Visión por Computador en Imágenes Médicas 3D

Carmen Azorín Martí
carmenazorin@correo.ugr.es

María Cribillés Pérez
mariacribilles@correo.ugr.es

Laura Lázaro Soraluce
lazarosoraluce@correo.ugr.es

José Manuel Román Rodríguez
joseroman03@correo.ugr.es

January 12, 2025

Abstract

Este artículo presenta un análisis de imágenes volumétricas a través de diversos tipos de redes neurales convolucionales (CNN) y segmentación de dichas imágenes. Además comparamos CNN 2D y 3D.

1 Introducción

En los últimos años, los avances en visión por computador han transformado significativamente el ámbito de la medicina, ofreciendo herramientas innovadoras para el análisis, diagnóstico y tratamiento de diversas patologías. Gracias al crecimiento exponencial de los datos médicos digitales, como imágenes de resonancia magnética (MRI), tomografías computarizadas (CT) y ultrasonidos, junto con los avances en aprendizaje profundo, la automatización de tareas complejas ha alcanzado niveles sin precedentes. Estas tecnologías han permitido no solo reducir la carga de trabajo de los especialistas médicos, sino también mejorar la precisión y la reproducibilidad de los diagnósticos.

La segmentación de imágenes, la detección de anomalías y la clasificación de tejidos son ejemplos de aplicaciones clave donde los métodos de visión por computador han mostrado resultados prometedores. En particular, la segmentación automática de estructuras anatómicas o regiones patológicas proporciona información crítica para la planificación quirúrgica, el seguimiento de enfermedades y la personalización de tratamientos.

2 Conjunto de datos y preprocesamiento

Para este proyecto se ha elegido el dataset **KiTS23**, el dataset **KiTS23 (Kidney Tumor Segmentation Challenge 2023)** es un conjunto de datos diseñado para fomentar investigaciones avanzadas en la segmentación de tumores renales a partir de imágenes médicas. Es parte de la serie de desafíos KiTS organizados para impulsar el desarrollo de modelos de aprendizaje automático en tareas relacionadas con el diagnóstico y tratamiento del cáncer de riñón.

Su principal objetivo es la segmentación de tumores renales y tejido circundante en imágenes médicas, específicamente en exploraciones de tomografía computarizada (TC).

2.1 Preprocesamiento

2.1.1 Preprocesamiento 3D

Se han definido rangos para entrenamiento, validación y test, adaptando el dataset ante la falta de imágenes de test originales.

Seleccionamos 337 casos para entrenamiento ($[0, 299] \cup [400, 436]$), 70 para validación ($[437, 506]$) y 81 para test ($[507, 587]$), excluyendo el rango inexistente $[300, 399]$. Los datos procesados se almacenan

para optimizar el uso de memoria.

Las imágenes 3D se redimensionaron a 128x128x128 mediante interpolación lineal y se normalizaron usando Z-Score, mejorando el rendimiento del modelo. Las máscaras se codificaron como enteros (0: fondo, 1: riñón, 2: cáncer, 3: quiste) para garantizar compatibilidad.

Durante el entrenamiento, los datos preprocesados se cargan en lotes adaptables, convirtiendo imágenes y máscaras en arrays numéricos para un uso eficiente de memoria.

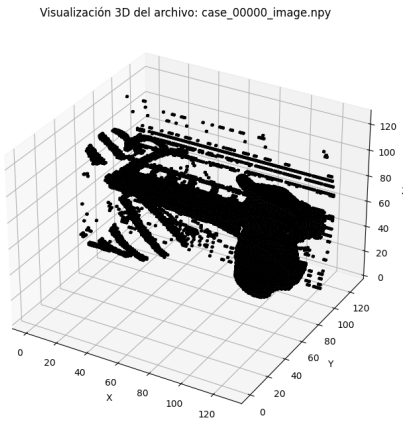

Figure 1: Visualización 3D

2.1.2 Cortes en 2D

Para sacar las slices 2D, hemos partido de las imágenes 3D ya normalizadas. Para hacer los cortes 2D a partir de datos volumétricos 3D, lo que hemos hecho es dividir cada volumen (imagen y máscara) en planos bidimensionales a lo largo de un eje específico (axial, sagital o coronal). Nosotros hemos elegido hacer el estudio con el plano axial ya que lo hemos visto el más intuitivo y el que más se suele utilizar en la literatura. Para cada sección, se seleccionan las correspondientes slices de la imagen y la máscara, que se guardan como archivos individuales en formato *.npy*. Esto permite convertir datos 3D en un formato 2D más manejable para entrenar modelos que trabajan con imágenes bidimensionales, como en tareas de segmentación médica.

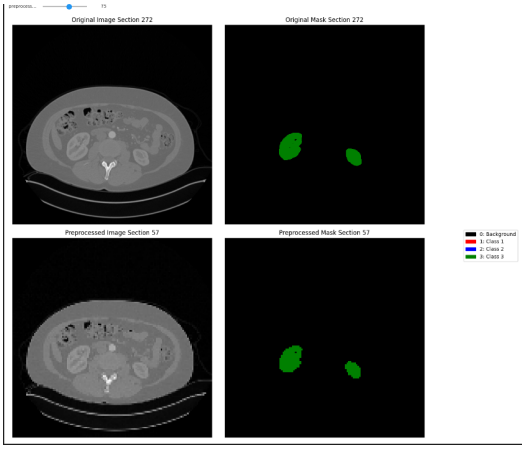

Figure 2: Cortes equidistantes axiales 2D

2.2 Sobremuestreo

Hemos implementado sobremuestreo (oversampling) para imágenes 2D y volúmenes 3D en el conjunto de entrenamiento, con el objetivo de abordar el desbalance extremo de clases en el dataset. La clase dominante (fondo) tiene significativamente más píxeles que las clases minoritarias (riñón, tumor, quiste), lo que podría llevar al modelo a favorecer predicciones del fondo, afectando su capacidad para segmentar correctamente las clases más relevantes.

Para contrarrestar este problema, se asignaron pesos de muestreo a cada ejemplo según la frecuencia de las clases en su máscara de segmentación. Los pesos por clase se calculan como:

$$w_c = \frac{1}{n_c}$$

donde n_c es el número de píxeles de la clase c. A partir de esto, se calcula un peso para cada máscara basado en las clases presentes:

$$w_{ejemplo} = \sum_{c \in Clases-presente} w_c \cdot n_c$$

Adicionalmente, se ha probado submuestreo (subsampling) para reducir la presencia de la clase dominante, seleccionando menos imágenes de esta clase. Aunque útil, esta técnica puede ser menos efectiva si

la clase dominante contiene información importante para el contexto global.

3 Funciones de pérdida

Para la segmentación de imágenes médicas se emplean redes neuronales profundas, donde la función de pérdida define el objetivo de optimización y afecta la convergencia del modelo. A continuación, se describen las funciones de pérdida usadas en nuestros modelos:

3.1 Combined Loss: Dice Loss + Cross-Entropy

La función de pérdida combinada combina las ventajas del Dice Loss y Cross-Entropy, siendo útil en problemas de segmentación de imágenes con desbalance entre clases.

3.1.1 Dice Loss

El Dice Loss está basado en el índice Dice, que evalúa la superposición entre las predicciones del modelo (P) y las etiquetas reales (T).

El coeficiente Dice se calcula como:

$$Dice(P,T) = \frac{2\sum P_i T_i}{\sum P_i + \sum T_i}$$

Donde P_i son las predicciones del píxel i y T_i las etiquetas reales.

La pérdida se define como:

$$DiceLoss = 1 - Dice(P,T)$$

En segmentación multiclase, se calcula para cada clase y se promedia:

$$DiceLoss_{total} = 1 - \frac{1}{C}\sum_{c=1}^{C} Dice_c$$

Esto asegura la optimización del modelo en todas las clases, incluso en casos de desbalance.

3.1.2 Cross-Entropy Loss

La Cross Entropy Loss mide la discrepancia entre las probabilidades predichas por el modelo y las etiquetas reales, adaptándose para segmentación a nivel de píxel.

Se calcula como:

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

Donde N es el número total de píxeles, $y_{i,c}$ la etiqueta real del píxel i para la clase c, y $p_{i,c}$ la probabilidad predicha para la clase c en el píxel i.

La Cross Entropy Loss penaliza más fuertemente las predicciones incorrectas que están lejos de las etiquetas reales, y su valor disminuye cuando las predicciones $p_{i,c}$ se aproximan a las etiquetas $y_{i,c}$.

3.1.3 Combinación de ambas

La combinación de Dice Loss y Cross Entropy Loss es común en segmentación médica, como en el desafío KITS23, ya que maneja el desbalance de clases y optimiza el rendimiento general.

Por un lado, **Cross Entropy Loss** optimiza la probabilidad de clasificación correcta para cada píxel (pixel-wise). Por lo que, funciona bien con clases representadas ampliamente, pero penaliza poco las minoritarias. Por otro lado, **Dice Loss** mide la similitud global entre la máscara predicha y la real. Por lo que, es robusta ante el desbalance de clases, considerando adecuadamente clases minoritarias.

Esta combinación aprovecha lo mejor de ambas: la Cross Entropy Loss guía el aprendizaje por píxel, y la Dice Loss mejora la calidad global de las predicciones, especialmente para clases desbalanceadas.

La pérdida combinada se define como:

$$L_{combinada} = 0.5 \cdot L_{Dice} + 0.5 \cdot L_{CE}$$

3.2 Focal Loss

La Focal Loss es una extensión de la Cross Entropy Loss diseñada para manejar el desbalance de clases.

Reduce el impacto de las clases mayoritarias al enfocarse en ejemplos difíciles (como clases minoritarias) y dar menor peso a los ejemplos bien clasificados.

En problemas binarios, se introduce el término $(1-p_t)^\gamma$ para ajustar el peso de los ejemplos según su dificultad:

$$L_{Focal} = -\frac{1}{N}\sum_{t=1}^{N}\alpha_t(1-p_t)^\gamma log(p_t)$$

Donde N es el número de píxeles, y_i es la etiqueta real del píxel i, p_i es la probabilidad predicha para la clase positiva, $\gamma \geq 0$ es un parámetro de enfoque, que aumenta el peso en ejemplos difíciles al crecer su valor, y α es un parámetro opcional para manejar el desbalance entre clases positivas y negativas.

Cuando $\gamma = 0$, la Focal Loss se reduce a la Cross Entropy Loss. Con valores mayores de γ, aumenta la penalización para píxeles mal clasificados y disminuye para los bien clasificados.

En problemas multiclase, la Focal Loss se generaliza sumando las pérdidas de todas las clases.

3.2.1 Focal Loss ponderada por clases

La Focal Loss ponderada es una extensión de la Focal Loss combinada con pesos de clase, diseñada para manejar el desbalanceo de clases (las clases cáncer y quistes son críticas pero están subrepresentadas) de manera más precisa. Esta versión no solo reduce el impacto de las clases mayoritarias, sino que ajusta la importancia de cada clase de acuerdo con su representación en los datos. La hemos implementado con el objetivo principal de analizar su rendimiento dentro de SegResNet, en comparación con las otras funciones de pérdida implementadas (CBLoss, Focal Loss y Combined Loss), en lugar de comparar su desempeño entre varios modelos.

A diferencia de la Focal Loss tradicional, donde el parámetro α es un único valor para todo el conjunto de clases, en la Focal Loss ponderada, α_t es un vector que contiene un peso distinto para cada clase. Esto permite una ponderación más precisa según la frecuencia de las clases en los datos.

3.3 Class-Balanced Loss

La Class-Balanced Loss (CB Loss) adapta las contribuciones de cada clase según su frecuencia efectiva, lo que ayuda a mitigar el impacto de clases desbalanceadas.

La frecuencia efectiva se define como:

$$effective\_num = 1 - \beta^n$$

Donde n es el número de ejemplos de la clase y $\beta \in [0, 1)$ es un hiperparámetro que controla la sensibilidad al desbalance. Valores cercanos a 1 otorgan mayor peso a las clases minoritarias.

Los pesos de cada clase se calculan como:

$$w_c = \frac{1-\beta}{1-\beta^{n_c}}$$

Donde n_c es el número de ejemplos de la clase c. Estos pesos se normalizan para que su suma sea igual al número total de clases.

En nuestro caso, combinamos la CB Loss con la Focal Loss para enfocar el aprendizaje en las regiones difíciles y desbalanceadas, como las pequeñas áreas tumorales, penalizando menos los píxeles de fondo.

4 Métricas de evaluación

Se han utilizado el Índice de Superposición por Unión (IoU) y el Coeficiente Dice para evaluar el desempeño de los modelos en la segmentación de imágenes. Estas métricas son ideales para cuantificar la coincidencia entre las predicciones y las etiquetas reales.

4.1 Intersection over Union (IoU)

El IoU mide la superposición relativa entre la región predicha y la verdadera. Para una clase c, se define como:

$$IoU_c = \frac{|P_c \cap T_c|}{|P_c \cup T_c|}$$

Donde P_c son los píxeles predichos para la clase c, y T_c son los píxeles reales de la clase c.

El IoU varía entre 0 y 1, siendo 1 una coincidencia perfecta, mientras que valores bajos indican segmentaciones deficientes.

4.2 Coeficiente Dice

El Coeficiente Dice otorga más peso a la intersección entre regiones predicha y real. Para una clase c, se define como:

$$Dice_c = \frac{2|P_c \cap T_c|}{|P_c| + |T_c|}$$

Donde $|P_c|$ son los píxeles predichos para la clase c, y $|T_c|$ son los píxeles reales de la clase c.

Al igual que el IoU, el Dice varía entre 0 y 1, pero es más sensible a la coincidencia relativa, lo que lo hace útil en casos con clases desbalanceadas.

5 Modelos 3D

Para nuestro trabajo, se emplearon tres arquitecturas de redes neuronales convolucionales (CNN) ampliamente utilizadas en la segmentación semántica de imágenes volumétricas 3D: U-Net, V-Net y SegResNet. Estas arquitecturas han sido adaptadas y optimizadas para procesar los datos 3D del KiTS23, en el cual se incluyen volúmenes que representan fondo, riñón, timor y quistes.

5.1 U-Net 3D

La arquitectura U-Net tiene una estructura en forma de "U" con dos trayectorias principales:

1. Ruta de contracción (encoder): captura características jerárquicas mediante una serie de capas convolucionales seguidas de operaciones de agrupamiento (max pooling). Esto reduce progresivamente las dimensiones espaciales mientras aumenta la profundidad de las características.

2. Ruta de expansión (decoder): reconstruye las características extraídas a la resolución original mediante operaciones de upsampling y convoluciones transpuestas. En cada etapa, las características de la ruta de contracción se combinan con las de la expansión mediante conexiones salto (skip connections), permitiendo preservar detalles especiales finos.

U-Net destaca por su simplicidad y su capacidad para combinar información global y local gracias a las conexiones de salto. Además, es particularmente efectiva en tareas de segmentación donde los límites entre regiones son críticos.

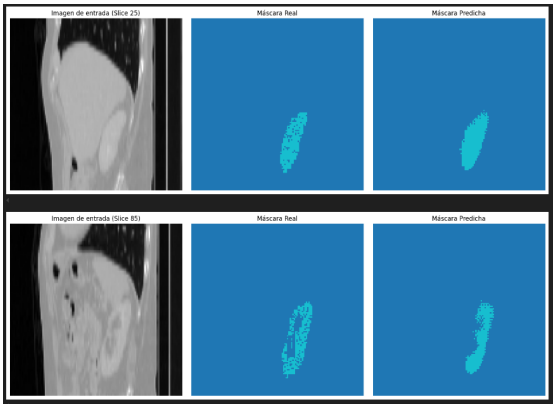

Figure 3: Resultados UNet 3D

5.2 V-Net

La V-Net fue diseñada específicamente para segmentación de imágenes médicas volumétricas 3D. Introduce varias mejoras y modificaciones para manejar datos volumétricos directamente.

La estructura de la V-Net es similar a la U-Net. Sin embargo, en lugar de usar convoluciones simples, incorpora convoluciones residuales. Cada bloque convolucional contiene múltiples capas con conexiones residuales, lo que facilita el entrenamiento de redes profundas al mitigar el problema del gradiente desaparecido. V-Net también emplea funciones de activación paramétricas (como la Parametric ReLU), para mejorar la capacidad de aprendizaje en datos médicos.

V-Net es robusta frente a clases pequeñas y segmentaciones volumétricas complejas. Y, gracias al uso de convoluciones residuales, mejora la convergencia y el desempeño en volúmenes 3D.

5.3 SegResNet

SegResNet es una arquitectura basada en redes neuronales convolucionales profundas que se especializa en datos 3D. Usa un diseño con convoluciones residuales y bloques de codificador y de decodificador, lo que permite extraer características y evitar el desvanecimiento del gradiente.

La inicialización de los pesos se realiza mediante el inicializador de Xavier, similar al de Kaiming usado en el modelo de U-Net 3D. Se ha hecho finetuning: las primeras capas del encoder se congelan durante el entrenamiento, por lo que solo las capas del decoder se actualizan. Así podemos aprovechar características previamente aprendidas por el modelo en otras tareas relacionadas. Esto acelera el entrenamiento y mejora la generalización, reduciendo el riesgo de sobreajuste.

SegResNet proporciona un balance entre complejidad computacional y desempeño, lo que la hace especialmente buena para escenarios donde la capacidad de cómputo es limitada.

6 Modelos 2D

A parte de experimentar con las imágenes en 3D, queremos hacer una aproximación a lo visto durante el curso en clase. Es decir, de las imágenes 3D que tenemos preprocesadas, hemos hecho cortes 2D equidistantes para poder aplicar algoritmos y extraer características de las imágenes. Viendo la gran cantidad de imágenes que tenemos hemos elegido hacer 128 cortes. En total hemos trabajado con 124928 imágenes 2D, en las que esta presente un gran desbalanceo de clases. Muy pocas veces encontramos tumor o quiste con respecto al fondo.

En general, las CNN 3D superan a las CNN 2D en tareas de aprendizaje profundo médico gracias a la información espacial que aportan. Sin embargo, su aplicación es limitada debido a la mayor necesidad de datos, recursos computacionales, tiempo de entrenamiento y la escasez de modelos 3D preentrenados disponibles públicamente.

Al trabajar con imágenes 2D, que no pesan tanto, el batch size se ha podido aumentar notablemente de 4 a 64.

6.1 U-Net con un codificador preentrenado ResNet50

Hemos implementado un modelo de segmentación de imágenes utilizando U-Net con un codificador preentrenado (resnet50).

Para la configuración del modelo, hemos utilizado el codificador de resnet50, preentrenado en ImageNet. Para ello nos ha hecho falta la librería de *segmentation_models_pytorch*. Como función de pérdida hemos utilizado focal loss, definida anteriormente y como optimizador: usamos Adam con una tasa de aprendizaje fija de $1e-4$.

Entrenamos por 15 épocas. En entrenamiento, en el forward pass calculamos la salida del modelo y la pérdida mientras que en el backward pass y optimización, se calculan los gradiente y los pesos se actualizan. En cuanto a las métricas, utilizamos Dice Score e IoU. Además, enseñamos en los resultados el Dice de cada clase donde nos revela el gran desbalanceo ya que siempre predice mucho mejor el fondo.

Clase	**IoU**	**Dice**
Clase 0 fondo	0.9951	0.9975
Clase 1 kidney	0.4079	0.2324
Clase 2 tumor	0.1171	0.0592
Clase 3 quiste	0.0048	0.0010

Table 1: Resultados en test 3D U-Net

6.2 U-Net con oversampling

Para poder subsanar esto hemos realizado oversampling y subsampling. Al haber explicado ya como hemos hecho esta técnica, nos centramos en los resultados. Al estar creando ejemplos de las clases minoritarias, deberíamos de ver como mejoran los resultados en test:

Clase	**IoU**	**Dice**
Clase 0 fondo	0.9951	0.9975
Clase 1 kidney	0.4473	0.2432
Clase 2 tumor	0.1042	0.0452
Clase 3 quiste	0.0001	0.000

Table 2: Resultados en test 3D U-Net con oversampling

En entrenamiento si mejora mucho ya que es donde estamos haciendo el sobremuestreo y todas las clases las predice bastante bien. Sin embargo, estamos sobreaprendiendo ya que en test y en validación no mejora.

6.3 U-Net con subsampling

Esta técnica no es del todo recomendable, pero ya que no nos ha ayudado mucho el sobremuestreo, queríamos probar. No es recomendable ya que estamos eliminando ejemplos y, en general, queremos tener cuantos más ejemplos mejor. Vamos a eliminar las imágenes que contengan más del 99% con todo fondo, ya que es la clase que menos nos importa y es la clase más dominante sin duda. Por tanto, nos podría llegar a servir para reducir el sesgo y quedándonos con los datos "interesantes".

Clase	**IoU**	**Dice**
Clase 0 fondo	0.9924	0.9961
Clase 1 kidney	0.3179	0.2175
Clase 2 tumor	0.0590	0.0351
Clase 3 quiste	0.0007	0.0004

Table 3: Resultados test 3D con subsampling

En general, vemos como empeora y podemos concluir que esta técnica no es adecuada.

7 Experimentos 3D

7.1 U-Net

Clase	**IoU**	**Dice**
Clase 0 fondo	0.9936	0.9968
Clase 1 kidney	0.4639	0.6274
Clase 2 tumor	0.0245	0.0454
Clase 3 quiste	0.0004	0.0001

Table 4: Resultados obtenidos de usar U-Net con sobremuestreo y Combined Loss en test

7.2 V-Net

Clase	**IoU**	**Dice**
Clase 0 fondo	0.9931	0.9966
Clase 1 kidney	0.3743	0.5395
Clase 2 tumor	0.0442	0.0713
Clase 3 quiste	0	0.0001

Table 5: Resultados obtenidos de V-Net con sobremuestreo y Combined Loss en test

Clase	**IoU**	**Dice**
Clase 0 fondo	0.9931	0.9966
Clase 1 kidney	0.3743	0.5395
Clase 2 tumor	0.0442	0.0713
Clase 3 quiste	0	0.0001

Table 6: Resultados obtenidos de V-net con sobremuestreo y Focal Loss en test

Clase	**IoU**	**Dice**
Clase 0 fondo	0.9938	0.9969
Clase 1 kidney	0.4118	0.5773
Clase 2 tumor	0.0934	0.1546
Clase 3 quiste	0	0

Table 7: Resultados obtenidos de V-Net con sobremuestreo y CB Loss en test

7.3 SegResNet

Clase	**IoU**	**Dice**
Clase 0 fondo	0.9927	0.9963
Clase 1 kidney	0.4130	0.5728
Clase 2 tumor	0.0107	0.0175
Clase 3 quiste	0	0

Table 8: Resultados obtenidos de SegResNet con sobremuestreo y Combined Loss

Clase	IoU	Dice
Clase 0 fondo	0.9933	0.9967
Clase 1 kidney	0.4220	0.5778
Clase 2 tumor	0.0928	0.1462
Clase 3 quiste	0	0

Table 9: Resultados de SegResNet con sobremuestreo y Focal Loss en test

Clase	IoU	Dice
Clase 0 fondo	0.9936	0.9968
Clase 1 kidney	0.4396	0.5938
Clase 2 tumor	0.0861	0.1339
Clase 3 quiste	0	0

Table 10: Resultados de SegResNet con sobremuestreo y CB Loss en test

Clase	IoU	Dice
Clase 0 fondo	0.9883	0.9941
Clase 1 kidney	0.3480	0.5142
Clase 2 tumor	0.1433	0.2344
Clase 3 quiste	0.0033	0.0065

Table 11: Resultados de SegResNet con sobremuestreo y Focal Loss Weighted en test

7.4 Comparación

Podemos observar en los resultados obtenidos que SegResNet con CB Loss logra un rendimiento consistente en validación y test, superando a otros modelos como U-Net y V-Net. Esto podría deberse a que SegResNet usa fine-tuning para aprovechar las características aprendidas en anteriores problemas, mientras que U-Net y V-Net son implementaciones estándar que no tienen tanta capacidad para extraer características.

8 Conclusiones

En este proyecto se evaluaron arquitecturas 2D y 3D de redes neuronales convolucionales (CNN) para la segmentación de imágenes médicas. Los modelos 3D, como SegResNet, U-Net 3D y V-Net, mostraron un mejor desempeño al aprovechar la información espacial completa de los volúmenes, destacando SegResNet con CB Loss por sus resultados consistentes tras el fine-tuning. No obstante, el desbalance de clases sigue siendo un desafío, afectando métricas como Dice e IoU, especialmente en clases minoritarias como tumores y quistes.

Los modelos 2D, aunque más eficientes computacionalmente, perdieron información crucial al trabajar con cortes bidimensionales, resultando en un rendimiento inferior, incluso con técnicas como sobremuestreo y focal loss.

En conclusión, las arquitecturas 3D son más efectivas para segmentación volumétrica, pero se requiere continuar trabajando en estrategias para abordar el desbalance de clases y mejorar la generalización.

References

[1] Joseph Chen and Benson Jin. A 2d u-net for automated kidney and renal tumor segmentation. 2019.

[2] Nicholas Heller, Fabian Isensee, Dasha Trofimova, Resha Tejpaul, Zhongchen Zhao, Huai Chen, Lisheng Wang, Alex Golts, Daniel Khapun, Daniel Shats, et al. The kits21 challenge: Automatic segmentation of kidneys, renal tumors, and renal cysts in corticomedullary-phase ct. *arXiv preprint arXiv:2307.01984*, 2023.

[3] Xin Wang and et al. Su. 2.75 d: Boosting learning by representing 3d medical imaging to 2d features for small data. *Biomedical Signal Processing and Control*, 84:104858, 2023.

[4] Wikipedia. Escala hounsfield. https://es.wikipedia.org/wiki/Escala_Hounsfield, 2024.

[5] Michael Yeung, Evis Sala, Carola-Bibiane Schönlieb, and Leonardo Rundo. Unified focal loss: Generalising dice and cross entropy-based losses to handle class imbalanced medical image segmentation. *Computerized Medical Imaging and Graphics*, 95:102026, 2022.