: Modelo de superficie 3D del dispositivo WEB obtenido mediante segmentación manual de la máscara de la CT.

Esta segmentación manual se hizo usando la máscara de la CT 3D con contraste debido a la mayor visibilidad del dispositivo que esta modalidad ofrece. Sin embargo, esta imagen no siempre está disponible y es por eso que en este trabajo se utilizaron las imágenes CT 3D con contraste para entrenar los modelos. A diferencia de la máscara, en la imagen con contraste se observa el flujo sanguíneo que rodea al dispositivo dificultando la detección de su contorno de manera manual. Al entrenar los modelos de segmentación con las imágenes CT 3D con contraste no sólo se desarrolló una herramienta que facilita el trabajo de los especialistas sino también que puede ser utilizado en todos los casos, sin estar limitado a la disponibilidad de la máscara de la CT 3D con contraste.

ORGANIZACIÓN

Durante la fase inicial del preprocesamiento del dataset, se enfrentó el desafío de homogeneizar los formatos de los archivos de imágenes y segmentaciones. Las imágenes originales estaban en formato .nii.gz, el cual era compatible con la red neuronal utilizada. Sin embargo, las segmentaciones se encontraban en formato .vtk, que no era compatible con los requisitos del modelo. Para abordar esta discrepancia, se empleó el software Slicer 3D, el cual facilitó la conversión de las segmentaciones al formato .nii.gz necesario para la red neuronal. Este proceso aseguró la uniformidad en el formato de todas las imágenes y sus respectivas segmentaciones, finalizando así la preparación del conjunto de datos para su uso en las etapas subsiguientes del estudio.

Para la división del dataset en el contexto de este estudio, se adoptó una estrategia de validación cruzada (*cross-validation*) utilizando el método conocido como *k-fold cross-validation* [10]. Esta es una técnica utilizada en el aprendizaje automático para evaluar la capacidad predictiva de un modelo. Consiste en dividir el conjunto de datos en múltiples subconjuntos, entrenar el modelo en algunos de estos subconjuntos y validarlos en los restantes, repitiendo este proceso varias veces como pliegues (*folds*) se tengan. Esto permite utilizar todos los datos disponibles tanto para el entrenamiento como para la validación, proporcionando una evaluación más robusta y reduciendo la posibilidad de sobreajuste.

Dado el tamaño limitado del conjunto de datos, se optó por separar una única imagen para ser utilizada exclusivamente como conjunto de prueba (*testing*), mientras que las restantes 15 imágenes fueron utilizadas para el entrenamiento y la validación. Se implementaron 5 pliegues en la validación cruzada. En cada pliegue, se separaron 3 imágenes como conjunto de validación, dejando las otras 12 imágenes para ser utilizadas como conjunto de entrenamiento. Esta separación permitió realizar un entrenamiento exhaustivo y una evaluación robusta del modelo en diferentes subconjuntos de datos, asegurando que cada imagen tuviera la oportunidad de ser parte tanto del conjunto de entrenamiento como del conjunto de validación en al menos uno de los pliegues. Implementar 5 dobleces es un enfoque estándar y ampliamente aceptado en validación cruzada, proporcionando un equilibrio adecuado entre la evaluación del modelo y la eficiencia computacional. Este número de pliegues optimiza el uso del conjunto de datos disponible y asegura suficiente diversidad en los datos de entrenamiento y validación.

Para las próximas secciones del informe, se utilizará la imagen designada para las diferentes pruebas, con el propósito de contrastar las inferencias de los modelos con esta segmentación original correspondiente al paciente P011. Esta elección se basó en que esta imagen refleja el tamaño de aneurisma más prevalente en el conjunto de datos usados [11], lo cual se alinea con los objetivos de la investigación orientada hacia un modelo básico.

Continuando, se presentan dos imágenes complementarias para profundizar en los resultados obtenidos. En la primera imagen (ver figura 7), se exhibe la visualización completa del paciente P011, abarcando todos los tejidos, huesos y otras estructuras relevantes. Esta representación ofrece una visión panorámica de la anatomía cerebral y su entorno, Enriqueciendo la comprensión del contexto médico en su totalidad.

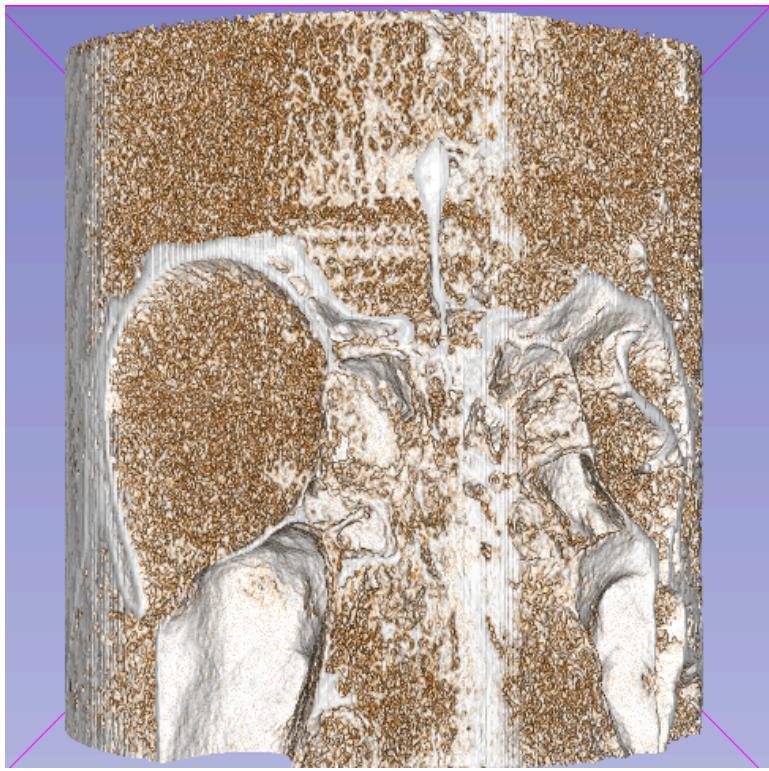


Figura 7: Paciente P011 sin filtrado de tejidos ni estructura ósea.

Por otro lado (ver figura 8), se presenta una versión procesada de la imagen original. En esta representación, se ha aplicado una máscara para suprimir los tejidos no esenciales, destacando únicamente las arterias y el aneurisma, junto con su tratamiento. Este enfoque resalta específicamente las áreas de interés para el diagnóstico y tratamiento de los aneurismas cerebrales, ofreciendo una visualización más precisa y pertinente para la evaluación clínica.

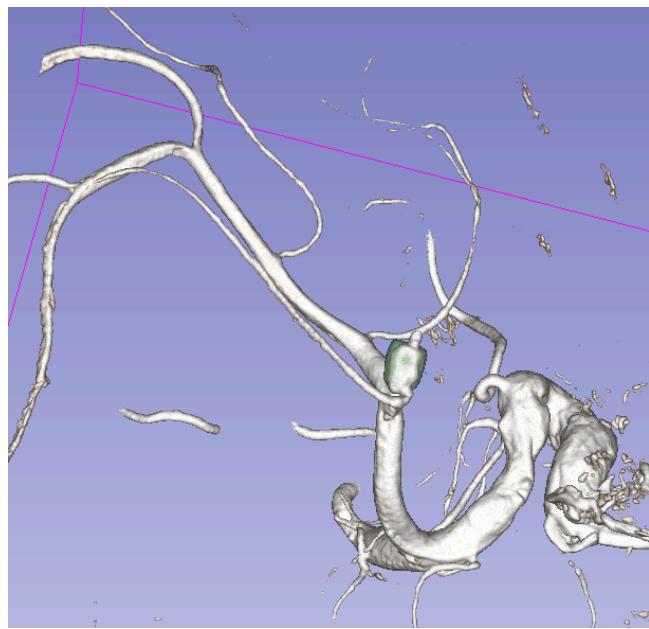


Figura 8: Paciente P011 con un filtrado de tejidos y estructuras irrelevantes.

Por último, dado que la imagen del paciente P011 es la que se reservó para pruebas, el objetivo es que las inferencias de los modelos reflejen resultados similares a las imágenes de segmentación manual de su dispositivo WEB. Estas representaciones detalladas, con vistas lateral derecha y superior, muestran cómo se ha realizado la segmentación del dispositivo en las imágenes médicas del paciente (ver figuras 9 y 10). El propósito es garantizar que los modelos entrenados puedan identificar y segmentar con precisión los dispositivos en nuevas imágenes.

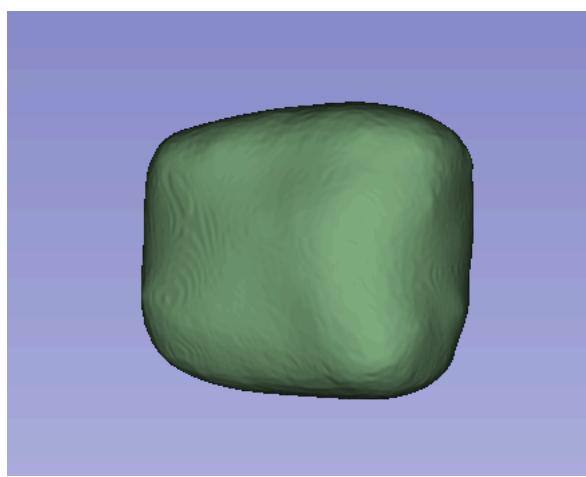


Figura 9: Vista lateral derecha del dispositivo endosacular de paciente P011.

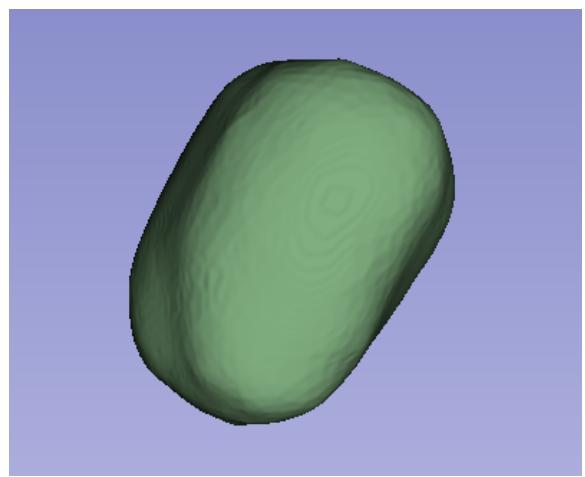


Figura 10: Vista superior del dispositivo endosacular de paciente P011.

PLANIFICACIÓN Y PREPROCESAMIENTO

CONFIGURACIÓN DE DIRECTORIOS Y VARIABLES DE ENTORNO

Durante la fase inicial de este estudio, se realizaron esfuerzos exhaustivos para planificar y llevar a cabo el preprocesamiento de los datos necesarios para el posterior entrenamiento del modelo. Se logró descargar exitosamente los registros de pacientes relevantes para la investigación y se realizó una configuración meticulosa del entorno de desarrollo en los entornos locales y en el servidor utilizado.

En primer lugar, se procedió con la instalación completa y funcional de UNet junto con todas sus dependencias en el servidor de Pladema. Esta etapa fue esencial para asegurar la disponibilidad de las herramientas necesarias y poder implementar y ejecutar el modelo nnUNet de manera efectiva durante las fases subsiguientes del estudio. En nnUNet, la configuración de directorios se centra en tres carpetas principales, cada una con su función específica para facilitar el flujo de trabajo.

La primera, *nnUNet_raw*, desempeña un papel crucial al almacenar los conjuntos de datos en su formato original. Estos conjuntos de datos se organizan en sub carpetas individuales, identificadas por un código de tres dígitos seguido del nombre único del conjunto de datos (ver figura 11). Dentro de cada carpeta de conjunto de datos se esperan subcarpetas que contienen imágenes de entrenamiento (*imagesTr*), imágenes de prueba opcionales (*imagesTs*), y etiquetas de segmentación para las imágenes de entrenamiento (*labelsTr*).

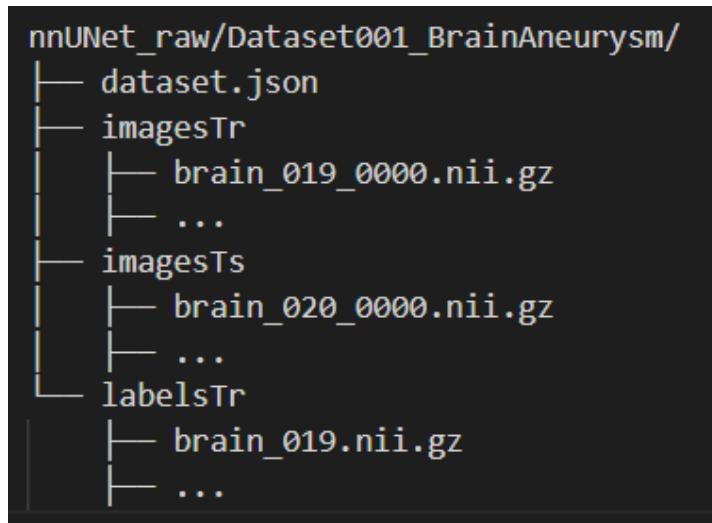


Figura 11: Configuración de directorios para almacenar el dataset.

Además, cada conjunto de datos tiene un archivo nombrado *dataset.json* que contiene metadatos relevantes para los entrenamientos. Estos son, el campo *channel_names*, que especifica los nombres de los diferentes canales de imágenes presentes en el conjunto de datos. El campo *labels* define las etiquetas de segmentación utilizadas, asignando un valor

numérico a cada categoría de interés, como fondo y zonas específicas del área segmentada. El campo *numTraining* indica el número de casos de entrenamiento disponibles en el conjunto de datos, proporcionando una referencia sobre la cantidad de datos utilizados para entrenar el modelo. El campo *file-ending* especifica la extensión de los archivos de imagen, asegurando que el sistema reconozca el formato correcto de los archivos. Opcionalmente, el campo *overwrite_image_reader_writer* define el lector/escritor de imágenes que se debe utilizar, permitiendo al usuario especificar un método particular para manejar los archivos de imagen.

Por otro lado, siguiendo con los directorios más generales, *nnUNet_preprocessed* se utiliza para almacenar los datos preprocesados, que son fundamentales durante el entrenamiento del modelo. La elección de esta ubicación es crucial, ya que esta carpeta debe estar en una unidad de almacenamiento con baja latencia de acceso y alto rendimiento para garantizar un procesamiento eficiente.

Finalmente, la carpeta *nnUNet_results* tiene un propósito claro, especificar dónde se guardarán los pesos del modelo entrenado y cualquier otro resultado generado por la red.

Después de establecer la estructura de directorios siguiendo las pautas proporcionadas por nnUNet, se procedió a configurar tres variables de entorno adicionales para facilitar el acceso a las carpetas de datos directamente desde los comandos utilizados en el estudio. Estas variables son esenciales para que nnUNet pueda ubicar dónde se guardarán los datos en bruto, los datos preprocesados y los modelos entrenados.

PREPROCESAMIENTO Y MANEJO DE LA SALIDA

Posteriormente, se utilizó el comando *nnUNetv2_plan_and_preprocess* con el objetivo de ejecutar el preprocesamiento. Este comando permite verificar la integridad y estructura del conjunto de datos utilizado en el estudio, asegurando que estuviera listo y correctamente formateado para su procesamiento posterior. Dentro de este proceso, el comando también adaptó y cambió el formato de las segmentaciones asociadas con los datos para garantizar la coherencia y compatibilidad de las etiquetas de segmentación con los requisitos específicos de la red nnUNet.

Al finalizar se crea una nueva subcarpeta en el directorio *nnUNet_preprocessed*, nombrada según el conjunto de datos, en este caso, “Dataset001_BrainAneurysm”. Al completarse el comando, se generan los archivos *dataset_fingerprint.json* y *nnUNetPlans.json*, que contienen información relevante sobre los datos proporcionados y los planes de nnUNet. Además, se crean subcarpetas que contienen los datos preprocesados para las configuraciones de UNet, garantizando que todo esté preparado para las siguientes fases.

El archivo *dataset_fingerprint.json* contiene información muy importante sobre las características específicas de las imágenes. Este archivo captura una serie de propiedades inherentes al conjunto de datos, como el tamaño de las imágenes, la distribución de los voxels y la información de intensidad. Estos detalles son utilizados por la red para diseñar y ajustar configuraciones específicas de UNet que son óptimas para los datos preprocesados. La información que brinda este archivo permite al sistema determinar estrategias de preprocesamiento adecuadas, como la normalización de la intensidad, el espaciamiento objetivo para el re-muestreo y el tamaño de los parches para el entrenamiento del modelo.

Por otro lado, el archivo *nnUNetPlans.json* contiene los planes de entrenamiento generados a partir del análisis del archivo *dataset_fingerprint.json*. Este archivo incluye todos los parámetros necesarios para la configuración del modelo, como la estructura de la red, las estrategias de aumento de datos, las funciones de pérdida, la tasa de aprendizaje y el optimizador a utilizar. También, asegura que todas las etapas del pipeline de nnUNet, desde el preprocesamiento hasta el entrenamiento y la evaluación, se realicen de manera coherente y optimizada.

Con esta preparación exhaustiva, se establece una base sólida para el entrenamiento del modelo, garantizando un rendimiento óptimo y una segmentación precisa en las imágenes médicas. Ahora, con todos los datos y configuraciones en su lugar, el enfoque se dirige hacia la fase de entrenamiento, donde se evaluará y refinará el modelo para alcanzar los objetivos de la investigación.

ACERCA DE LAS INFERENCIAS

En las siguientes secciones se presentarán detalladamente los resultados obtenidos junto con los modelos desarrollados. Por lo tanto, es importante realizar una breve introducción sobre cómo se procedió con las inferencias para evaluar estos modelos.

Durante el desarrollo de este estudio, cada pliegue de cada modelo fue sometido a una inferencia utilizando una imagen designada exclusivamente para pruebas como se mencionó previamente. Esta imagen de prueba se mantuvo consistente a lo largo de las evaluaciones para asegurar una comparación justa entre los diferentes modelos. Las validaciones de los entrenamientos proporcionaron métricas cuantitativas determinantes para evaluar la precisión y el rendimiento de cada pliegue. Además de estas métricas, se llevó a cabo un análisis visual detallado comparando las segmentaciones resultantes con las imágenes originales utilizando Slicer 3D. Este análisis visual fue esencial para comprender la calidad y la precisión de las segmentaciones producidas por los modelos. El enfoque integrado de evaluación cuantitativa y visual proporcionó una visión completa del desempeño de los entrenamientos y daba el puntapié de los ajustes que se podrían analizar para el próximo entrenamiento.

MÉTRICAS CUANTITATIVAS Y SU IMPORTANCIA

Las métricas cuantitativas son herramientas fundamentales para evaluar el rendimiento de modelos de aprendizaje automático, especialmente en tareas como la segmentación de imágenes médicas. A lo largo del proceso de entrenamiento de cada pliegue, se utilizaron técnicas de validación para medir la capacidad del modelo al generalizar a datos no vistos.

Las validaciones permiten evaluar el desempeño del modelo en conjuntos de datos separados, distintos de aquellos utilizados para el entrenamiento, lo que ayuda a estimar cómo se comportará el modelo frente a datos nuevos en la práctica.

Gracias a estas validaciones realizadas por la red neuronal en cada uno de los pliegues, se obtuvieron cuatro indicadores fundamentales relacionados a las predicciones del modelo a lo largo del entrenamiento, estos son:

- **Verdaderos Positivos (TP):** Son los casos en los que el modelo predice correctamente la clase positiva. En otras palabras, son las instancias en las que el modelo identificó correctamente la presencia de dispositivo intromacular en el paciente.
- **Verdaderos Negativos (TN):** Son los casos en los que el modelo predice correctamente la clase negativa. Esto significa que el modelo ha identificado correctamente la ausencia del dispositivo buscado.
- **Falsos Positivos (FP):** Ocurren cuando el modelo predice incorrectamente la clase positiva. Es decir, el modelo indica la presencia de la característica o condición

cuando en realidad está ausente. Estos errores pueden ser críticos dependiendo del contexto, ya que representan falsas alarmas.

- **Falsos Negativos (FN):** Son los casos en los que el modelo predice incorrectamente la clase negativa. En otras palabras, el modelo no detecta la presencia de la característica o condición cuando en realidad está presente.

Estos valores son esenciales para calcular las métricas cuantitativas. Cada métrica proporciona información específica sobre diferentes aspectos del rendimiento del modelo, como su capacidad para identificar correctamente regiones de interés (TP), evitar falsas identificaciones (FP), capturar correctamente regiones no relevantes (TN) y no pasar por alto regiones importantes (FN).

La interpretación de estas métricas cuantitativas es esencial para comprender la efectividad y la precisión del modelo en la tarea de segmentación. Mediante el análisis de estas métricas, es posible identificar áreas de mejora y realizar ajustes en el proceso de entrenamiento para optimizar su rendimiento y su capacidad de generalización a nuevos datos. Entre las métricas cuantitativas calculadas a partir de los datos de las validaciones se tienen:

- **Tasa de Error de Clasificación:** Esta métrica representa la proporción de predicciones incorrectas realizadas por el modelo sobre el total de predicciones.

$$Tasa\ de\ Error = \frac{FP + FN}{VP + VN + FP + FN}$$

- **DICE:** El coeficiente DICE es una medida de similitud entre la segmentación predicha por el modelo y la segmentación de referencia (*ground truth*). Indica la superposición entre las áreas segmentadas.

$$DICE = \frac{2 \times VP}{2 \times VP + FP + FN}$$

- **Sensibilidad:** La sensibilidad es la proporción de verdaderos positivos (TP) respecto a todos los casos positivos reales (TP + FN). Indica la capacidad del modelo para identificar correctamente las instancias positivas.

$$Sensibilidad = \frac{VP}{VP + FN}$$

- **Precisión:** La precisión es la proporción de verdaderos positivos (TP) respecto a todas las predicciones positivas realizadas por el modelo (TP + FP). Representa la exactitud de las predicciones positivas.

$$Precisión = \frac{VP}{VP + FP}$$

- **Especificidad:** La especificidad es la proporción de verdaderos negativos (TN) respecto a todos los casos negativos reales (TN + FP). Indica la capacidad del modelo para identificar correctamente las instancias negativas.

$$Especificidad = \frac{VN}{VN + FP}$$

- **Exactitud:** La exactitud es la proporción de predicciones correctas realizadas por el modelo sobre el total de predicciones realizadas. Es una medida general del rendimiento del modelo en todas las clases.

$$Exactitud = \frac{VP + VN}{VP + VN + FP + FN}$$

Como se mencionó, todas las métricas se calcularon para cada pliegue de cada modelo y se recopilaron a modo de historial de mejoras. En cada entrenamiento, se utilizó el modelo obtenido en la última época para las comparaciones, ya que la implementación no permitía guardar los modelos de cada época, lo que impedía seleccionar el mejor dentro de cada ciclo de entrenamiento. Una vez finalizados los 5 pliegues, se eligió el mejor modelo entre ellos basándose en las métricas cuantitativas obtenidas. La tabla correspondiente con las métricas de cada entrenamiento proporciona una referencia clara y detallada de la evolución del desempeño a lo largo del tiempo, permitiendo identificar tendencias, evaluar la eficacia de diferentes enfoques y optimizar continuamente el proceso de aprendizaje.

ENTRENAMIENTOS

Durante la etapa de entrenamiento, se aplicó una metodología cíclica de prueba y evaluación para optimizar el rendimiento del modelo. Este enfoque iterativo abarcó desde el preprocesamiento de datos hasta la inferencia y el análisis de resultados, permitiendo ajustes en la red o en el proceso de entrenamiento según fuera necesario. La implementación de esta metodología fue esencial para el éxito del proyecto al permitir maximizar el uso de los datos disponibles y mejorar continuamente el modelo a lo largo de múltiples ciclos de evaluación. Cada ciclo incluyó fases de preprocesamiento de datos, entrenamiento del modelo, evaluación de resultados y ajustes iterativos, lo que condujo a mejoras significativas en la precisión y calidad de las predicciones del modelo de segmentación de dispositivos endosaculares.

En el contexto de los entrenamientos realizados hasta el momento con el conjunto de datos de los pacientes, se llevaron a cabo múltiples modelos de entrenamiento utilizando diferentes variantes de la clase *nnUNetTrainer* y ajustes de parámetros específicos. A continuación, se detalla el enfoque y los resultados obtenidos para cada uno de los modelos. A lo largo de las figuras presentadas, la segmentación original se muestra en color verde mientras que la inferida se presenta en amarillo.

GRÁFICOS DE REFERENCIA

Al finalizar el entrenamiento de un pliegue con la nnUNet, se generan tres gráficos esenciales que permiten analizar en detalle el desempeño de los modelos. Estos gráficos son fundamentales para comprender y evaluar el comportamiento del modelo durante su entrenamiento y optimización. A continuación, se describen los tres gráficos generados.

FUNCIONES DE PÉRDIDA Y PRECISIÓN DEL MODELO

El gráfico de la figura 12 ilustra cómo evolucionan las métricas de pérdida (*loss*) y el puntaje (*score*) DICE a lo largo de las diferentes épocas de entrenamiento. La pérdida representa la función objetivo que el modelo intenta minimizar, mientras que el puntaje DICE es una métrica de precisión utilizada comúnmente en tareas de segmentación médica, se calcula comparando la superposición entre dos conjuntos, en este caso la segmentación predicha por el modelo y la segmentación real o de referencia. En el gráfico, la curva *loss_tr* muestra la evolución de la pérdida en el conjunto de entrenamiento, mientras que *loss_val* representa la pérdida en el conjunto de validación. La disminución de estas curvas indica que el modelo está aprendiendo efectivamente a minimizar el error tanto en el conjunto de entrenamiento como en el de validación, lo cual es un signo positivo de que el modelo está mejorando su capacidad de predicción.

Sin embargo, la aparición de valores negativos en la pérdida es inesperada y podría considerarse incorrecta, dado que se espera que la disminución de la pérdida se acerque a 0, pero sin moverse a valores negativos. A continuación, se ofrece una explicación de por qué ocurre este fenómeno.

La función de pérdida utilizada por la nnU-Net combina dos componentes: la pérdida de DICE y la pérdida de entropía cruzada (cross-entropy loss). La formulación total de la pérdida se expresa como:

$$L_{total} = L_{DICE} + L_{CE}$$

Pérdida DICE: Es una métrica que se utiliza comúnmente en tareas de segmentación, ya que mide la superposición entre las predicciones del modelo y las etiquetas verdaderas. Se define como:

$$L_{DC} = - \frac{2}{|K|} \sum_{k \in K} \frac{\sum_{i \in I} u_i^k v_i^k}{\sum_{i \in I} u_i^k + \sum_{i \in I} v_i^k}$$

Donde:

- u es la salida de la función softmax de la red.
- v es la codificación one-hot del mapa de segmentación verdadero.
- I es el número de voxels en el lote de entrenamiento.
- K es el número de clases.

La ecuación devuelve un valor negativo como resultado del cálculo del coeficiente DICE. Por lo tanto, los valores resultantes estarán entre -1 y 0, siendo -1 el valor objetivo.

Pérdida de Entropía Cruzada: Mide la diferencia entre la distribución de probabilidad predicha por el modelo y la distribución real de las etiquetas. Se utiliza para optimizar la clasificación de cada voxel en las diferentes clases. Se define como:

$$L_{CE} = - \frac{1}{N} \sum_{i=1}^N \sum_{j=1}^C (y_{i,j} * \log(p_{i,j}))$$

Donde:

- N es el número de instancias.
- C es el número de clases.
- $y_{i,j}$ son las etiquetas verdades para la clase j e instancia i.
- $p_{i,j}$ es la probabilidad prevista para la clase j e instancia i.

La ecuación devuelve un valor entre 0 y 1, siendo 1 el máximo error posible y 0 la ausencia de error, lo que quiere decir que es el valor objetivo.

Combinación de pérdidas: Al combinar ambas pérdidas, nnU-Net no solo mejora la precisión de las predicciones (a través de la pérdida de DICE) sino también asegura que las probabilidades de clase sean correctas (a través de la pérdida de entropía cruzada). Esta combinación permite que el modelo sea más robusto y efectivo en la segmentación de imágenes médicas, adaptándose a diferentes tareas y conjuntos de datos de manera automática. Cabe resaltar que el valor objetivo de esta función de pérdida es -1, tal como lo expresa la siguiente ecuación:

$$L_{\text{total objetivo}} = L_{\text{DICE objetivo}} + L_{\text{CE objetivo}} = -1 + 0 = -1$$

La curva *pseudo DICE* refleja la precisión del modelo, y *pseudo DICE avg* indica la precisión promediada. Este gráfico es crucial para identificar posibles problemas de sobreajuste o subajuste y para evaluar la capacidad del modelo de generalizar sobre los datos de validación. El prefijo “pseudo” hace referencia a que se trata de una variación aproximada del coeficiente DICE, utilizado cuando no es posible calcular el índice DICE real de manera inmediata, como durante el entrenamiento del modelo. Aunque no es tan preciso como el DICE score verdadero, el *pseudo DICE* proporciona una buena indicación del rendimiento del modelo y permite monitorear su capacidad de predicción. Esto es útil para evaluar el comportamiento del modelo sin necesidad de realizar cálculos exhaustivos, ayudando a detectar de manera temprana problemas como el sobreajuste o subajuste.

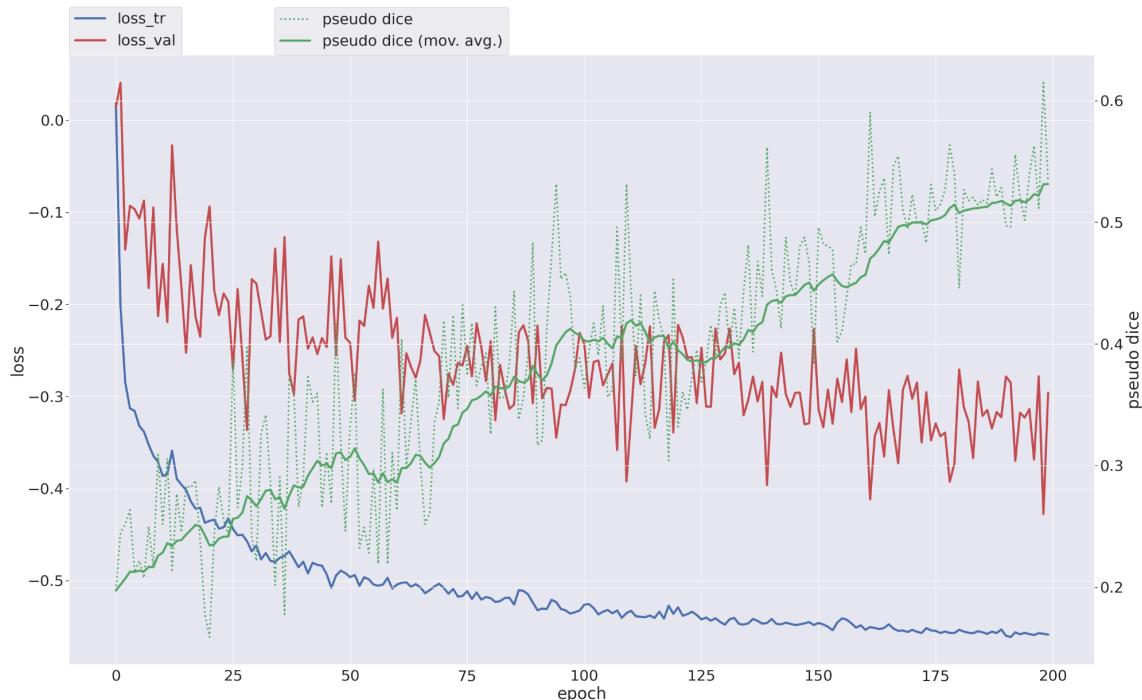


Figura 12: Ejemplo de gráfico de comportamiento de función de pérdida y pseudo DICE a lo largo de un entrenamiento de 200 épocas.

DURACIÓN DE LAS ÉPOCAS

La duración de cada época de entrenamiento se muestra en el gráfico de la figura 13. Esta representación gráfica es importante para entender la eficiencia del proceso y para identificar posibles cuellos de botella que puedan estar ralentizando el trabajo de la red. Un análisis detallado de este gráfico puede conducir a optimizaciones en el tiempo de entrenamiento, mejorando así la eficiencia global del aprendizaje del modelo.

TASA DE APRENDIZAJE

La tasa de aprendizaje es un hiper parámetro clave que controla la velocidad con la que el modelo ajusta sus parámetros en respuesta al gradiente de error. La figura 14 presenta cómo varía dicha tasa a lo largo del entrenamiento, proporcionando información sobre la estrategia de ajuste utilizada (por ejemplo, *learning rate scheduling*). Una correcta gestión de la tasa de aprendizaje es vital para asegurar una convergencia adecuada del modelo.

ESTRATEGIA DE AJUSTE DE LA TASA DE APRENDIZAJE

La técnica utilizada por la nnUnet es *PolyLRScheduler*, la cual se utiliza para controlar cómo se modifica la tasa de aprendizaje durante el proceso de entrenamiento. Esta estrategia se basa en una función polinómica, lo que significa que la tasa de aprendizaje disminuye de forma no lineal a medida que avanza el entrenamiento. A continuación se describe en detalle cómo funciona esta estrategia siguiendo una fórmula polinómica que calcula el aprendizaje por época.

$$lr_{epoch} = lr_{initial} * \left(1 - \frac{epoch}{epoch_{max}}\right)^{power}$$

- **Lr-initial:** La tasa de aprendizaje inicial al comienzo del entrenamiento.
- **Epoch:** La época actual del entrenamiento.
- **Epochs_{max}:** El número total de épocas planificadas para el entrenamiento.
- **Power:** Un hiper parámetro que controla la tasa de decrecimiento de la tasa de aprendizaje. Un valor de power mayor que 1 hará que la tasa de aprendizaje disminuya más lentamente al principio y más rápidamente hacia el final. Un valor menor que 1 tendrá el efecto contrario.

VENTAJAS

- **Convergencia Suave:** Al disminuir la tasa de aprendizaje de manera gradual, el *PolyLRScheduler* ayuda a que el modelo ajuste sus parámetros de manera más fina, especialmente en las últimas etapas del entrenamiento. Esto puede resultar en una mejor convergencia y una mayor precisión.
- **Evita Oscilaciones:** Una reducción polinómica ayuda a evitar grandes oscilaciones en la pérdida del modelo, proporcionando un ajuste más estable y consistente.

-
- **Flexibilidad:** El parámetro power permite ajustar la curva de decrecimiento de la tasa de aprendizaje, proporcionando flexibilidad para adaptarse a diferentes tipos de problemas y modelos.

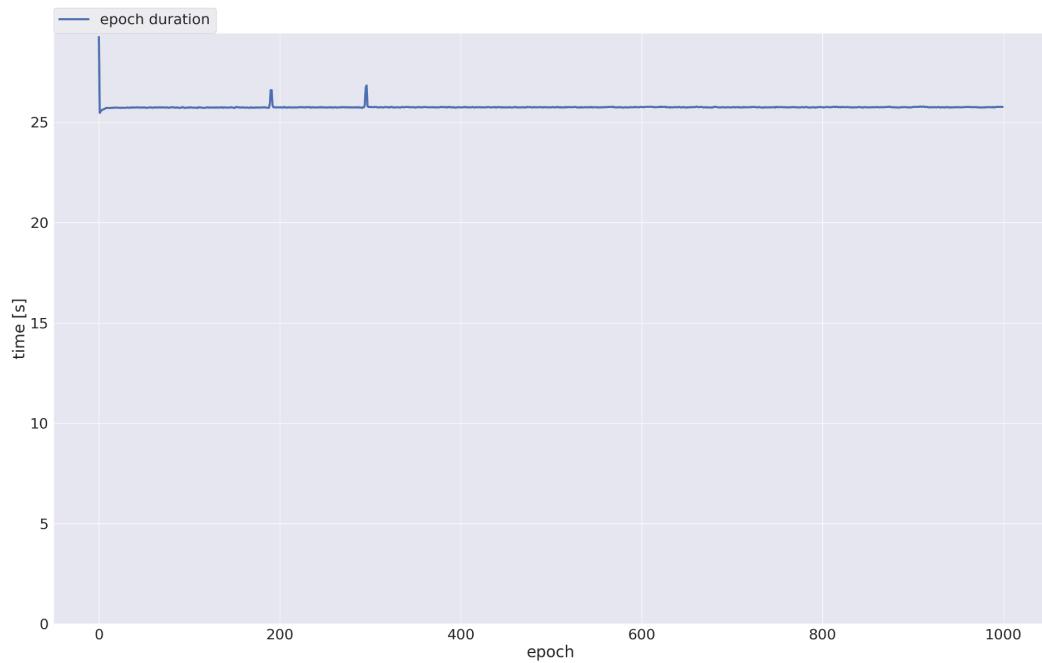


Figura 13: Ejemplo de gráfico de duración de las épocas.

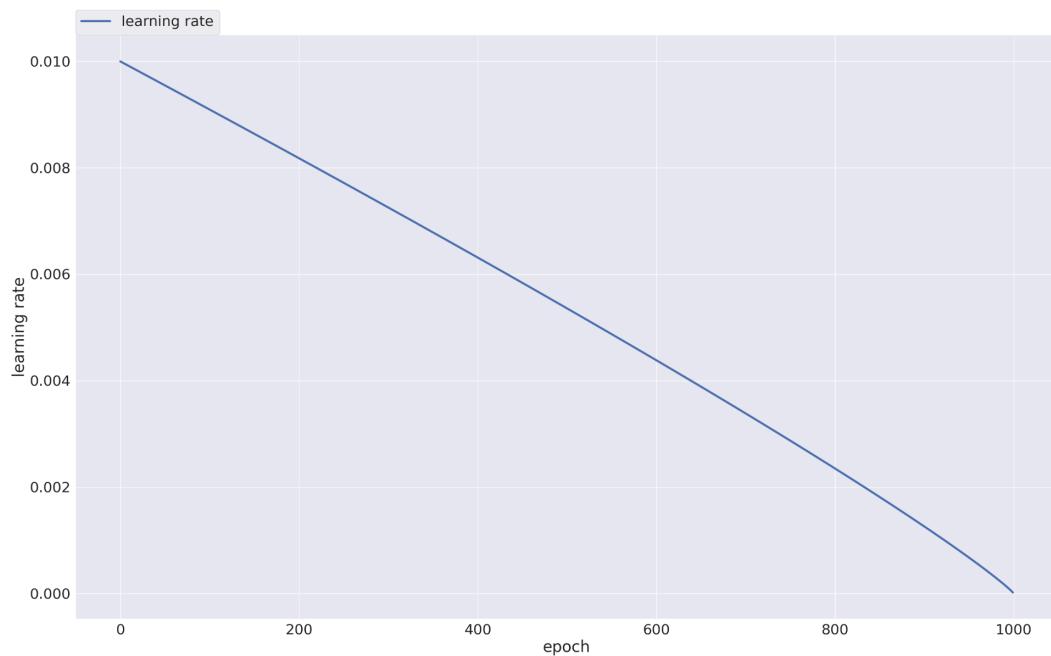


Figure 14: Ejemplo de gráfico de tasa de aprendizaje con respecto a la cantidad de épocas.

MODELO 0

Este modelo se entrenó utilizando los parámetros de la red por defecto, sin aplicar ninguna variante de la clase *nnUNetTrainer*, sino la que ya provee la arquitectura. Lo único que se modificó fue la cantidad de épocas por defecto, que originalmente era de 1000 y se redujo a 200. Esta reducción se realizó para evitar el sobreajuste (concepto abordado al final de este modelo), dado el reducido tamaño de las muestras disponibles, y para reducir los largos tiempos de entrenamiento. Dejando de lado esto, los parámetros quedaron con sus valores originales (ver tabla 1).

Entrenamiento Modelo 0	
Parámetro	Valor
Configuración	3d_fullres
Entrenador	nnUNetTrainer
Tamaño del lote	2
Épocas	200
Iteraciones por época	250
Tasa de aprendizaje inicial	0.01
Porcentaje de sobremuestreo de fondo	0.33
Decaimiento del peso	3,00E-05
Pliegues entrenados	[0, 1, 2, 3, 4]
Multietiqueta	FALSE

Tabla 1: Configuración utilizada para el entrenamiento del Modelo 0.

Luego del entrenamiento, se observó que los pliegues 2 (ver figura 13 y 14) y 3 (ver figura 15 y 16) presentaron los resultados más prometedores en términos de métricas de evaluación, lo que indica un rendimiento destacado en estas particiones específicas del conjunto de datos. Este entrenamiento permitió dar un primer paso en la investigación y tomar confianza con la red y su implementación. Los resultados obtenidos fueron positivos, teniendo en cuenta que no hubo modificaciones ni ajustes significativos.

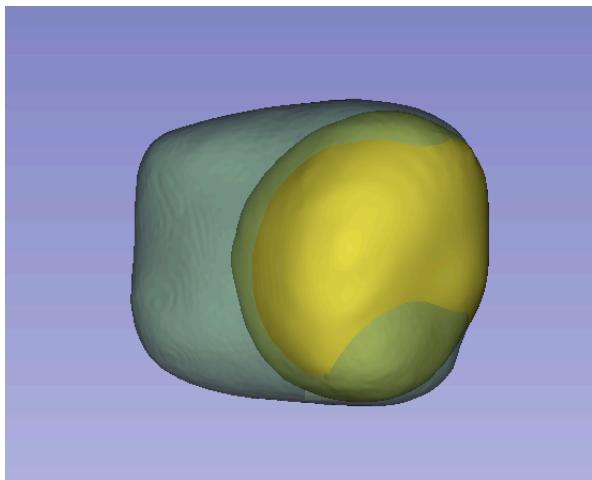


Figura 13: Vista lateral de la inferencia obtenida utilizando el pliegue 2 en contraste con la segmentación original.

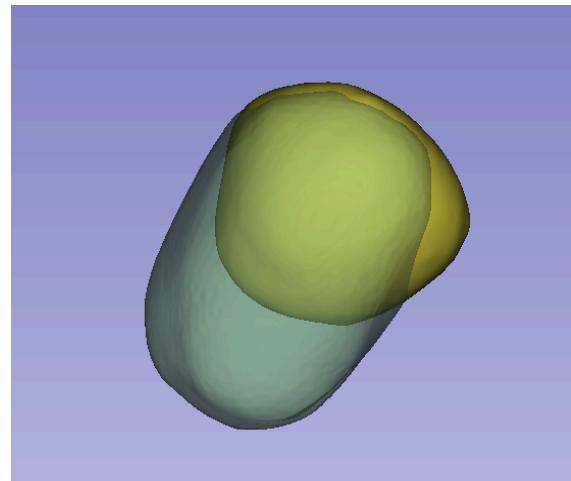


Figura 14: Vista superior de la inferencia obtenida utilizando el pliegue 2 en contraste con la segmentación original.

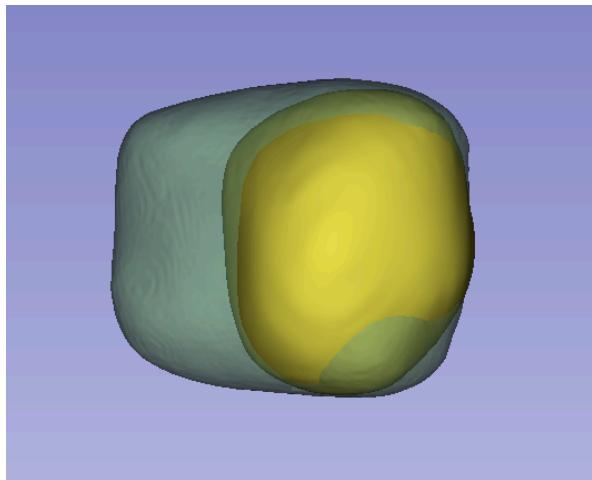


Figura 15: Vista lateral de la inferencia obtenida utilizando el pliegue 3 en contraste con la segmentación original.

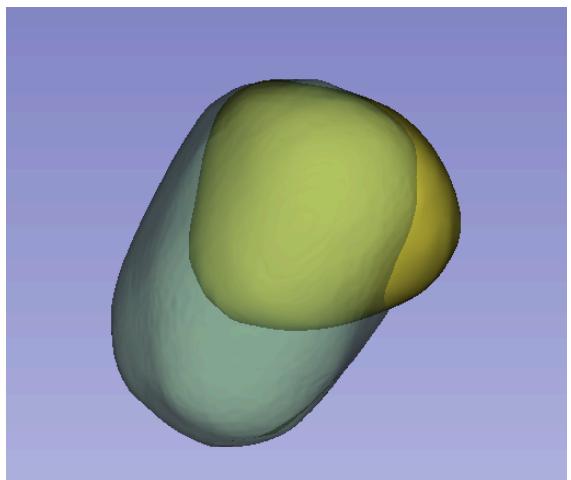


Figura 16: Vista lateral de la inferencia obtenida utilizando el pliegue 3 en contraste con la segmentación original.

A continuación, se presentan las métricas cuantitativas obtenidas de los dos pliegues anteriores (ver tabla 2 y 3). Estas métricas son fundamentales para evaluar el rendimiento del modelo y su capacidad de generalización en distintas particiones del conjunto de datos. Los resultados obtenidos reflejan la eficacia del modelo en términos de precisión, sensibilidad, especificidad y otras métricas relevantes, proporcionando una visión detallada del comportamiento del modelo en estos dos pliegues específicos.

Modelo 0 - Pliegue 2	
Métrica	Valor
Tasa de Error de Clasificación	0,000044
Índice DICE	0,879339
Sensibilidad	0,884399
Precisión	0,874337
Especificidad	0,999977
Exactitud	0,999956

Tabla 2: Métricas cuantitativas del pliegue 2 del Modelo 0.

Modelo 0 - Pliegue 3	
Métrica	Valor
Tasa de Error de Clasificación	0,000034
Índice DICE	0,816696
Sensibilidad	0,721689
Precisión	0,940510
Especificidad	0,999995
Exactitud	0,999966

Tabla 3: Métricas cuantitativas del pliegue 3 del Modelo 0.

Estas diferencias observables pueden significar que el modelo tiene un rendimiento más equilibrado en el pliegue 2, con una mejor capacidad para identificar correctamente las verdaderas instancias positivas (alta sensibilidad: 0.884399 frente a 0.721689) y una mejor superposición general entre las predicciones y las etiquetas reales (índice DICE: 0.879339 frente a 0.816696). En contraste, el pliegue 3 muestra una mayor precisión (0.940510 frente a 0.874337), lo que sugiere que el modelo en este pliegue es más conservador en sus predicciones positivas, priorizando la reducción de falsos positivos.

Estas variaciones pueden estar relacionadas con la distribución específica de datos en cada pliegue. El pliegue 2 parece ofrecer un conjunto de datos que permite al modelo aprender características que favorecen la detección y segmentación más precisa, mientras que el pliegue 3 podría estar compuesto por datos que favorecen una mayor precisión a expensas de una menor sensibilidad.

SOBREAJUSTE

Como se mencionó antes, este modelo estaba configurado inicialmente para entrenar durante 1000 épocas. Sin embargo, al analizar el conjunto de datos limitado utilizado para el entrenamiento, se observó un claro caso de sobreajuste en las curvas de aprendizaje. Esto

sugirió que el modelo estaba sobreajustando los datos de entrenamiento y necesitaba ser ajustado para mejorar su capacidad de generalización.

El sobreajuste (*overfitting*) es un fenómeno común en el aprendizaje automático que ocurre cuando un modelo se ajusta demasiado bien a los datos de entrenamiento específicos utilizados, en lugar de aprender patrones generales que se puedan aplicar a datos nuevos o no vistos. Se manifiesta como un rendimiento deficiente del modelo en datos de prueba o validación, a pesar de tener un buen desempeño en los datos de entrenamiento.

Este sobreajuste se da cuando un modelo es demasiado complejo en relación con la cantidad y la variabilidad de los datos de entrenamiento disponibles. Esto puede suceder especialmente en conjuntos de datos pequeños, donde el modelo puede memorizar los ejemplos de entrenamiento en lugar de aprender las características subyacentes que generalizan bien a nuevas muestras.

En el contexto de este trabajo, se entrenó un modelo durante 1000 épocas con el conjunto de datos limitado para detectar la presencia de sobreajuste y dejarlo plasmado en este informe (ver figura 17). Aquí, se puede observar la evolución de la función de pérdida en los datos de validación (*loss_val*), representada por la curva roja. Inicialmente, la curva de pérdida en los datos de validación desciende, indicando que el modelo está aprendiendo y mejorando su capacidad para generalizar. Sin embargo, después de cierto número de épocas, la curva roja comienza a ascender. Este aumento en la función de pérdida es una señal clara de sobreajuste.

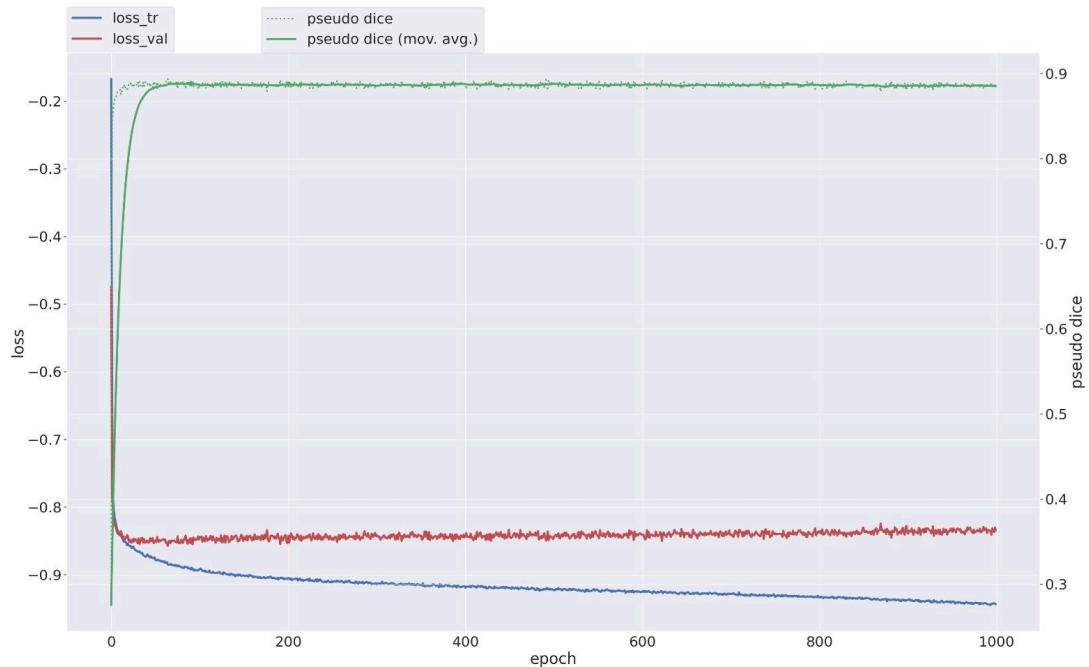


Figura 17: Gráfico del entrenamiento de un modelo entrenado con 1000 épocas.

Este gráfico es crucial para identificar y entender el punto a partir del cual el modelo comienza a sobreajustarse a los datos de entrenamiento. Una buena analogía para explicar este fenómeno es el comportamiento humano cuando memorizamos respuestas específicas

para un examen en lugar de comprender el material de manera general. Similar a un estudiante que memoriza respuestas específicas y luego se encuentra perdido cuando las preguntas del examen son ligeramente diferentes, un modelo sobreajustado ha aprendido demasiado bien los detalles específicos de los datos de entrenamiento y falla al enfrentarse a nuevos datos que no se ajustan exactamente a lo que ha memorizado. Esto subraya la importancia de encontrar un equilibrio adecuado, asegurando que el modelo pueda generalizar bien en diversas situaciones, tal como un buen estudiante que entiende los conceptos en lugar de simplemente memorizar respuestas.

Ahora bien, si llevamos esta información al dataset utilizado en este trabajo, podemos ver un gráfico del Modelo 0 (ver figura 18) entrenado con 200 épocas donde se observa que las curvas de aprendizaje tienen un rendimiento cada vez mejor en los datos de entrenamiento (*loss_tr*) pero sin una mejora significativa en los datos de validación (*loss_val*).

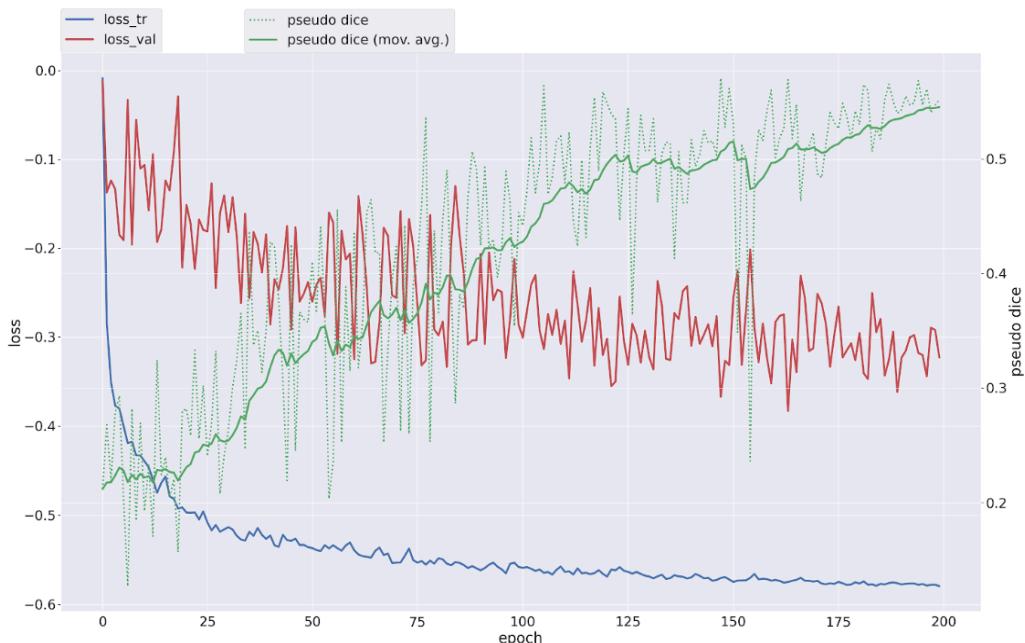


Figura 18: Gráfico del entrenamiento del Modelo 0 entrenado con 200 épocas.

Esto condujo a la realización de dos nuevos entrenamientos. El Modelo 1, donde se implementó el aumento de datos con el objetivo de aumentar artificialmente la cantidad de muestras disponibles para el entrenamiento. Esta estrategia permite al modelo aprender de una variedad más amplia de datos sintéticos, lo que puede ayudar a mitigar el sobreajuste y mejorar la generalización del modelo.

Por otro lado, el Modelo 2, donde se redujo la cantidad de épocas de entrenamiento con el fin de realizar menos iteraciones en los datos. Al disminuir el número de iteraciones, se busca evitar que el modelo memorice demasiado los datos específicos de entrenamiento y, en cambio, se concentre en aprender patrones más generales que puedan aplicarse a datos nuevos.

MODELO 1

Para este modelo, se hizo uso de uno de los entrenadores (*trainers*) prefabricados por la red, una variante específica de la clase *nnUNetTrainer* llamada *nnUNetTrainerDA5* (*DA* corresponde a *Data Augmentation* y *5* indica la versión).

El *nnUNetTrainerDA5*, es una variante del entrenador *nnUNetTrainer* que se especializa en la preparación de datos, implementando técnicas avanzadas de aumentación de datos. Utiliza estrategias de aumentación más agresivas para mejorar la generalización del modelo. Algunas de las principales funciones implementadas incluyen la configuración dinámica de rotación, reflejo (*mirroring*) y tamaño inicial del parche (*patch size*) para adaptarse a diferentes dimensiones y características de los datos. Estas estrategias de aumento de datos son fundamentales para garantizar la generalización y robustez del modelo en diversas condiciones de entrada. De esta forma los parámetros quedaron de igual manera que en el Modelo 0 a excepción del entrenador (ver tabla 4).

Entrenamiento Modelo 1	
Parámetro	Valor
Configuración	3d_fullres
Entrenador	nnUNetTrainerDA5
Tamaño del lote	2
Épocas	200
Iteraciones por época	250
Tasa de aprendizaje inicial	0.01
Porcentaje de sobremuestreo de fondo	0.33
Decaimiento del peso	3,00E-05
Pliegues entrenados	[0, 1, 2, 3, 4]
Multietiqueta	FALSE

Tabla 4: Configuración utilizada para el entrenamiento del Modelo 1.

El pliegue 4 de este modelo fue el que se destacó como el mejor rendimiento logrado, demostrando la eficacia del aumento de datos en mejorar la capacidad de generalización (ver figura 19 y 20).

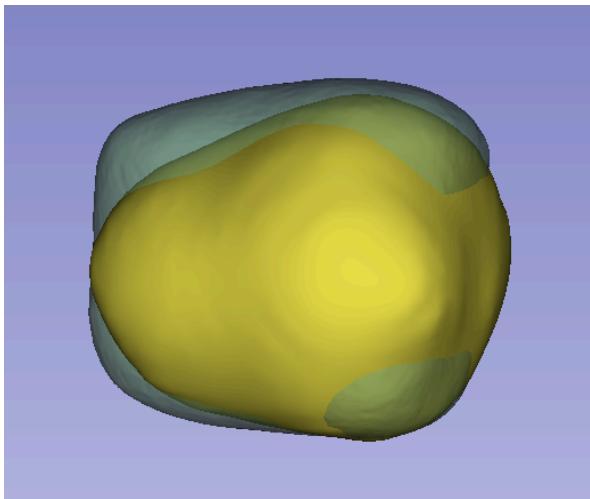


Figura 19: Vista lateral de la inferencia obtenida utilizando el pliegue 4 en contraste con la segmentación original.

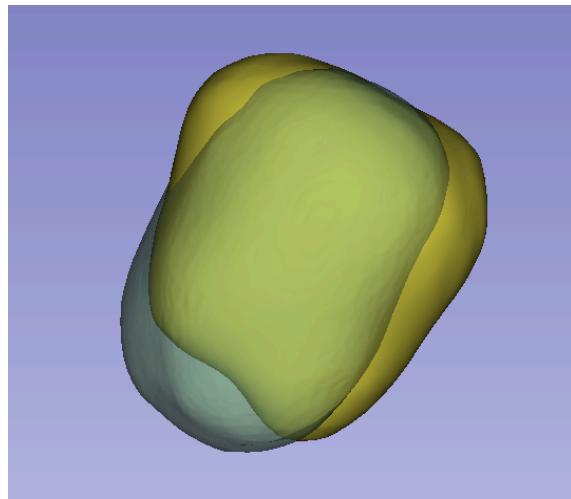


Figura 20: Vista superior de la inferencia obtenida utilizando el pliegue 4 en contraste con la segmentación original.

A continuación, se presentan las métricas cuantitativas obtenidas del modelo 1 en el pliegue 4 (ver tabla 5). Al analizarlas, se observa un equilibrio entre precisión y sensibilidad. Estas métricas son indicativas de la capacidad del modelo para realizar predicciones precisas mientras mantiene una capacidad robusta para detectar instancias positivas. Esto sugiere que el modelo es capaz de generar resultados confiables y consistentes superando ampliamente al Modelo 0.

Modelo 1 - Pliegue 4	
Métrica	Valor
Tasa de Error de Clasificación	0,000022
Índice DICE	0,721604
Sensibilidad	0,619576
Precisión	0,863860
Especificidad	0,999995
Exactitud	0,999978

Tabla 5: Métricas cuantitativas del pliegue 4 del Modelo 1.

MODELO 2

El modelo 2 fue entrenado utilizando una implementación personalizada del *nnUNetTrainer* padre, conocida como *nnUNetTrainerAneurysm*. Este entrenador es una implementación personalizada creada específicamente para el contexto del trabajo. En él se seleccionan transformaciones clave como Transformación Gamma, Transformación de Brillo y Rotación de 90 Grados, entre otras, y se ajustan los hiper parámetros de la red para optimizar su rendimiento en la tarea específica. Se heredó de la clase *nnUNetTrainer* y se realizaron las modificaciones necesarias en esa implementación. Para este modelo inicial, se

ajustó el número de épocas de entrenamiento a 100 en lugar de las 200 como se venía trabajando, quedando la configuración como se muestra a continuación (ver tabla 6).

Los resultados que se muestran (ver figuras 21 y 22) provienen del pliegue 0, este fue el único pliegue entrenado, para de esta forma, obtener una salida rápida que permitió conocer la respuesta de la red ante tal pequeña cantidad de épocas. Sin embargo no se alcanzaron resultados a la altura de los modelos anteriores, demostrando que el cambio realizado no fue propicio para mejorar.

Entrenamiento Modelo 2	
Parámetro	Valor
Configuración	3d_fullres
Entrenador	nnUNetTrainerAneurysm
Tamaño del lote	2
Épocas	100
Iteraciones por época	250
Tasa de aprendizaje inicial	0.01
Porcentaje de sobremuestreo de fondo	0.33
Decaimiento del peso	3,00E-05
Pliegues entrenados	[0]
Multietiqueta	FALSE

Tabla 6: Configuración utilizada para el entrenamiento del Modelo 2.

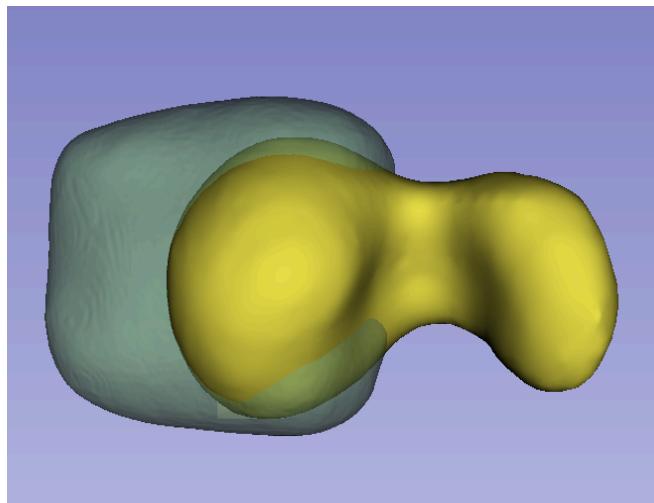


Figura 21: Vista lateral de la inferencia obtenida utilizando el pliegue 0 en contraste con la segmentación original.

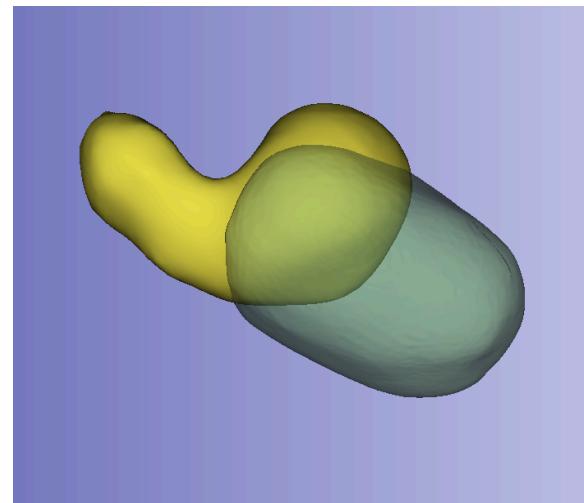


Figura 22: Vista superior de la inferencia obtenida utilizando el pliegue 0 en contraste con la segmentación original.

Las métricas del Modelo 2 en el pliegue 0 (ver tabla 7), muestran un rendimiento considerablemente inferior en comparación con los modelos anteriores. La tasa de error de

clasificación es más alta, alcanzando un valor de 0.000320, lo que indica una menor precisión en la clasificación de las instancias. El índice DICE muestra un rendimiento deficiente, con un valor de 0.391244. Esto sugiere que las predicciones del modelo tienen una superposición significativamente menor con las etiquetas reales, lo que podría indicar una segmentación menos precisa. La sensibilidad del modelo es baja, con un valor de 0.258824, lo que indica que el modelo tiene dificultades para identificar correctamente las instancias positivas. Esto sugiere que hay un alto número de falsos negativos, es decir, casos positivos que el modelo no logra detectar. A pesar de que la precisión es relativamente alta, con un valor de 0.801107, esto puede ser engañoso dado el bajo rendimiento en otras métricas. Es importante destacar que la alta precisión puede ser el resultado de un alto número de verdaderos negativos, lo cual no necesariamente refleja una buena capacidad de detección de las instancias positivas.

Modelo 2 - Pliegue 0	
Métrica	Valor
Tasa de Error de Clasificación	0,000320
Índice DICE	0,391244
Sensibilidad	0,258824
Precisión	0,801107
Especificidad	0,999974
Exactitud	0,999680

Tabla 7: Métricas cuantitativas del pliegue 0 del Modelo 2.

RESUMEN ENTRENAMIENTOS 1

Se continuó realizando varias pruebas para mejorar el modelo. En una de ellas, se redujo la cantidad de iteraciones por época de 250 a solo 3. Esta modificación buscaba disminuir la cantidad de datos que el modelo ve en cada paso de entrenamiento, lo que podría ayudar a evitar el sobreajuste. Además, al reducir las iteraciones por época, el proceso de entrenamiento se acelera, lo que resulta útil para conjuntos de datos grandes o cuando se busca optimizar el tiempo de entrenamiento. Sin embargo, esta reducción drástica llevó a una menor convergencia del modelo y a una capacidad de aprendizaje limitada, lo que se reflejó en resultados desfavorables. Las métricas obtenidas mostraron un desempeño deficiente en comparación con los estándares requeridos para la tarea. La alta tasa de error de clasificación y el bajo índice DICE indicaron dificultades significativas en la clasificación precisa y la superposición con las etiquetas reales. La sensibilidad, precisión e índice DICE fueron extremadamente bajos, demostrando una incapacidad para identificar correctamente las instancias positivas y una falta de equilibrio entre precisión y sensibilidad. Aunque la especificidad y la exactitud eran altas, no compensan las deficiencias en otras áreas clave de rendimiento. Además, se llevaron a cabo varios experimentos adicionales para evaluar el impacto de aplicar transformaciones de datos durante el entrenamiento. En estas pruebas, se intentó implementar métodos de aumento de datos directamente en nuestra clase personalizada *nnUNetTrainerAneurysm* utilizando la función *get_training_transforms* provista por la red, pero sin aplicar transformaciones adicionales a las imágenes. Como era de esperar, estos modelos mostraron un rendimiento inferior en comparación con aquellos que incorporaron aumento de datos. Estas pruebas fueron diseñadas específicamente como técnicas y no se consideran representativas en términos de rendimiento final. La utilización de un número de iteraciones por época tan bajo también influyó en los resultados de estas evaluaciones, confirmando que las transformaciones y el número adecuado de iteraciones son claves en el rendimiento del modelo.

En otra variante, se copiaron las funciones del entrenador *nnUNetTrainerDA5*, que habían mostrado buenos resultados en el Modelo 1, al entrenador personalizado. Se modificó el porcentaje de sobremuestreo de fondo para ajustar la atención del modelo hacia los píxeles correspondientes al dispositivo endoscópico. Este parámetro controla el porcentaje de sobremuestreo aplicado a las regiones de primer plano durante el entrenamiento, siendo útil para abordar desequilibrios en las clases. Se ajustó el valor de 0.33 a 0.5 para aumentar el enfoque del modelo en las regiones relevantes para la detección del aneurisma, manteniendo la cantidad de iteraciones por época en 3. Los resultados de la inferencia fueron desfavorables debido a la baja cantidad de iteraciones, limitando el aprendizaje efectivo del modelo. Sin embargo, este experimento mostró una ligera mejora en comparación con otros anteriores. La tasa de error de clasificación siguió siendo baja, pero las métricas de sensibilidad y precisión aún no alcanzaron niveles satisfactorios. El índice DICE también fue bajo, indicando que el modelo tiene dificultades para identificar correctamente las regiones de interés. A pesar de ello, la exactitud y especificidad permanecieron altas, sugiriendo que el modelo es bueno para identificar las regiones negativas.

MODELO 5

En este modelo específico, se conservaron las transformaciones predeterminadas y se extendió la cantidad de épocas de entrenamiento a 400 (ver tabla 8). El propósito detrás de esta extensión fue desarrollar una curva de aprendizaje más detallada y alcanzar un equilibrio óptimo en el volumen de entrenamiento del modelo.

Entrenamiento Modelo 5	
Parámetro	Valor
Configuración	3d_fullres
Entrenador	nnUNetTrainerAneurysm
Tamaño del lote	2
Épocas	400
Iteraciones por época	3
Tasa de aprendizaje inicial	0.01
Porcentaje de sobremuestreo de fondo	0.33
Decaimiento del peso	3,00E-05
Pliegues entrenados	[0]
Multietiqueta	FALSE

Tabla 8: Configuración utilizada para el entrenamiento del Modelo 5.

A partir de las 250 épocas, se observó (ver figura 23) que el coeficiente pseudo DICE comenzaba a descender, lo que sugiere una posible estabilización o incluso sobreajuste del modelo en ese punto. Esta conclusión se acentúa además al ver que la curva de pérdida con los datos de validación (*loss_val*) comienza a ascender.

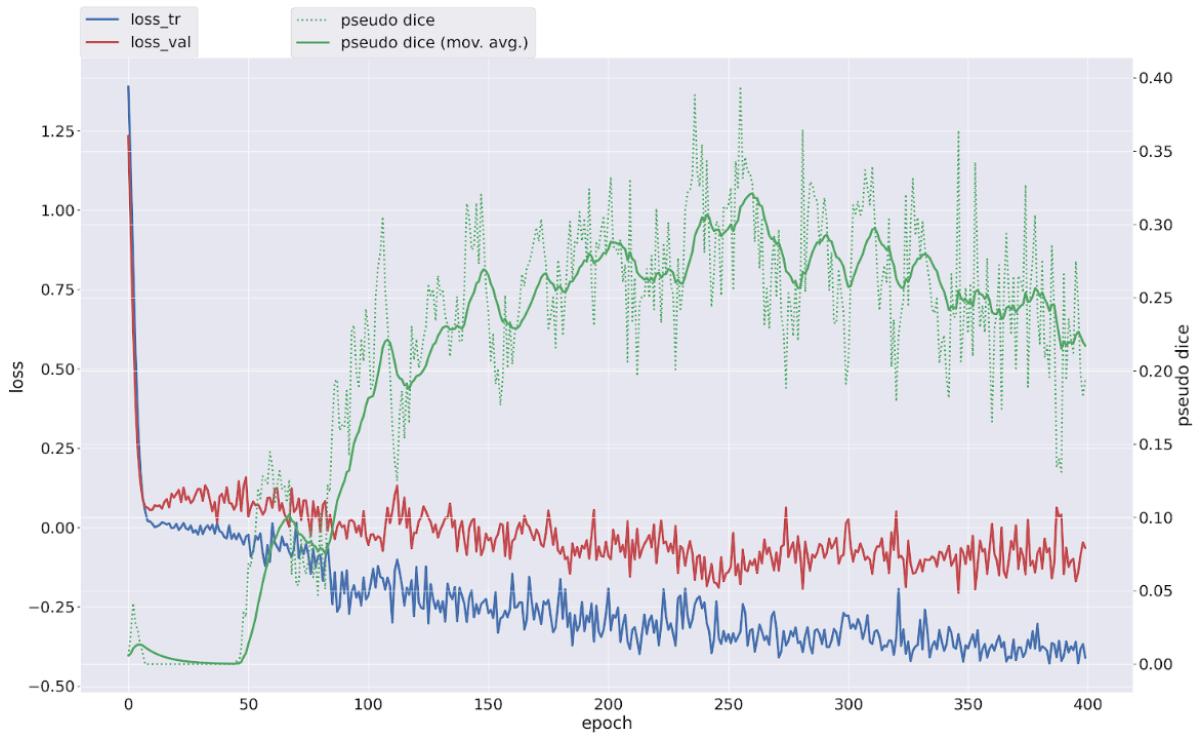


Figura 23: Gráfico del entrenamiento del Modelo 5 entrenado con 400 épocas.

Los resultados de la inferencia no fueron satisfactorios (ver figura 24), lo que llevó a la decisión de descartar el modelo. No obstante, es importante destacar que el propósito principal de este entrenamiento era determinar la cantidad óptima de épocas.

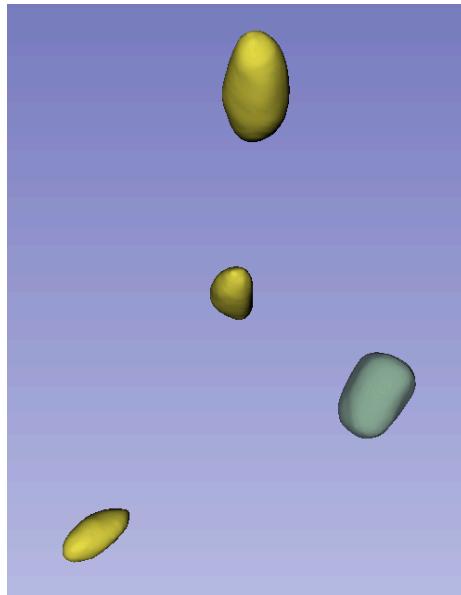


Figura 24: Vista superior de la inferencia obtenida utilizando el pliegue 0 en contraste con la segmentación original.

Haciendo un análisis de las métricas cuantitativas (ver tabla 9), la tasa de error de clasificación es bastante baja, lo cual parece prometedor, pero otras métricas como el índice DICE, la sensibilidad, la precisión y el índice DICE son extremadamente bajos sugiriendo dificultades para identificar correctamente las clases de interés en los datos de evaluación.

Modelo 5 - Fold 0	
Métrica	Valor
Tasa de Error de Clasificación	0,000493
Índice DICE	0,038124
Sensibilidad	0,024584
Precisión	0,084870
Especificidad	0,999895
Exactitud	0,999507

Tabla 9: Métricas cuantitativas del pliegue 0 del Modelo 5.

El aumento de las épocas de entrenamiento permitió explorar más a fondo el comportamiento del modelo, pero la observación de la disminución del coeficiente pseudo DICE y el aumento de la pérdida de validación después de cierto punto sugiere la necesidad de tener precaución con el sobreajuste. Dado que el objetivo del entrenamiento era buscar la cantidad óptima de épocas, y los resultados de la inferencia no fueron exitosos, es razonable concluir que el modelo no ha alcanzado un rendimiento satisfactorio y que, por lo tanto, se justifica la decisión de descartarlo.

Una variante adicional mantuvo los mismos parámetros que en la prueba anterior, pero con una redistribución de las imágenes entre los conjuntos de datos de entrenamiento y validación. Se añadió una imagen adicional al conjunto de entrenamiento y se eliminó una imagen del conjunto de validación para cada pliegue del modelo. Este ajuste en la distribución de los datos tenía como objetivo evaluar el impacto de una mayor cantidad de datos de entrenamiento en el rendimiento del modelo. Sin embargo, esta modificación no logró una mejora significativa en las métricas de rendimiento, sugiriendo que la adición y reducción de una imagen a los conjuntos correspondientes no contribuyeron a una mejora. A pesar de una baja tasa de error de clasificación y alta especificidad, la baja precisión y el índice DICE reflejaron una pobre superposición entre las segmentaciones predichas y las reales. La necesidad de explorar nuevos enfoques para mejorar las capacidades de segmentación del modelo persiste.

REVISIÓN DE LO APRENDIDO

Los resultados obtenidos resaltan varios puntos críticos aprendidos a lo largo de las pruebas realizadas. Uno de los factores más significativos es la importancia del número de iteraciones por época. Con solo 3 iteraciones por época, el modelo carecía de la exposición adecuada a los datos, limitando severamente su capacidad de aprendizaje. Normalmente, un

mayor número de iteraciones permite ajustes más precisos y refinados, lo que se traduce en una mejor capacidad para generalizar a nuevos datos.

Además, las funciones del entrenador *nnUNetTrainerDA5* han tenido un impacto significativo en los modelos entrenados. Estas funciones han demostrado ser fundamentales para garantizar un aumento efectivo de muestras, mejorando la capacidad del modelo para generalizar y producir resultados precisos. La incorporación de aumento de datos evidenció ser una práctica esencial para abordar desequilibrios en las clases y mejorar el rendimiento general del modelo.

Es crucial considerar también la cantidad de épocas en el entrenamiento del modelo. Aunque se ha observado que 250 épocas pueden ser adecuadas para maximizar el aprendizaje, especialmente sin aplicar aumento de datos, es posible incrementar este número ajustando ciertos parámetros. Cuando se aplica aumento de datos, un mayor número de épocas permite al modelo procesar repetidamente el conjunto de datos original, facilitando un aprendizaje más profundo y refinado. Esta repetición prolongada del proceso de aprendizaje brinda al modelo la oportunidad de capturar patrones más sutiles y complejos en los datos, mejorando su capacidad para generalizar a nuevas instancias.

Por lo tanto, al optimizar la cantidad de épocas en función de la presencia o ausencia de aumento de datos, se logra un equilibrio adecuado entre la capacidad de aprendizaje del modelo y la eficiencia computacional del proceso de entrenamiento. Con un aumento de datos tan robusto como el proporcionado por las operaciones de *nnUNetTrainerDA5*, es factible extender la cantidad de épocas a 400 y aún así obtener beneficios significativos en el aprendizaje del modelo.

En el caso de la redistribución de imágenes entre los conjuntos de entrenamiento y validación, la adición de una imagen al conjunto de entrenamiento y la eliminación de una imagen del conjunto de validación no contribuyeron a una mejora en las métricas de rendimiento por lo que se descartaron cambios en este aspecto.

El próximo entrenamiento pondrá a prueba todas las lecciones aprendidas hasta ahora, buscando crear un modelo altamente efectivo y con respuestas destacadas. Aumentar la cantidad de épocas, mantener un número adecuado de iteraciones por época y utilizar técnicas de aumento de datos robustas serán estrategias clave para optimizar los modelos y alcanzar las segmentaciones buscadas.

MODELO 10

Con el aprendizaje obtenido de las experiencias anteriores y las mejoras implementadas, el Modelo 10 representa un paso adelante en la búsqueda de resultados óptimos. Integrando las funciones clave del entrenador *nnUNetTrainerDA5* para un aumento de datos efectivo, junto con un ajuste en la cantidad de épocas adecuado a las necesidades del modelo (ver tabla 10) y la naturaleza de los datos, se espera lograr mejoras significativas en los resultados de la inferencia. Este enfoque refinado y estratégico busca maximizar el aprendizaje del modelo y su capacidad para generalizar a nuevos datos, sentando así las bases para alcanzar resultados apropiados en las tareas de inferencia planteadas.

Entrenamiento Modelo 10	
Parámetro	Valor
Configuración	3d_fullres
Entrenador	nnUNetTrainerAneurysm
Tamaño del lote	2
Épocas	400
Iteraciones por época	250
Tasa de aprendizaje inicial	0.01
Porcentaje de sobremuestreo de fondo	0.5
Decaimiento del peso	3,00E-05
Pliegues entrenados	[0,1,2,3,4]
Multietiqueta	FALSE

Tabla 10: Configuración utilizada para el entrenamiento del Modelo 10.

Después del entrenamiento, se pudo observar que el pliegue 2 (ver figuras 25 y 26) presentó los resultados cualitativos más prometedores, junto con un rendimiento excepcional en términos de métricas de evaluación (ver tabla 11). Este hallazgo señala un desempeño destacado en esta partición específica del conjunto de datos. Un análisis detallado de las métricas revela un rendimiento particularmente favorable en el Modelo 10 para el pliegue 2. Con una tasa de error de clasificación extremadamente baja de 0,000038 y un índice DICE de 0,896534, estos resultados indican una capacidad sobresaliente del modelo para clasificar con precisión y capturar la similitud entre las predicciones y los datos de referencia.

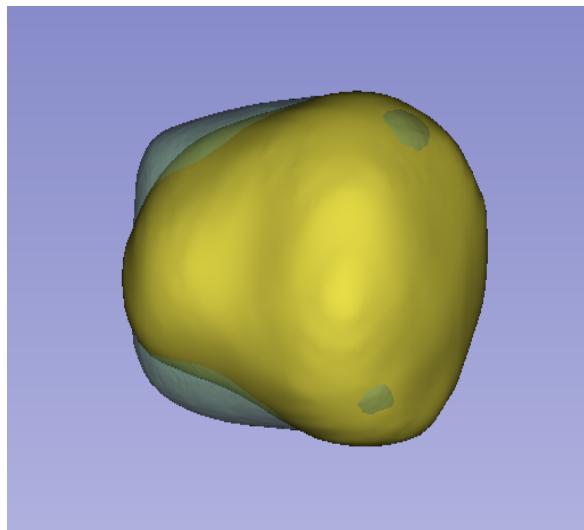


Figura 25: Vista lateral de la inferencia obtenida utilizando el pliegue 2 en contraste con la segmentación original.

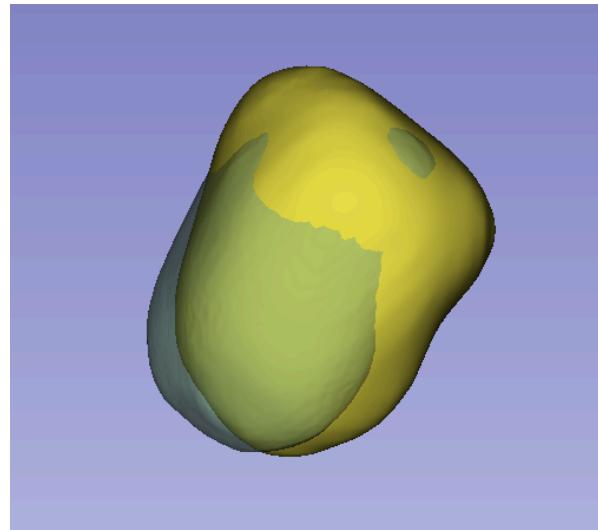


Figura 26: Vista superior de la inferencia obtenida utilizando el pliegue 2 en contraste con la segmentación original.

Modelo 10 - Fold 2	
Métrica	Valor
Tasa de Error de Clasificación	0,000038
Índice DICE	0,896534
Sensibilidad	0,913110
Precisión	0,880549
Especificidad	0,999978
Exactitud	0,999962

Tabla 11: Métricas cuantitativas del pliegue 2 del Modelo 10.

MODELO 11

Inspirado por el éxito alcanzado con el Modelo 10, se decidió replicar su configuración en un nuevo modelo, con una única variación: aumentar la cantidad de épocas a 600 (ver tabla 12). La intención era explorar si una prolongación del proceso de entrenamiento podría conducir a una mejora adicional en la curva de aprendizaje. Sin embargo, durante la evaluación del modelo resultante, se identificó un fenómeno de sobreajuste. Este fenómeno, atribuible al incremento en el número de épocas, indicaba que el modelo estaba aprendiendo a ajustarse excesivamente a los datos de entrenamiento, comprometiendo su capacidad de adaptarse a datos nuevos. A pesar de los resultados favorables observados durante la inferencia inicial, este hallazgo destaca la importancia de mantener un equilibrio entre el proceso de entrenamiento y la capacidad del modelo para generalizar.

Entrenamiento Modelo 11	
Parámetro	Valor
Configuración	3d_fullres
Entrenador	nnUNetTrainerAneurysm
Tamaño del lote	2
Épocas	600
Iteraciones por época	250
Tasa de aprendizaje inicial	0.01
Porcentaje de sobremuestreo de fondo	0.5
Decaimiento del peso	3,00E-05
Pliegues entrenados	[0,1,2,3,4]
Multietiqueta	FALSE

Tabla 12: Configuración utilizada para el entrenamiento del Modelo 11.

Para el Modelo 11 en el pliegue 2, se observan inferencias positivas (ver figuras 27 y 28) y métricas cuantitativas altamente prometedoras (ver tabla 13), reflejando un rendimiento general sólido en la tarea de clasificación. La Tasa de Error de Clasificación, que mide la proporción de instancias clasificadas incorrectamente, es excepcionalmente baja, lo que indica una alta precisión en las predicciones del modelo. El índice DICE, que evalúa la similitud entre las predicciones del modelo y los datos de referencia, también muestra un valor notablemente alto, sugiriendo una buena capacidad del modelo para capturar la forma y estructura de las instancias clasificadas.

Sin embargo, es importante destacar que estas métricas podrían verse influenciadas por el sobreajuste detectado. Para respaldar esta observación, se incluyó un gráfico (ver figura 29) que muestra, entre otras cosas, la función de pérdida en los datos de validación en relación con el número de épocas. Este gráfico revela una tendencia de aumento en la función de pérdida después de aproximadamente 400 épocas, indicando un posible inicio de sobreajuste. Además, se observa que la curva representada por la función de pseudo DICE promedio también comienza a descender a partir de este punto, lo que refuerza la idea anterior. Este hallazgo refleja el límite óptimo de aprendizaje, lo que implica que prolongar el entrenamiento más allá de este límite no resultará en mejoras significativas en la capacidad de generalización del modelo.

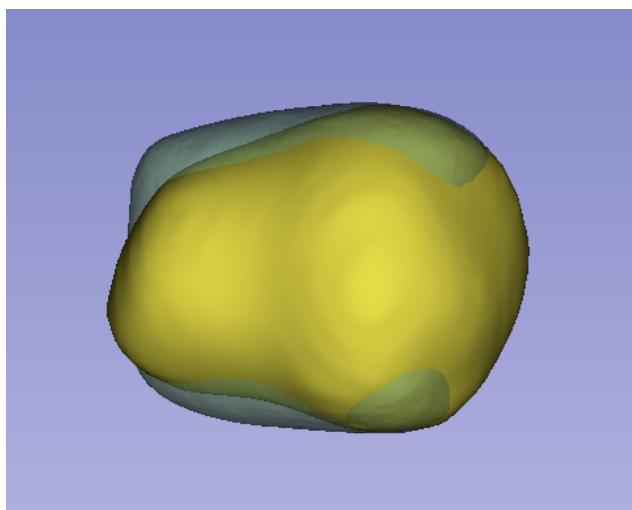


Figura 27: Vista lateral de la inferencia obtenida utilizando el pliegue 2 en contraste con la segmentación original.

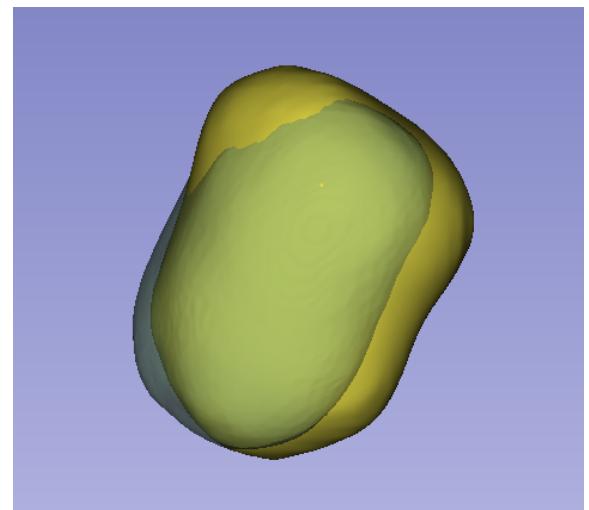


Figura 28: Vista superior de la inferencia obtenida utilizando el pliegue 2 en contraste con la segmentación original.

Modelo 11 - Fold 2	
Métrica	Valor
Tasa de Error de Clasificación	0,000037
Índice DICE	0,897631
Sensibilidad	0,912428
Precisión	0,883306
Especificidad	0,999978
Exactitud	0,999963

Tabla 13: Métricas cuantitativas del pliegue 2 del Modelo 11.

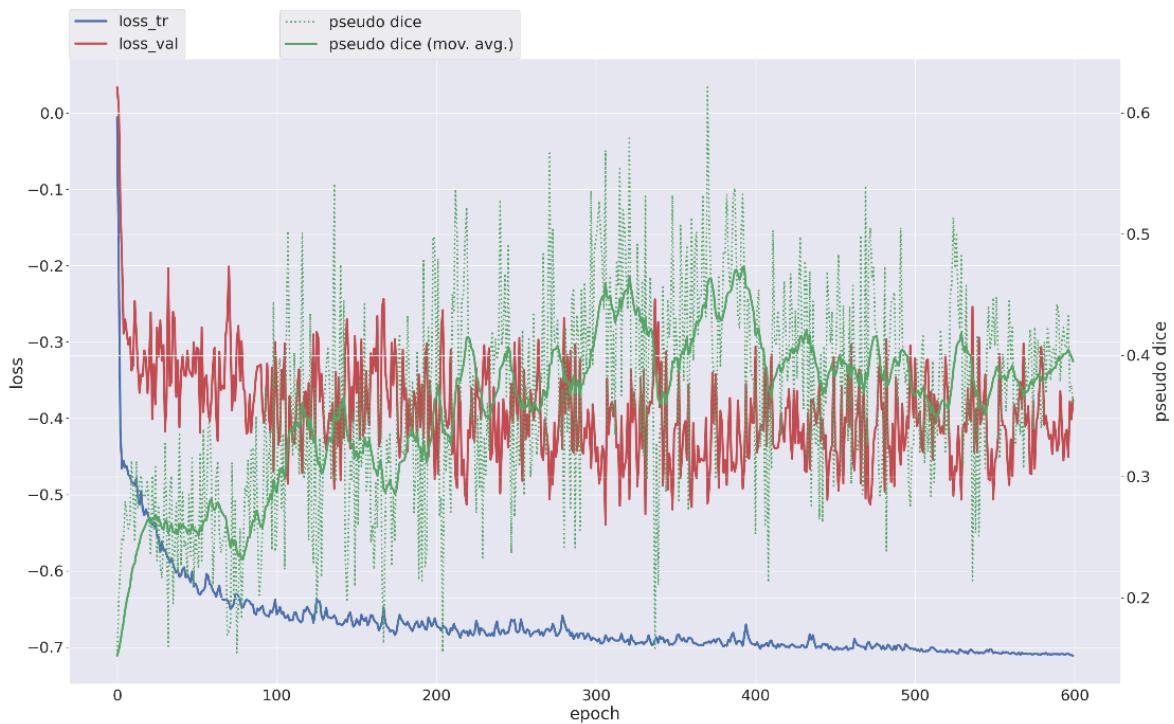


Figura 29: Gráfico del entrenamiento del Modelo 11 entrenado con 600 épocas.

RESUMEN ENTRENAMIENTOS 2

Después de identificar el sobreajuste en el modelo anterior, se tomó la decisión de modificar la configuración para una nueva prueba. En un esfuerzo por mitigar su impacto, se redujo la cantidad de épocas a 400, basándonos en la cantidad de épocas que resultaron efectivas en pruebas previas, y se aumentó el parámetro de sobremuestreo de fondo del 0.5 al 0.75. Estas modificaciones fueron implementadas con el propósito de aumentar la capacidad del modelo para detectar de manera efectiva un dispositivo de tamaño reducido en las imágenes. El incremento del sobremuestreo se orientó a enriquecer el conjunto de entrenamiento con más ejemplos del dispositivo de interés, lo cual permite al modelo capturar y generalizar mejor las características específicas de este objeto en diversas condiciones visuales. A pesar de estas medidas correctivas no se obtuvieron buenos resultados, por lo que se tomó la decisión de descartar esta configuración. Las inferencias lograron aciertos notables en el dispositivo WEB, pero el modelo cometía errores al clasificar partes del fondo (falsos positivos), reflejando desafíos persistentes en la optimización de la segmentación.

En otro intento, se optó por mantener los parámetros de configuraciones previas exitosas, realizando una modificación específica en la estructura de la red neuronal. En este sentido, se aumentó el tamaño del kernel desde el archivo *nnUNetplans.json*, pasando de un tamaño de 3x3x3 a 5x5x5 voxels. Esta decisión condujo a la creación de una nueva configuración denominada *3d_fullres_Aneurysm*, la cual hereda directamente de la configuración *3d_fullres* dada por nnUNet. Con un kernel más grande, la red neuronal puede explorar y procesar volúmenes de datos más extensos en cada paso de convolución, mejorando la capacidad de representación y potencialmente el rendimiento del modelo. El entrenamiento no cumplió con las expectativas previstas, lo que llevó al descarte de esta configuración.

Para otra prueba, se decidió realizar un ajuste específico en el parámetro de descomposición del peso (*weight_decay*). En lugar de utilizar el valor previo de 3e-5, se redujo a 3e-6. Modificar este parámetro intentó mejorar la regularización del modelo, ayudando a evitar el sobreajuste. Aunque el modelo logró una buena sensibilidad y especificidad, la tasa de error de clasificación y otros indicadores no alcanzaron los niveles deseados, reafirmando la necesidad de seguir explorando otras configuraciones y ajustes para optimizar el rendimiento.

Se adoptó una estrategia alternativa mediante el reentrenamiento de un modelo previamente utilizado, enfocando la segmentación exclusivamente en el dispositivo intrasacular al ajustar el conjunto de etiquetas. Se mantuvo el valor del decaimiento del peso en 3e-6. Esta técnica permite capitalizar las capacidades previamente adquiridas del modelo, mejorando la eficiencia del entrenamiento y potencialmente elevando la precisión en la segmentación del dispositivo intrasacular. A pesar de no haber alcanzado resultados favorables inicialmente, el reentrenamiento no se descartó debido a posibles fallos atribuibles a la configuración, lo que sugiere la necesidad de ajustes adicionales.

Aunque los modelos no mostraron resultados cualitativos significativos durante las pruebas, es destacable que estos experimentos permitieron descartar ciertas hipótesis y ajustes en los parámetros, particularmente en relación con el decaimiento del peso y el porcentaje de sobremuestreo. Estas modificaciones, implementadas de manera específica en cada modelo, representan valiosas contribuciones a la investigación y búsqueda de un modelo óptimo para la segmentación precisa de dispositivos intrasaculares en imágenes médicas. Este proceso iterativo proporciona claridad sobre las estrategias efectivas de regularización y ajuste de hiperparámetros, además sienta las bases para futuras mejoras y refinamientos del modelo final.

MODELO 15

El Modelo 15 presenta un enfoque innovador para enfrentar los desafíos en la inferencia de la parte proximal del dispositivo intrasacular utilizado en el tratamiento de aneurismas cerebrales. Esta parte proximal es el segmento del dispositivo que cubre el cuello del aneurisma y está en contacto directo con el flujo sanguíneo. Como se observa en los resultados inferidos por los modelos anteriores, esta área representa la mayor dificultad para determinar si un voxel pertenece al dispositivo endosacular o no (ver figura 30). Para enfrentar este problema se mantuvieron los mismos hiper parámetros que el Modelo 10 y un kernel de 3x3x3 voxels, pero se aumentó la segmentación a tres etiquetas (fondo, arterias y dispositivos endosaculares), permitiendo una solución más enfocada y precisa, concluyendo en una configuración multi-etiqueta (ver tabla 14).

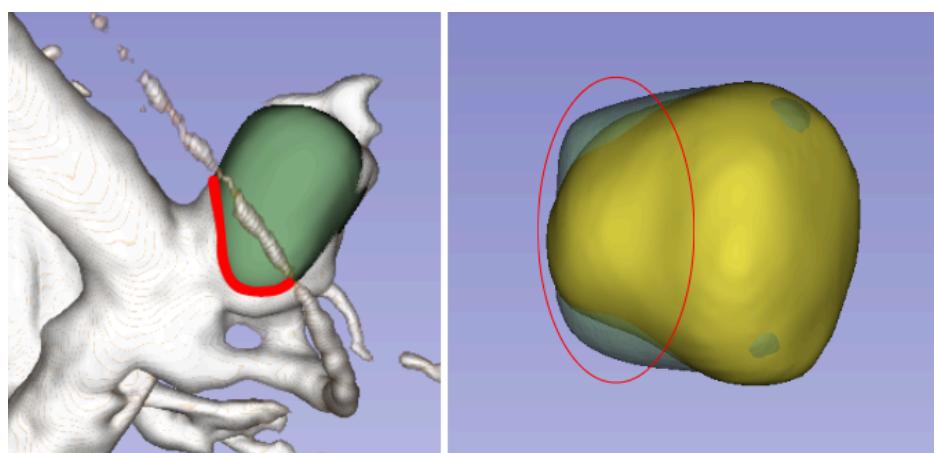


Figura 30: Se ve remarcado en rojo la parte proximal del dispositivo intrasacular.

Entrenamiento Modelo 15	
Parámetro	Valor
Configuración	3d_fullres_Aneurysm
Entrenador	nnUNetTrainerAneurysm
Tamaño del lote	2
Épocas	400
Iteraciones por época	250
Tasa de aprendizaje inicial	0.01
Porcentaje de sobremuestreo de fondo	0.5
Decaimiento del peso	3e-05
Pliegues entrenados	[0,1,2,3,4]
Multietiqueta	TRUE

Tabla 14: Configuración utilizada para el entrenamiento del Modelo 15.

Para realizar la segmentación del árbol arterial, se empleó un enfoque integral que comenzó con la exploración detallada de la imagen que incluía todos los tejidos junto con las estructuras óseas, dentro del entorno de Slicer. Primeramente con el volumen renderizado se empleó una operación *shift* a partir de la aplicación de un preset ofrecido por el software capaz de detectar arterias y mantenerlas intactas a medida que se desplazan el resto de tejidos sin importancia. Se puede ver progresivamente esta operación en las siguientes imágenes (ver figuras 31, 32, 33 y 34).

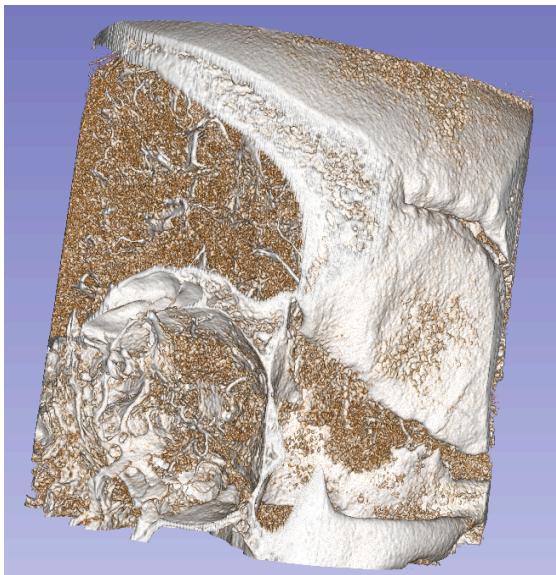


Figura 31: Imagen correspondiente al paciente 019 sin aplicación de operación de *shift*.

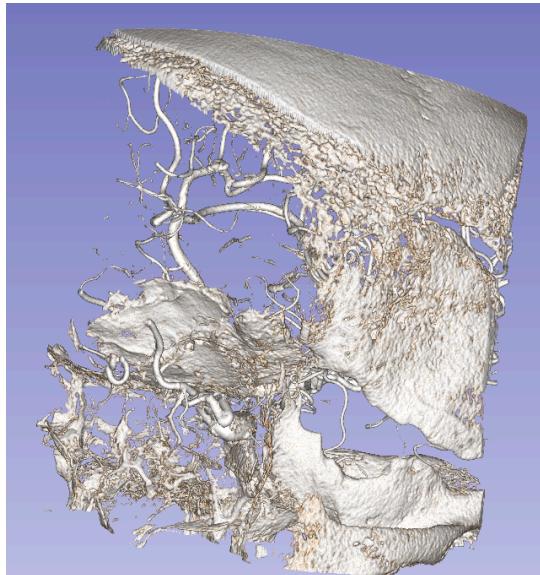


Figura 32: Imagen correspondiente al paciente 019 con aplicación de operación de *shift* parcial.

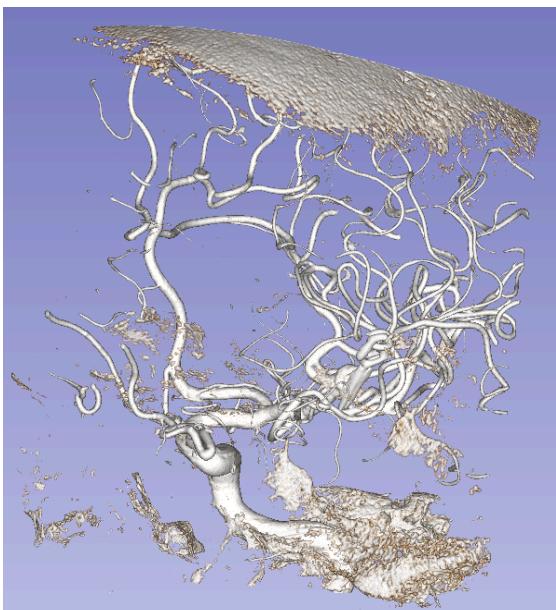


Figura 33: Imagen correspondiente al paciente 019 con aplicación de operación de *shift* parcial.

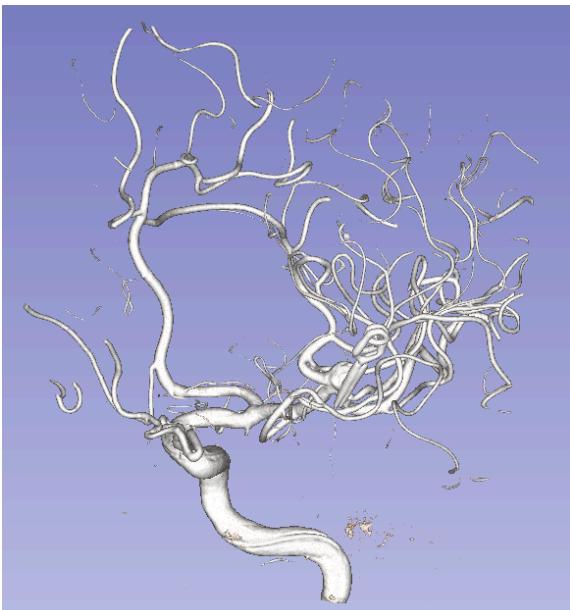


Figura 34: Imagen correspondiente al paciente 019 luego de la aplicación de operación de *shift*.

Después de eliminar los tejidos no deseados, se procedió a utilizar técnicas avanzadas de segmentación automática y semi-automática disponibles en la plataforma. La herramienta *threshold* dentro del software en el apartado de edición de segmentaciones, permitió realizar la segmentación basada en el rango de intensidad del volumen de origen. Se delimitaron meticulosamente las arterias para asegurar una cobertura completa y precisa del árbol arterial dentro del entorno tridimensional de las imágenes. Este enfoque garantiza la captura de todas las ramificaciones arteriales relevantes, esenciales para la planificación y ejecución precisa de las operaciones de sustracción y refinamiento de segmentación posteriores. Simultáneamente, se llevó a cabo un meticuloso proceso de edición utilizando las tijeras y el selector de islas, para corregir las segmentaciones arteriales incorrectas que podrían haber interferido con el entrenamiento del modelo. El resultado se observa a continuación (ver figuras 35 y 36).

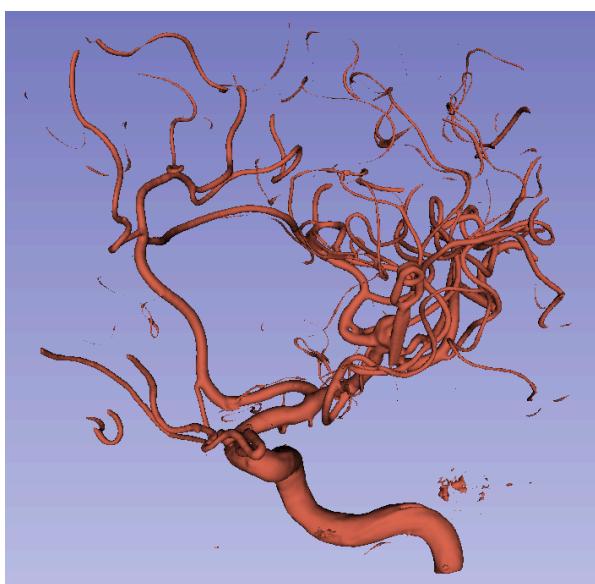


Figura 35: Segmentación arterial parcial obtenida de la imagen del paciente 019 sin ediciones agregadas.

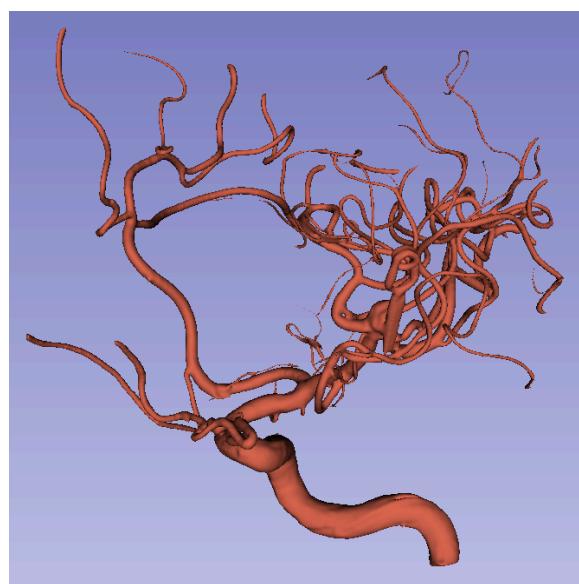


Figura 36: Segmentación arterial final obtenida de la imagen del paciente 019 con edición para eliminar detalles.

Luego del proceso anterior, todavía queda un problema pendiente de abordar: la superposición de voxels entre la segmentación arterial y la del dispositivo de tratamiento. El problema se puede observar mejor en las siguientes imágenes (ver figuras 37 y 38).

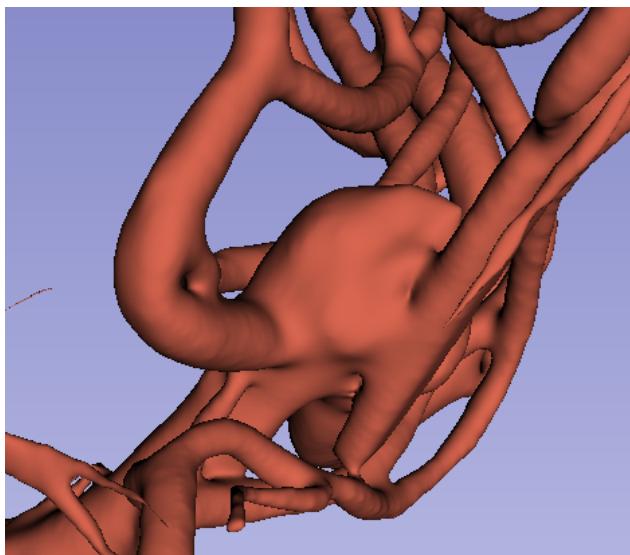


Figura 37: Detalle de la segmentación arterial donde se visualiza la zona del aneurisma tratado sin la segmentación del dispositivo ni sustracción aplicada.

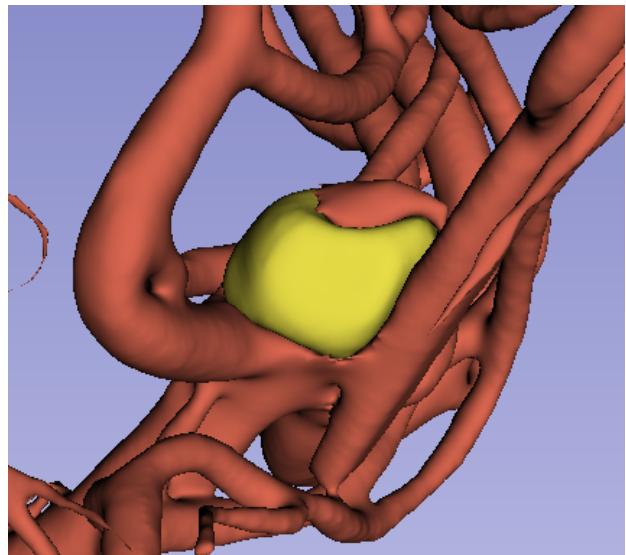


Figura 38: Detalle de la segmentación arterial donde se visualiza la zona del aneurisma tratado junto a la segmentación del dispositivo pero sin sustracción aplicada.

Con el fin de enfrentar esta compleja situación se hizo uso de las capacidades de operaciones booleanas de Slicer, llevando a cabo sustracciones precisas para eliminar selectivamente los voxels correspondientes a la arteria que coincidían con el dispositivo WEB. Esto aseguró la integridad de este último sin comprometer su precisión, facilitando así un proceso de preprocesamiento más eficaz y preciso para las imágenes del conjunto de datos. Finalmente se obtuvo el siguiente resultado (ver figuras 39 y 40).

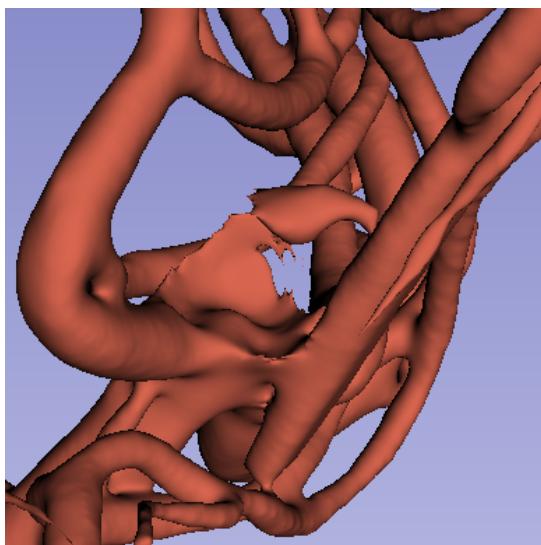


Figura 39: Detalle de la segmentación arterial donde se visualiza la zona del aneurisma tratado sin la segmentación del dispositivo pero con sustracción aplicada.

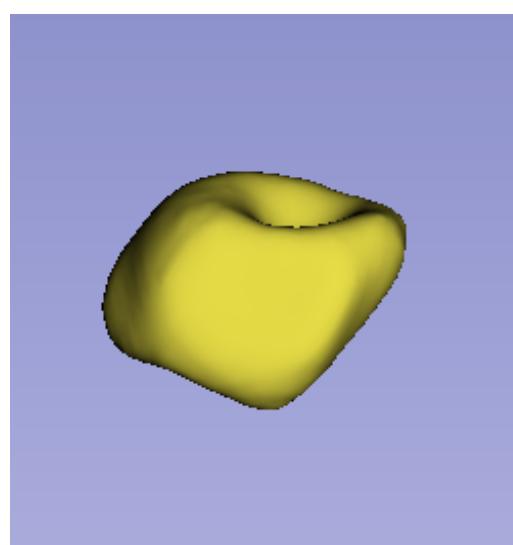


Figura 40: Detalle de la segmentación del dispositivo para el tratamiento del aneurisma cerebral del paciente 019 luego de las operaciones aplicadas. Se puede ver que no fue afectado.

Después de completar estos procesos de segmentación y sustracción para cada una de las imágenes provistas, se procedió a un segundo preprocesamiento de los datos para integrar las segmentaciones refinadas. Esto implicó la creación de un nuevo conjunto de datos unificado que combinaba las dos segmentaciones en un solo archivo, donde cada voxel estaba etiquetado según su pertenencia a las clases específicas mencionadas. Esto preparó adecuadamente el terreno para un entrenamiento óptimo del modelo, con un enfoque detallado en mejorar la precisión y consistencia en la segmentación de la parte proximal del dispositivo intrasacular.

El entrenamiento del Modelo 15 fue positivo y demostró una mejor capacidad de detección en la región de interés, como se evidencia en sus métricas destacadas (ver tabla 15). Sin embargo, se observaron diferencias significativas en el desempeño entre el Modelo 15 y su predecesor, el Modelo 10. El Modelo 10 se destacó por su alto índice DICE de 0.896534, indicando una segmentación precisa y cercana a la referencia, en comparación con el índice DICE del Modelo 15, que fue de 0.792228. Además, el Modelo 10 mostró una sensibilidad notablemente superior de 0.913110 frente a la sensibilidad de 0.756789 del Modelo 15, subrayando su habilidad para identificar correctamente los verdaderos positivos. En contraste, el Modelo 15 presentó una detección errónea significativa de dispositivo en una región identificada como fondo (ver figuras 41 y 42). Esta discrepancia se reflejó en su menor índice DICE y sensibilidad en comparación con el Modelo 10, indicando una segmentación menos precisa y una mayor discrepancia con la referencia.

Aunque el Modelo 15 supera al Modelo 10 en métricas como precisión, especificidad, tasa de error de clasificación y exactitud, las métricas clave para la segmentación de imágenes médicas, como el índice DICE y la sensibilidad, demuestran que el Modelo 10 es superior en general para tareas de segmentación precisa y fiable en aplicaciones médicas críticas. Estas diferencias subrayan la necesidad continua de explorar soluciones innovadoras para mejorar la precisión y la fiabilidad de los modelos en aplicaciones médicas, donde la exactitud y la consistencia en la detección de dispositivos es determinante para el diagnóstico y tratamiento adecuados.

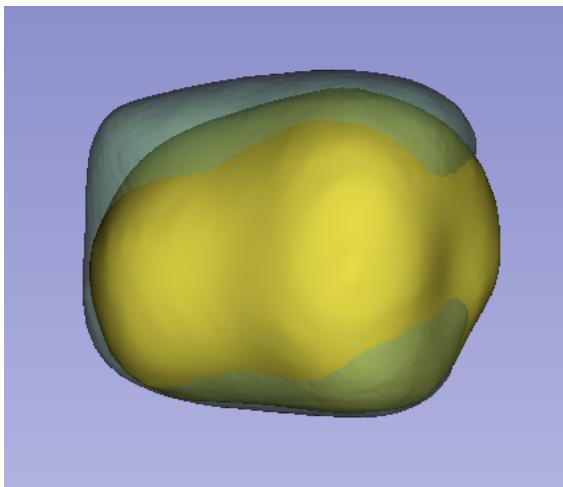


Figura 41: Vista lateral de la inferencia obtenida utilizando el pliegue 4 en contraste con la segmentación original.



Figura 42: Vista superior de la inferencia obtenida utilizando el pliegue 4 en contraste con la segmentación original.

Modelo 15 - Fold 4	
Métrica	Valor
Tasa de Error de Clasificación	0,000019
Índice DICE	0,792228
Sensibilidad	0,756789
Precisión	0,831148
Especificidad	0,999993
Exactitud	0,999981

Tabla 15: Métricas cuantitativas del pliegue 4 del Modelo 15.

RESUMEN ENTRENAMIENTOS 3

En esta sección se presentan los modelos probados posteriormente al Modelo 15, detallando las modificaciones realizadas y los resultados obtenidos. Para mejorar el trabajo y darle un ajuste final, se decidió inicialmente reentrenar el Modelo 10 modificando el tamaño del lote. Por defecto, este valor está configurado en 2, lo que implica que el modelo procesa dos ejemplos simultáneamente durante cada iteración del entrenamiento. La primera modificación consistió en reducir este valor a 1, con la expectativa de que un procesamiento más detallado y secuencial de los datos pudiera optimizar el aprendizaje del modelo. Sin embargo, los resultados obtenidos fueron negativos, sugiriendo que esta configuración no proporcionó los beneficios esperados.

En la siguiente fase, se optó por aumentar el lote a 3. Incrementar este valor permite al modelo promediar gradientes sobre más ejemplos en cada iteración, lo que puede llevar a una convergencia más estable y rápida. Los resultados de esta configuración fueron muy positivos, superando incluso al Modelo 10 original y estableciéndose como el mejor modelo hasta el momento. No obstante, se decidió continuar con los experimentos para intentar

perfeccionar aún más el rendimiento del modelo. Se prosiguió, reentrenando el Modelo 10, manteniendo el batch size en 2, pero aumentando la cantidad de iteraciones por época a 300 (valor por defecto de 250). Aunque esta modificación produjo resultados positivos, se detectó un problema de sobreajuste, donde el modelo mostraba un buen rendimiento en el conjunto de entrenamiento, pero no lograba generalizar adecuadamente en datos no vistos.

Para concluir, después de un extenso proceso de ajustes y reentrenamientos, se alcanzaron conclusiones clave sobre las configuraciones óptimas que darán paso al modelo final. A través de la combinación de diversas estrategias de modificación de hiperparámetros, arquitectura de la red, y técnicas de aumento de datos, se identificó un modelo que no solo supera a sus predecesores en términos de precisión y capacidad de generalización, sino que también se establece como el resultado más efectivo de este trabajo. En la siguiente sección, se presenta el modelo final obtenido, destacando sus características y sus implicaciones para futuras aplicaciones y desarrollos en el campo.

MODELO FINAL

Finalmente, se entrenó el último modelo, aquel que cumple con las expectativas previstas y realiza segmentaciones altamente precisas. Antes de pasar con los resultados, se adjuntan dos tablas. Una de ellas con todos los cambios que se fueron realizando a lo largo de este proyecto, en él se reflejan los modelos exitosos, los fallidos y las modificaciones realizadas en cada uno de los mismos. (ver tabla 16).

	Cantidad de épocas	Tamaño del lote	Iteraciones por época	Porcentaje de sobremuestreo	Decaimiento del peso	Multietiqueta	Aumento de datos	Tamaño del kernel
Modelo 0	200	2	250	0,33	3,00E-05	No	No	3x3x3
Modelo 1	200	2	250	0,33	3,00E-05	No	Si	3x3x3
Modelo 2	100	2	250	0,33	3,00E-05	No	Si	3x3x3
Modelo 3	100	2	3	0,33	3,00E-05	No	Si	3x3x3
Modelo 4	100	2	3	0,33	3,00E-05	No	No	3x3x3
Modelo 5	400	2	3	0,33	3,00E-05	No	Si	3x3x3
Modelo 6	400	2	3	0,33	3,00E-05	No	Si	3x3x3
Modelo 7	400	2	3	0,33	3,00E-05	No	No	3x3x3
Modelo 8	400	2	3	0,5	3,00E-05	No	Si	3x3x3
Modelo 9	400	2	3	0,33	3,00E-05	No	Si	3x3x3
Modelo 10	400	2	250	0,5	3,00E-05	No	Si	3x3x3
Modelo 11	600	2	250	0,5	3,00E-05	No	Si	3x3x3
Modelo 12	400	2	250	0,75	3,00E-05	No	Si	3x3x3
Modelo 13	400	2	250	0,5	3,00E-05	No	Si	5x5x5
Modelo 14	400	2	250	0,5	3,00E-05	No	Si	3x3x3
Modelo 15	400	2	250	0,5	3,00E-05	Si	Si	3x3x3
Modelo 16	400	2	250	0,75	3,00E-06	Si	Si	3x3x3
Modelo 17	200	2	250	0,75	3,00E-06	No	Si	3x3x3
Modelo 18	400	2	250	0,75	3,00E-06	No	Si	3x3x3
Modelo 19	400	1	250	0,5	3,00E-05	No	Si	3x3x3
Modelo 20	400	3	250	0,5	3,00E-05	No	Si	3x3x3
Modelo 21	400	2	300	0,5	3,00E-05	No	Si	3x3x3
Modelo Final	400	3	275	0,5	3,00E-05	No	Si	3x3x3

Tabla 16: Configuración de todos los modelos entrenados. Modificaciones detalladas en color amarillo, modelos fallidos en rojo y exitosos en verde.

La otra tabla refleja una comparación de métricas cuantitativas (ver tabla 17). Se definieron una serie de umbrales para identificar a los modelos como exitosos en base a las cualidades más significativas. Estos umbrales son:

- **Tasa de error < 0.0005:** Para garantizar que el modelo comete muy pocos errores.
- **DICE > 0.85:** Para asegurar que las áreas segmentadas predichas coinciden en al menos un 85% con las áreas reales, lo que implica alta calidad de segmentación.
- **Sensibilidad > 0.9:** Para asegurar que el modelo detecta al menos el 90% de las áreas importantes, minimizando falsos negativos.
- **Precisión > 0.85:** Para garantizar que al menos el 85% de las áreas segmentadas predichas son correctas, minimizando falsos positivos.

Modelos		Tasa de Error de Clasificación	DICE	Sensibilidad	Precisión	Especificidad	Exactitud
Nro. modelo	Pliegue						
Modelo 0	2	0,000044	0,879339	0,884399	0,874337	0,999977	0,999956
Modelo 1	4	0,000022	0,721604	0,619576	0,86386	0,999995	0,999978
Modelo 2	0	0,00032	0,391244	0,258824	0,801107	0,999974	0,99968
Modelo 3	0	0,00084	0,071009	0,080831	0,063315	0,999525	0,99916
Modelo 5	0	0,000493	0,038124	0,024584	0,08487	0,999895	0,999507
Modelo 6	1	0,000603	0,270825	0,620707	0,173197	0,999466	0,999397
Modelo 8	2	0,000726	0,146124	0,346134	0,09261	0,999391	0,999274
Modelo 10	2	0,000038	0,896534	0,91311	0,880549	0,999978	0,999962
Modelo 11	2	0,000037	0,897631	0,912428	0,883306	0,999978	0,999963
Modelo 12	0	0,00147	0,21152	0,496095	0,134416	0,998730	0,99853
Modelo 14	2	0,000041	0,890508	0,920637	0,862289	0,999974	0,999959
Modelo 15	2	0,000019	0,792228	0,756789	0,831148	0,999993	0,999981
Modelo 16	4	0,001141	0,88161	0,908747	0,856047	0,999282	0,998859
Modelo 19	0	0,00115	0,894245	0,906101	0,882696	0,999350	0,99885
Modelo Final	2	0,000033	0,912255	0,952908	0,874928	0,999976	0,999967
Umbral		< 0,00005	> 0,85	> 0,9	> 0,85	> 0,99	> 0,99
						Todos los modelos se encuentran por encima del umbral	

Tabla 17: Métricas cuantitativas de los modelos entrenados. Destacando en verde aquellos valores que superan el umbral definido.

Para el modelo final, abordando el sobreajuste del Modelo 21, se llevó a cabo una nueva prueba combinando los análisis de los modelos previos. Se disminuyó la cantidad de iteraciones por época a 275 y se aumentó el tamaño de lote a 3. Esta prueba se realizó utilizando el conjunto de datos sin segmentaciones de arterias, buscando encontrar un equilibrio que mejorará la capacidad de generalización del modelo. Los resultados de esta prueba fueron muy buenos, con métricas e inferencias satisfactorias (ver tabla 18). Este modelo alcanza y supera los objetivos propuestos, consolidándose como el punto culminante de nuestra investigación y el mayor logro del proyecto.

Modelo Final - Fold 2	
Métrica	Valor
Tasa de Error de Clasificación	0,000033
Índice DICE	0,912255
Sensibilidad	0,952908
Precisión	0,874928
Especificidad	0,999976
Exactitud	0,999967

Tabla 18: Métricas cuantitativas del pliegue 2 del Modelo Final

Entre las métricas se destaca la especificidad del modelo, que alcanza un valor de 0.999976, indicando una capacidad excepcional para distinguir correctamente entre el

dispositivo de interés y el fondo. La alta exactitud de 0.999967 refleja la precisión general del modelo en la clasificación de los datos.

El fold 2 de este modelo presenta un gráfico favorable (ver figura 43) y coherente con las métricas en el que se observa la curva de pérdida de validación en continuo descenso. Esta tendencia indica una mejora constante del modelo durante el proceso de entrenamiento, demostrando la ausencia de sobreajuste. Además, la curva de pseudo DICE se mantiene en aumento a lo largo de las 400 épocas, lo que evidencia una mejora progresiva en el aprendizaje del modelo. Este gráfico es una prueba más del alto rendimiento alcanzado, validando el enfoque y las decisiones tomadas para esta última prueba.



Figura 43: Gráfico del entrenamiento del Fold 2 del Modelo Final.

Para finalizar, a partir de las métricas y el gráfico presentado, se da paso a las inferencias realizadas por el modelo, en las que se distingue a simple vista que la tarea de segmentación ha sido superior a los resultados obtenidos con el resto de los modelos presentados (ver figuras 44 y 45). Las imágenes resultantes muestran con claridad la capacidad del modelo para identificar y segmentar el dispositivo intrasacular con alta precisión. Cada inferencia refleja la robustez del modelo para generalizar en diferentes condiciones, manteniendo una consistencia notable en la segmentación. Estos resultados visuales confirman la eficacia del modelo final y consolidan el logro de los objetivos propuestos, demostrando la finalización de este trabajo y el impacto positivo de las mejoras implementadas en el proceso de entrenamiento.

Los resultados obtenidos del modelo final representan la culminación de esta investigación exhaustiva y el esfuerzo sostenido a lo largo de este proyecto en la búsqueda de soluciones precisas y eficaces en el campo de la segmentación médica.

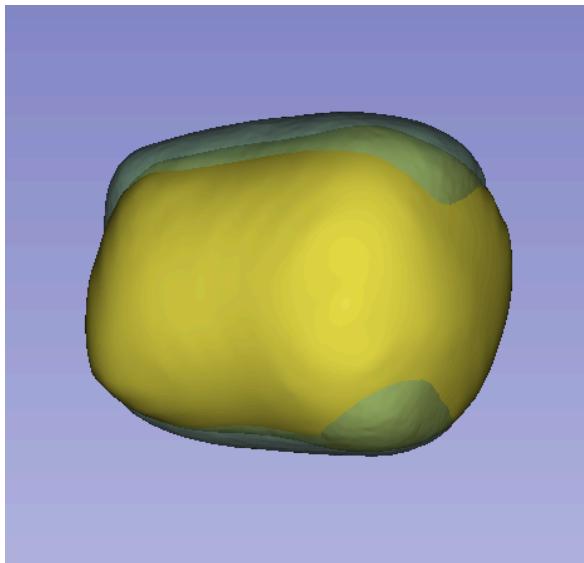


Figura 44: Vista lateral de la inferencia obtenida utilizando el pliegue 2 en contraste con la segmentación original.

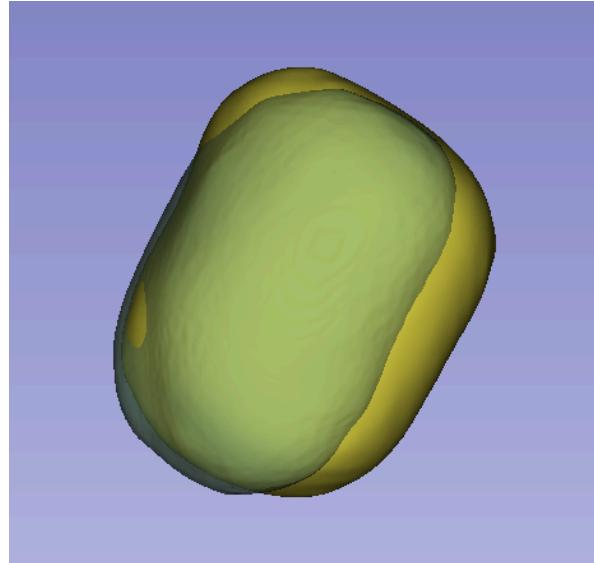


Figura 45: Vista superior de la inferencia obtenida utilizando el pliegue 2 en contraste con la segmentación original.

SERVICIO

Para concluir el presente trabajo, se desarrolló un servicio web que ofrece una interfaz intuitiva para realizar inferencias basadas en los modelos de mayor precisión para el análisis de imágenes médicas. Este servicio simplifica el acceso y uso de dichos modelos a través de cualquier navegador web, permitiendo a los usuarios cargar imágenes y obtener segmentaciones en tiempo real de manera eficiente. La implementación del servicio se realizó utilizando Python y el framework Streamlit, reconocido por su capacidad para crear interfaces de usuario rápidas y funcionales [7]. El servicio está diseñado para aceptar imágenes en formato .nii.gz y proporciona resultados con alta precisión. La interfaz web facilita una interacción fluida y eficiente con el modelo, permitiendo la aplicación de segmentación automática a conjuntos de datos propios de los usuarios con gran facilidad.

Este servicio está restringido para su uso exclusivamente dentro de la red de Pladema, lo que asegura la seguridad y el control de acceso a la aplicación. De este modo, sólo los usuarios autorizados dentro de la red tienen acceso a la herramienta, garantizando la confidencialidad de los datos médicos procesados. La implementación de este servicio no solo cierra el ciclo del trabajo de investigación al poner el modelo a disposición de otros usuarios de manera accesible, sino que también resalta el potencial de la inteligencia artificial en el ámbito médico mediante soluciones prácticas y orientadas al usuario.

SECCIONES

El servicio está estructurado en tres páginas principales: la página principal, la página de modelos utilizables y una página informativa sobre el servicio. A continuación, se presenta un desarrollo detallado de cada una de estas secciones.

La página principal (ver figura 46) está diseñada para proporcionar una experiencia de usuario clara y eficiente en el proceso de solicitud de inferencias. En esta sección, los usuarios pueden cargar imágenes médicas en formato .nii.gz directamente desde su dispositivo. La interfaz, concebida para ser intuitiva y fácil de usar, guía al usuario a través del proceso de carga de datos y selección del modelo de inferencia. Una vez que el usuario ha cargado los datos, puede elegir el modelo de inferencia más adecuado para su caso específico. La página principal está equipada con indicaciones visuales y mensajes de retroalimentación que aseguran que cada paso se realice de manera correcta y que los datos sean procesados sin errores.

La segunda página (ver figura 47) ofrece una visión general de los modelos disponibles para la inferencia. En esta sección, los usuarios pueden consultar una lista detallada de los modelos implementados, incluyendo sus características y especificaciones técnicas. Cada modelo está acompañado de una breve descripción que facilita la elección del más adecuado para las necesidades específicas del usuario. Esta página también incluye opciones para comparar diferentes modelos y acceder a documentación adicional sobre cada uno, proporcionando así la información necesaria para una toma de decisiones informada.

La última página del servicio (ver figura 48) está dedicada a proporcionar información adicional sobre el servicio en sí. En esta sección, los usuarios pueden encontrar detalles sobre

la finalidad del servicio, los beneficios que ofrece y cómo contribuye a la mejora de los procesos de análisis de imágenes médicas. Además, se incluye información sobre el equipo de desarrollo y las instituciones involucradas.

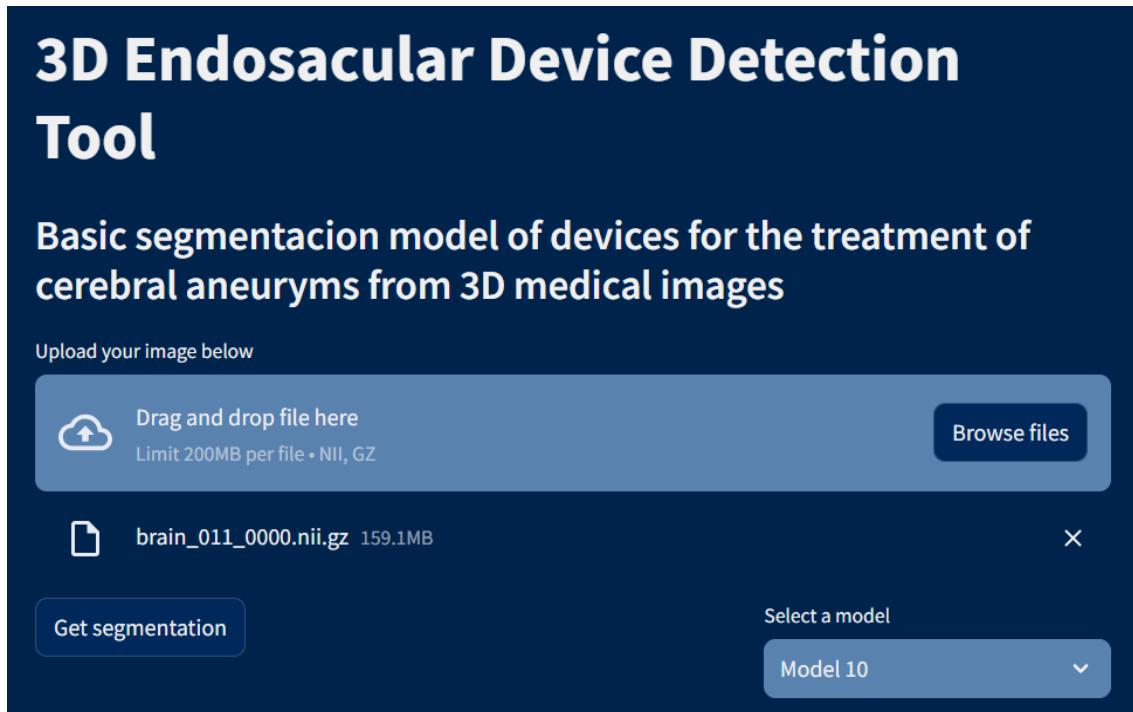


Figura 46: Página principal del servicio. Aquí se pueden cargar imágenes, seleccionar modelos y obtener segmentaciones en pocos minutos.

Model Specifications							
Basic information of each model							
Model	Classification Error	DICE	Sensitivity	Precision	F1 Score	Specificity	Accuracy
Model 10	0.000079	0.780565	0.7279169	0.9003842	0.7805647	0.9999883	0.9999208
Model 15	0.000750	0.863033	0.8424650	0.8886734	0.8630332	0.9995986	0.9992495
Model 24	0.000750	0.863033	0.8424650	0.8886734	0.8630332	0.9995986	0.9992495

i The metric values are the average of the folds

Figura 47: Página acerca de las métricas de los modelos utilizables. Se observan las especificaciones del modelo.

About

During the Supervised Professional Practice (SPP) period, a computer vision project in medicine focusing on brain aneurysm segmentation from 3D medical images was undertaken. This experience for Systems Engineers explored the application of artificial intelligence in medicine, enhancing both diagnosis and treatment. The project followed a combined approach, leveraging both in-person and remote work, with fixed schedules and seamless communication with the Yatiris team. A cyclical methodology of training and evaluation was utilized to optimize the segmentation model's performance, employing tools such as PyTorch, SLURM, and Slicer 3D.

Creators

David Burckhardt

Martin Vazquez Arispe

Dr. Ignacio Larrabide

Dra. Romina Luciana
Muñoz

Directors



Figura 48: Tercera página del servicio. Aquí se brinda información sobre el proyecto, objetivos, metodología y demás cuestiones como también los creadores y directores del mismo.

GUÍA DE USO

En esta sección, se detalla el proceso para solicitar inferencias a través de la web. Se proporciona una guía paso a paso que permite a los usuarios cargar sus datos y enviar una solicitud de inferencia. Esta guía tiene como objetivo proporcionar una comprensión clara del proceso y asegurar una interacción eficiente con el servicio.

- Carga de la Imagen:** El primer paso para solicitar una inferencia es cargar la imagen médica que se desea procesar. Para ello, debe hacer clic en el botón "*Browse files*". Esta acción abrirá el administrador de archivos del sistema operativo, permitiendo al usuario seleccionar el archivo de imagen deseado desde su dispositivo. Asegúrese de que el archivo esté en el formato correcto (.nii.gz) para garantizar que el servicio pueda procesarlo adecuadamente (ver figura 49).
- Selección del Modelo:** Una vez que la imagen haya sido cargada completamente, el siguiente paso es seleccionar el modelo de inferencia a utilizar. En la sección "*Select a model*", se presentará una lista de modelos disponibles para la inferencia. El usuario puede elegir el modelo más adecuado para su caso específico. En caso de no seleccionar ningún modelo, el sistema aplicará automáticamente el modelo que se encuentra en la primera posición en orden lexicográfico (ver figura 50).
- Solicitud de Inferencia:** Tras completar la carga de la imagen y la selección del modelo, el usuario está listo para realizar la petición de inferencia. Para proceder,

debe hacer clic en el botón "*Get segmentation*". Este botón iniciará el proceso de segmentación utilizando el modelo seleccionado y la imagen cargada (ver figura 51).

4. **Descarga de Resultados:** Finalmente, una vez que la inferencia se haya completado, el servicio ofrece la opción de descargar la imagen segmentada. Para descargar el resultado, simplemente haga clic en el botón "*Download inference*". La imagen solicitada será descargada en el formato adecuado, lista para ser revisada o utilizada según las necesidades del usuario (ver figura 52).

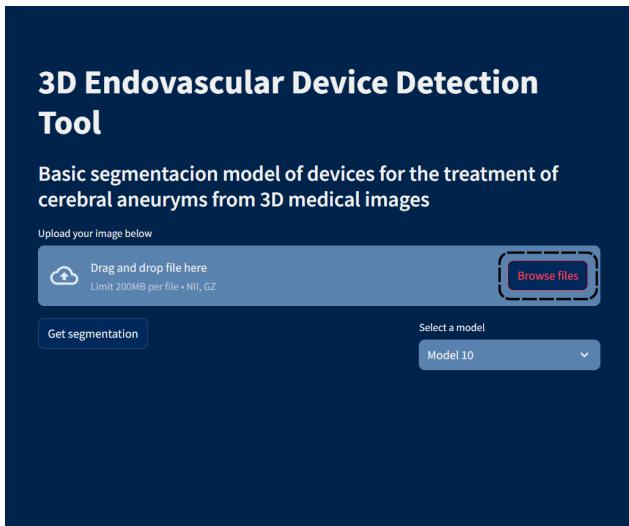


Figura 49: Captura indicando el botón para abrir el selector de archivos y cargar la imagen a procesar.

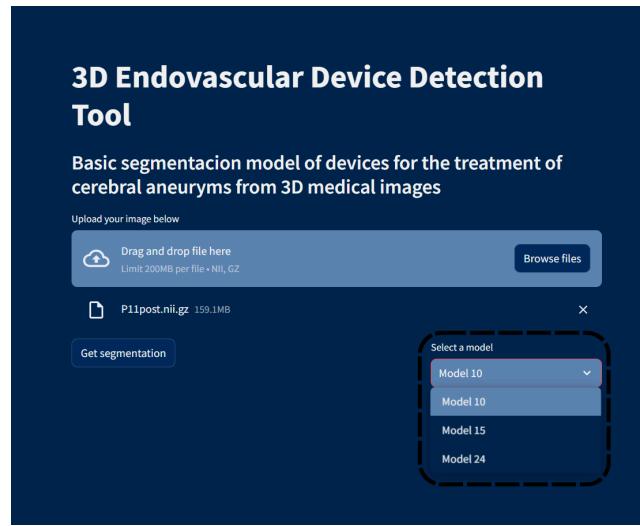


Figura 50: Captura con el selector de modelos desplegado donde se visualizan las opciones disponibles para trabajar.

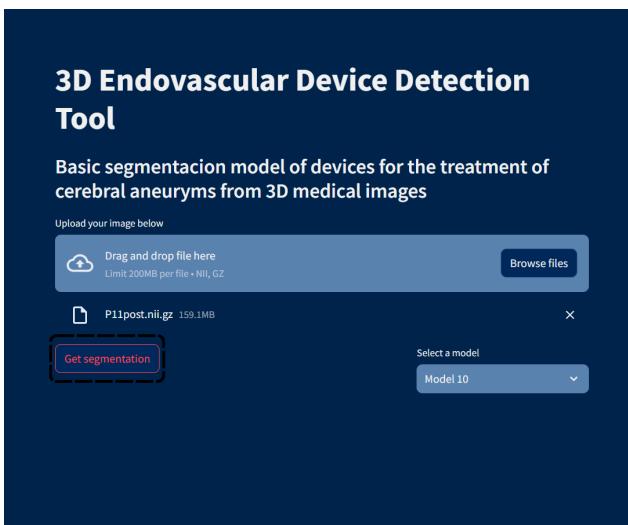


Figura 51: Captura indicando el botón que activa la inferencia y permite obtener la segmentación.

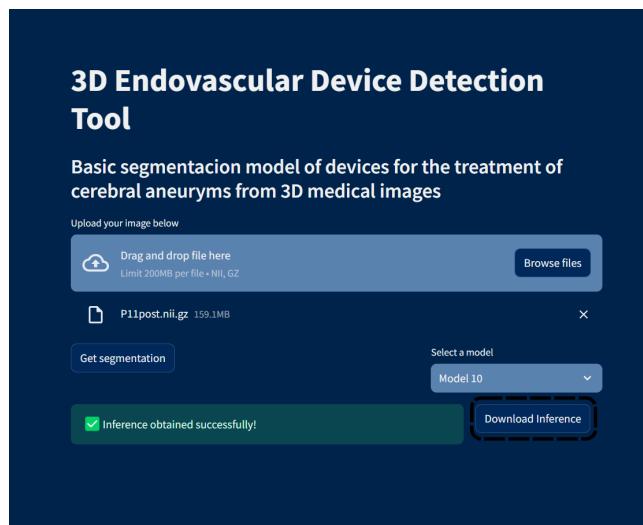


Figura 52: Captura indicando el botón para descargar la inferencia. Aquí se observa que fue obtenida con éxito.

APLICACIÓN DE CONOCIMIENTOS ACADÉMICOS

A lo largo del desarrollo de este proyecto, se integraron y aplicaron diversos conocimientos adquiridos a lo largo de toda la carrera de ingeniería, demostrando cómo cada área de estudio contribuyó de manera significativa a la realización exitosa del trabajo.

FUNDAMENTOS DE PROGRAMACIÓN Y ALGORITMOS

Desde los primeros cursos de programación, se adquirieron habilidades esenciales en lenguajes como Python y C++, las cuales fueron fundamentales para implementar y ajustar los algoritmos de aprendizaje de la red nnUNet. Los conceptos de estructuras de datos y algoritmos permitieron optimizar el procesamiento de grandes volúmenes de datos y mejorar la eficiencia del código. Se aplicaron principios de Introducción a la Programación 1 y 2 y Programación Orientada a Objetos para el manejo de clases, así como Análisis y Diseño de Algoritmos 1 y 2 para la eficiencia en la implementación.

MATEMATICAS Y ESTADISTICA

Las bases matemáticas y estadísticas fueron cruciales para comprender y aplicar técnicas de aprendizaje automático y profundo. El conocimiento en álgebra, cálculo, y probabilidad facilitó la comprensión de las redes neuronales y la evaluación de los modelos mediante métricas como DICE, sensibilidad y precisión. Se aplicaron conceptos de materias como Álgebra 1, Álgebra Lineal, Análisis Matemático 1 y 2, Matemática Discreta, Probabilidad y Estadística y Cálculo Diferencial e Integral. Particularmente, el cálculo del gradiente y la aplicación de derivadas para la retropropagación en redes neuronales son fundamentales en estos procesos.

PROCESAMIENTO DE IMÁGENES Y MANEJO DE DATOS

Otras materias de la carrera proporcionaron los fundamentos para la manipulación y análisis de imágenes médicas. Este conocimiento fue clave para administrar y preprocesar los datos, aplicar técnicas de segmentación y evaluar visualmente los resultados de los modelos. Las técnicas aprendidas en la optativa Procesamiento de Imágenes, que incluye reducción de ruido, detección de bordes, segmentación y otros procesamientos elementales, fueron directamente aplicadas como fortaleza teórica del proyecto. Además, se utilizó la compresión y descompresión de datos, comprendiendo sus efectos y consecuencias gracias a la cátedra de Teoría de la Información.

DESARROLLO DE SOFTWARE Y GESTIÓN DE PROYECTOS

Las metodologías de desarrollo de software, como el ciclo de vida iterativo incremental y las prácticas ágiles, fueron tenidas en cuenta para planificar y gestionar el proyecto de manera efectiva. La gestión de versiones mediante sistemas como Git y la documentación adecuada aseguraron un desarrollo ordenado y colaborativo. Además, se emplearon herramientas de gestión de proyectos para monitorear el progreso y cumplir con

los plazos establecidos, apoyándonos en conocimientos de Metodologías de Desarrollo de Software y Métodos Ágiles.

INTELIGENCIA ARTIFICIAL

Los cursos avanzados sobre inteligencia artificial proporcionaron el conocimiento necesario para seleccionar y configurar modelos adecuados, como el nnUNet, y para implementar técnicas de entrenamiento y validación. La comprensión de redes neuronales convolucionales y el ajuste de hiper parámetros fueron críticos para lograr modelos precisos y eficientes en la segmentación de imágenes médicas. Estos conceptos fueron fortalecidos a lo largo de la carrera en materias como Programación Exploratoria, Taller de Inteligencia Artificial y Sistemas de Recomendación.

COMUNICACION Y GESTION DE SERVIDORES

La gestión de servidores y la comunicación de datos fueron esenciales para el manejo eficiente del proyecto. Se utilizaron herramientas y técnicas aprendidas en Comunicación de Datos 1 y 2 y Sistemas Operativos, lo que permitió configurar y administrar servidores a través de VPN y Wi-Fi, asegurando una conectividad robusta y segura para el procesamiento distribuido de los datos.

La combinación de estos conocimientos y habilidades adquiridos a lo largo de la carrera permitió abordar y resolver los desafíos del proyecto final de manera integral y efectiva. Este proyecto no solo es un reflejo del aprendizaje teórico, sino también de la capacidad para aplicar estos conocimientos en situaciones prácticas y complejas, demostrando una preparación completa y robusta para enfrentar retos profesionales en el campo de la ingeniería.

CONCLUSIONES Y TRABAJO FUTURO

Este proyecto representa un avance significativo en la integración de la ingeniería de sistemas con la medicina, ofreciendo nuevas posibilidades para la detección y tratamiento de aneurismas cerebrales a través de soluciones tecnológicas innovadoras. A lo largo de esta investigación, se exploraron en profundidad las capacidades de las redes neuronales y su aplicación en el análisis de imágenes médicas, lo que condujo al desarrollo de un modelo de aprendizaje automático altamente efectivo y eficiente.

El objetivo principal fue crear un sistema capaz de identificar dispositivos endosaculares con precisión y rapidez en pacientes con aneurismas cerebrales, empleando datos reales y técnicas avanzadas de procesamiento de imágenes. Esta contribución resalta el potencial transformador de la ingeniería aplicada a la medicina, reflejando un compromiso con la innovación para mejorar los diagnósticos y tratamientos médicos. La implementación exitosa de este modelo como un servicio accesible en los servidores de Pladema representa un logro significativo, llevando la investigación más allá del ámbito académico para convertirla en una herramienta práctica en entornos clínicos reales.

Mirando hacia adelante, se vislumbran múltiples oportunidades para ampliar y mejorar este trabajo. Una dirección prometedora sería la implementación de un modelo multi-clase para la segmentación de aneurismas cerebrales, permitiendo una identificación más detallada y específica de diferentes estructuras en las imágenes médicas. Esto podría conducir a mejoras significativas en el diagnóstico y tratamiento personalizado, impactando directamente en la atención médica y la calidad de vida de los pacientes.

AGRADECIMIENTOS

Agradecemos sinceramente a todos los investigadores de Pladema por brindarnos la oportunidad de utilizar sus servidores y tecnología para llevar a cabo este proyecto. El desarrollo exitoso de esta investigación no hubiera sido posible sin su colaboración y apoyo. Queremos extender un agradecimiento especial al grupo Yatiris por facilitar la conexión a la VPN, acceso a los servidores y otros aspectos técnicos esenciales.

También agradecemos al Dr. Ignacio Larrabide por su liderazgo y seguimiento constante que permitieron la realización de este proyecto de principio a fin. Su orientación fue invaluable en cada etapa del proceso. Además, reconocemos y agradecemos a la Dra. Romina Luciana Muñoz y a la Dra. Camila García por proporcionar el dataset necesario para este estudio, su valioso apoyo y asesoramiento en todo lo relacionado con los datos y el modelo nnUNet. Su experiencia fue invaluable para el éxito de este trabajo colaborativo.

Este proyecto representa un esfuerzo conjunto y multidisciplinario que ha sido posible gracias al compromiso y la colaboración de todos los involucrados. Estamos profundamente agradecidos por la oportunidad de trabajar con un equipo tan dedicado y talentoso.

REFERENCIAS

- [1] Página web del Applied Computer Vision Lab (ACVL) del DKFZ: ACVL - Applied Computer Vision Lab. Recuperado de <https://www.dkfz.de/en/ai-lab.html>
- [2] Repositorio de nnUNet personalizado en GitHub: MartinVazquez1982/nnUNet. Recuperado de <https://github.com/MartinVazquez1982/nnUNet>
- [3] Top Doctors. Aneurisma cerebral. Recuperado el 10 septiembre de 2023, de <https://www.topdoctors.es/diccionario-medico/aneurisma-cerebral>
- [4] Stanford University. (2023, Marzo, 8). Redes Neuronales Convolucionales. Coursera. <https://www.coursera.org/learn/convolutional>
- [5] UI Health ¿Cómo se diagnostica un aneurisma cerebral? Recuperado el 10 septiembre de 2023, de <https://hospital.uillinois.edu/es/primary-and-specialty-care/neurologia-y-neurocirugia/condiciones-neur>
- [6] Resource Blog (2019, Junio, 18). First in S. Florida: New ‘WEB’ Device Closes, Heals Brain Aneurysms, <https://baptisthealth.net/baptist-health-news/new-web-device-closes-heals-brain-aneurysms>
- [7] Streamlit. Streamlit, Recuperado de: <https://streamlit.io/>
- [8] SchedMD. SLURM Documentation. Recuperado de <https://slurm.schedmd.com/documentation.html>
- [9] Mayo Clinic. Aneurysms: Symptoms and Causes. Recuperado de <https://www.mayoclinic.org/es/diseases-conditions/aneurysms/symptoms-causes/syc-20354633>.
- [10] Wikipedia. Cross-validation (statistics). Recuperado el 15 de mayo de 2024, de [https://en.wikipedia.org/wiki/Cross-validation_\(statistics\)](https://en.wikipedia.org/wiki/Cross-validation_(statistics)).
- [11] Wikipedia. Aneurisma cerebral: Clasificación por tamaño. Recuperado el 15 de mayo de 2024, de https://es.wikipedia.org/wiki/Aneurisma_cerebral#Clasificaci%C3%B3n_por_tama%C3%B1o.
- [12] Muñoz, R., Narata, A. P., Obradó, L., Fernandez, H., & Larrabide, I. (2022). Simulation of intra-saccular devices for pre-operative device size selection: Method and validation for sizing and porosity simulation. *Computers in Biology and Medicine*, 147, 105744.
- [13] Muñoz, R., Dazeo, N., Estevez-Areco, S., Janot, K., Narata, A. P., Rouchaud, A., & Larrabide, I. (2024). Modification of Woven Endo-Bridge After Intracranial Aneurysm Treatment: A Methodology for Three-Dimensional Analysis of Shape and Relative Position Changes. *Annals of Biomedical Engineering*, 52(5), 1403-1414.

-
- [14] Wiebers, D. O., International Study of Unruptured Intracranial Aneurysms Investigators, et al. (2003). Unruptured intracranial aneurysms: Natural history, clinical outcome, and risks of surgical and endovascular treatment. *The Lancet*, 362(9378), 103–110.
- [15] Narata, A. P., Moura, F., Larrabide, I., Chapot, R., Cognard, C., Januel, A. C., Velasco, S., Bouakaz, A., Patat, F., & Marzo, A. (2020). Role of distal cerebral vasculature in vessel constriction after aneurysm treatment with flow diverter stents. *Journal of NeuroInterventional Surgery*, 12(8), 818–826.
- [16] König, I., Weber, A., Weber, W., & Fischer, S. (2019). Dislocation of a WEB device into the middle cerebral artery. *Clinical Neuroradiology*, 29(2), 361–364.
- [17] Hope, J. K., Wilson, J. L., & Thomson, F. J. (1996). Three-dimensional CT angiography in the detection and characterization of intracranial berry aneurysms. *American Journal of Neuroradiology*, 17(3), 439–445.
- [18] Jeans, W. D. (1990). The development and use of digital subtraction angiography. *The British Journal of Radiology*, 63(747), 161–168.