

# Evaluación de modelos convolucionales basados en U-Net para la segmentación de imágenes usando los conjuntos de datos Montgomery y Shenzhen

John Patricio Serrano Carrasco

Agosto de 2024

**Resumen-** Se implementaron y evaluaron dos arquitecturas de Redes Neuronales Convolucionales (CNN), U-Net y U-Net++, para la segmentación de imágenes médicas utilizando los conjuntos de datos combinados de Montgomery y Shenzhen. Se emplearon 677 imágenes, de las cuales el 80 % se destinó al entrenamiento y el 20 % a la validación. Los modelos fueron entrenados durante 30 épocas utilizando el optimizador Adam y luego el optimizador SGD. El rendimiento se evaluó mediante el Coeficiente de Dice y el valor de loss durante el entrenamiento. Los resultados mostraron diferencias en la capacidad de segmentación entre ambas arquitecturas, destacando la efectividad de U-Net++ en la mejora de los resultados para los casos más complejos.

**Palabras claves**—Redes Neuronales Convolucionales, Segmentación de imágenes, Coeficiente de Dice, Función de pérdida.

## I. INTRODUCCIÓN

La segmentación de imágenes médicas es un desafío fundamental en el ámbito de la inteligencia artificial aplicada a la medicina, con un impacto directo en la precisión del diagnóstico y el tratamiento de diversas enfermedades. Este proceso implica la identificación automática de regiones de interés dentro de imágenes médicas, como radiografías, tomografías o resonancias magnéticas, lo que permite a los profesionales de la salud tomar decisiones más informadas y basadas en datos objetivos. El avance de las Redes Neuronales Convolucionales (CNN) ha abierto nuevas posibilidades para abordar el problema de la segmentación de imágenes médicas con un alto grado de precisión. Modelos como U-Net y U-Net++ han demostrado ser particularmente eficaces en la segmentación de imágenes médicas debido a su capacidad para capturar características a múltiples escalas y para manejar la complejidad de las imágenes médicas. U-Net es una arquitectura de encoder-decoder que ha sido ampliamente utilizada para tareas de segmentación debido a su capacidad para capturar tanto detalles finos como características globales en la imagen. U-Net++ es una extensión de U-Net que introduce conexiones adicionales entre las capas, lo que permite una mejor propagación de la información y, en teoría, una mayor precisión en la segmentación. Esta experiencia se centra en la implementación y evaluación de estas dos arquitecturas de CNN, utilizando los conjuntos de datos combinados de Montgomery y Shenzhen, los cuales contienen radiografías de tórax y máscaras de pulmones. El enfoque del estudio es comparar la eficacia de U-Net y U-Net++ en la segmentación de imágenes de radiografías de tórax y la capacidad de predecir correctamente máscaras correspondientes a los pulmones de la

radiografía, utilizando métricas como el Coeficiente de Dice y la pérdida (loss) durante el entrenamiento para evaluar el rendimiento de los modelos. Para ambos modelos la metodología fue consistente, utilizando el 80 % de los datos para entrenamiento y el 20 % de los datos para prueba. Se evaluaron los modelos utilizando los optimizadores Adam y SGD para poder comprobar cual de los dos es la mejor opción a la hora de realizar la segmentación de imágenes y se elaboraron gráficos de los valores de Coeficiente de Dice y loss respecto a las épocas para la comparación entre los modelos.

## II. DESARROLLO

Para el desarrollo de la experiencia, se configuró PyTorch para utilizar la GPU disponible, correspondiente a una NVidia RTX 3060 con Cuda 11.8, lo que permitió acelerar significativamente el entrenamiento y la evaluación de los modelos y es fundamental para manejar la carga computacional que implica el entrenamiento de modelos de CNN en un conjunto de datos grande.

El primer paso en la implementación fue la preparación de los datos. Como se mencionó en la introducción, se utilizaron los conjuntos de datos combinados de Montgomery y Shenzhen, que consisten en 677 radiografías de tórax con sus correspondientes máscaras de segmentación. El 80 % de las imágenes se destinó al conjunto de entrenamiento y el 20 % al conjunto de prueba, siguiendo la práctica estándar en la evaluación de modelos de aprendizaje profundo, con un total de 541 imágenes de entrenamiento y 136 imágenes de prueba respectivamente, cada una con su respectiva radiografía de torax y máscara correspondiente. Las imágenes fueron redimensionadas a 128x128 píxeles para reducir la carga computacional y acelerar el proceso de entrenamiento. Este tamaño fue elegido como un compromiso entre la pérdida de detalle en la imagen y la eficiencia computacional.

Para la carga de imágenes, se utilizó la biblioteca PIL, la cual demostró ser más confiable que otras alternativas como torchvision.io.read\_image, que presentaba inconsistencias en la calidad de las imágenes cargadas, mostrando algunas radiografías extremadamente oscuras sin razón aparente. Las imágenes se normalizaron dividiendo sus valores de píxel entre 255 para escalarlas a un rango [0, 1], lo que es esencial para que las redes neuronales puedan procesarlas adecuadamente.

Se implementaron las arquitecturas U-Net y U-Net++ utilizando **Segmentation Models con PyTorch**. Ambos modelos fueron implementados siguiendo la arquitectura establecida en el paper de cada modelo, utilizando un encoder ResNet34 y pesos pre-entrenados en ImageNet. El diagrama de bloques de U-Net se puede apreciar en la Figura 1 y el de U-Net++ en la Figura 2.

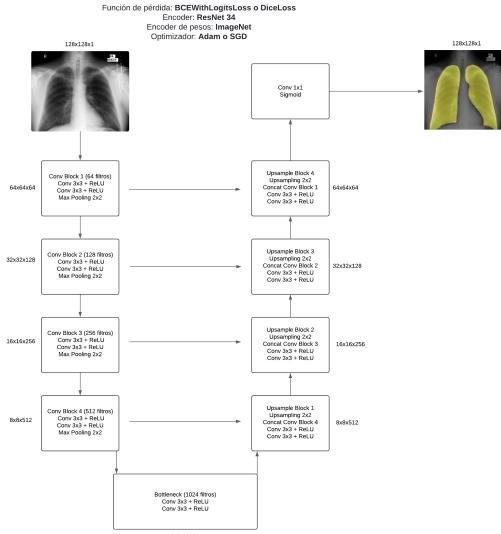


Figura 1: Diagrama de bloques de U-Net

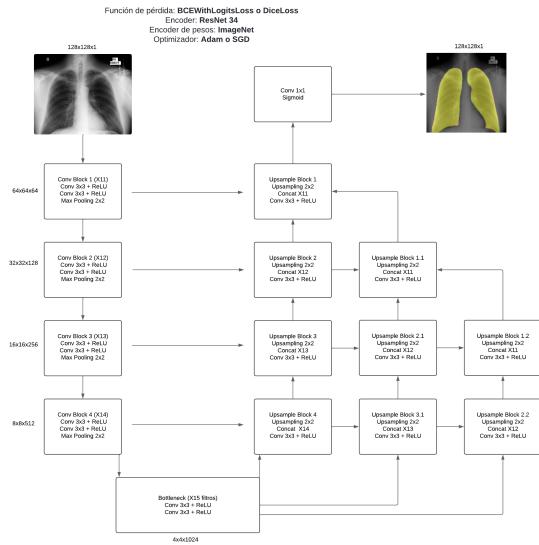


Figura 2: Diagrama de bloques de U-Net++

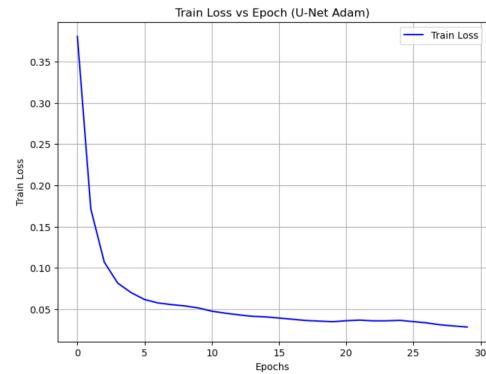
El entrenamiento se llevó a cabo utilizando dos optimizadores diferentes: Adam y SGD. Adam se utilizó en las primeras pruebas debido a su capacidad para adaptarse dinámicamente al aprendizaje de cada parámetro, lo que generalmente resulta en una convergencia más rápida y estable. Sin embargo, SGD también se implementó en etapas posteriores para evaluar si un optimizador más sencillo podría mejorar la generalización del modelo. Ambos modelos con ambos optimizadores fueron entrenados durante 30 épocas, con un tamaño de lote de 32 imágenes. Este tamaño de lote fue elegido para aprovechar la capacidad de procesamiento de la GPU, asegurando un equilibrio entre la velocidad de entrenamiento y la estabilidad del gradiente. Inicialmente, se utilizó la función de pérdida BCEWithLogitsLoss, una elección común en problemas de segmentación binaria, ya que combina la función de pérdida de entropía cruzada binaria con una capa de activación Sigmoid. Esta combinación permite una mejor convergencia durante

las primeras etapas del entrenamiento, facilitando la tarea de aprendizaje del modelo. Se eligieron 30 épocas para asegurar que el modelo pudiera converger adecuadamente, sin caer en un sobreentrenamiento. Este número fue determinado principalmente para poder evaluar los modelos con los distintos optimizadores rápidamente.

Durante el entrenamiento y evaluación, se registraron tanto el valor de loss como el Coeficiente de Dice en cada época junto con gráficos de épocas VS loss y épocas VS Coeficiente de Dice. Estas métricas permitieron comparar directamente el rendimiento de U-Net y U-Net++, especialmente en la segmentación de casos complejos. Adicionalmente, se implementó la visualización de las imágenes donde la máscara se superpone a la radiografía de tórax, donde el color rojo representa la predicción del modelo, el color verde es la máscara superpuesta y el color amarillo es lo correctamente segmentado. Al final de la experiencia se realiza el cambio de los hyperparámetros, donde se aumenta el número de épocas a 100 (lo que aumenta el tiempo de ejecución, especialmente con U-Net++) y la función de pérdida DiceLoss, para determinar cuánto influye la función de pérdida escogida en los resultados. Se desarrollaron funciones para evitar la repetición de código, además de Pandas para la manipulación de datos, NumPy para operaciones matemáticas, y Matplotlib para visualizar los resultados mediante gráficos e imágenes y realizar análisis comparativos detallados.

### III. RESULTADOS EXPERIMENTALES

Al utilizar la arquitectura U-Net con el optimizador Adam y la función de pérdida BCEWithLogitsLoss, los resultados obtenidos durante el entrenamiento y la evaluación muestran un desempeño sobresaliente en la tarea de segmentación de pulmones en imágenes. En primer lugar, el valor de loss durante el entrenamiento disminuye consistentemente a lo largo de las 30 épocas, comenzando en 0.3803 y estabilizándose en valores cercanos a 0.0285, lo que indica una rápida convergencia. Este comportamiento sugiere que el optimizador Adam ha sido capaz de ajustar los pesos del modelo de manera eficiente, minimizando la función de pérdida y mejorando el rendimiento de la segmentación.



mejora significativamente durante el proceso de entrenamiento. En las primeras épocas, el coeficiente de Dice aumenta rápidamente, alcanzando valores superiores a 0.96 después de la quinta época y manteniéndose estable en este rango hasta el final del entrenamiento. Este alto valor de Dice refleja que U-Net, con la ayuda de Adam, ha logrado capturar con precisión las características relevantes de las imágenes, resultando en segmentaciones de alta calidad.

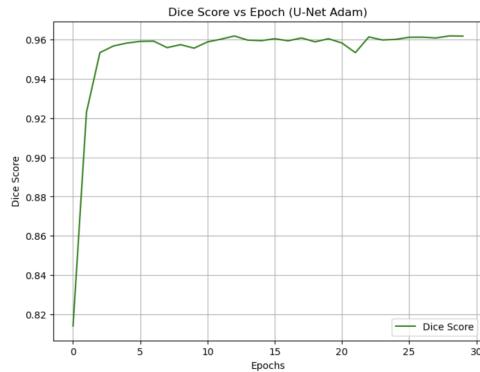


Figura 4: Gráfico de Dice VS Epoch (U-Net Adam)

El valor de loss en la evaluación sigue una tendencia descendente similar a la del entrenamiento, estabilizándose alrededor de 0.0572. Este comportamiento es indicativo de una buena generalización del modelo, ya que logra mantener un bajo nivel de pérdida en datos que no ha visto durante el entrenamiento, evitando problemas de sobreajuste.

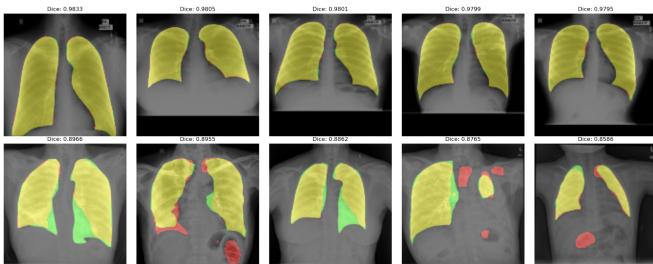


Figura 5: 5 casos fáciles y difíciles (U-Net Adam)

Respecto a las predicciones del modelo, como se puede apreciar en la Figura 5, estas son bastante certeras, donde se tiene que la peor predicción tiene un coeficiente de Dice de 0,8586, mientras que la mejor predicción tiene un Coeficiente de Dice de 0,9833. De la Figura 5 se desprende que los mejores casos o casos donde al modelo no le costó predecir (alto valor de Dice) suelen ser imágenes con características anatómicas bien definidas y una calidad de imagen clara. Estas imágenes tienen un buen contraste entre las estructuras pulmonares y los tejidos circundantes, muestran una estructura torácica simétrica y sin anomalías visibles, por lo que suelen ser segmentadas con alta precisión. El modelo parece adaptarse mejor cuando las variaciones anatómicas son mínimas, lo que reduce la complejidad de la tarea de segmentación. Respecto a los casos donde al modelo le costó predecir (bajo valor de Dice), imágenes donde los pulmones presentan formas irregulares, como en casos de deformidades o posiciones

atípicas durante la toma de la radiografía, suelen ser más difíciles de segmentar. Esta variabilidad no es siempre bien manejada por el modelo, resultando en sobresegmentación o subsegmentación de las regiones pulmonares, como se puede ver en la imagen que tiene el Coeficiente de Dice más bajo, donde el modelo incluso hizo una predicción en la parte inferior del tórax, demostrando que hubo dificultad de predecir correctamente la localización de los pulmones.

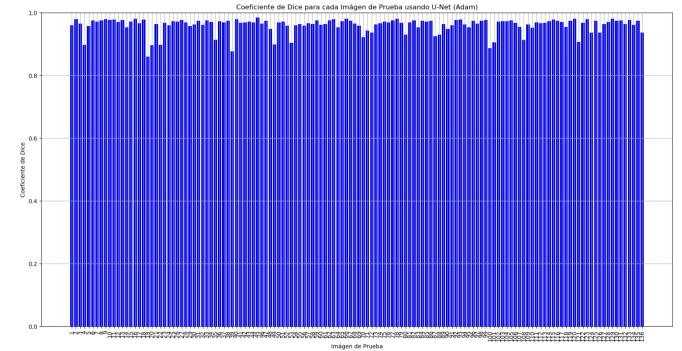


Figura 6: Gráfico de barras de Coeficiente de Dice de imágenes de prueba (U-Net Adam)

El gráfico de barras de los Coeficientes de Dice para las imágenes de prueba confirma que, aunque la mayoría de los valores se encuentran en un rango elevado, existen algunas imágenes que presentan un rendimiento significativamente inferior. Estas anomalías pueden deberse a variaciones en la calidad de las imágenes o incluso a limitaciones inherentes del modelo U-Net en la identificación de ciertas características, como se mencionó anteriormente.

**El cambio del optimizador de Adam a SGD (sin modificar algo más) en el modelo U-Net** tuvo un impacto significativo en el rendimiento general del modelo. Mientras que Adam mostró una rápida convergencia en el valor de loss de entrenamiento y una mejora constante en el coeficiente de Dice a lo largo de las épocas, el optimizador SGD mostró un comportamiento muy diferente. El valor de loss de entrenamiento disminuyó de manera más gradual y, aunque consistentemente bajó a lo largo de las épocas, el coeficiente de Dice se mantuvo relativamente bajo en comparación con Adam, con algunas alzas.

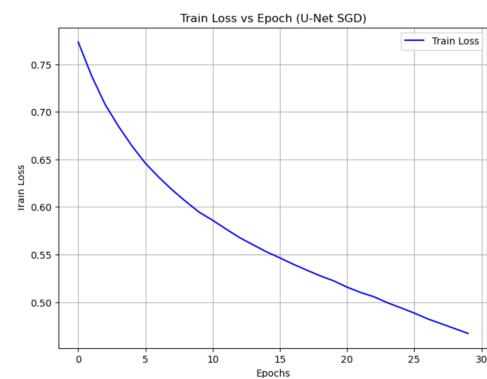


Figura 7: Gráfico de Loss VS Epoch (U-Net SGD)

Desde el principio, el coeficiente de Dice con SGD fue notablemente más bajo que con Adam, y en lugar de mejorar progresivamente, en muchas ocasiones se observó una tendencia a disminuir durante las primeras 20 épocas. Sin embargo, después de esta fase inicial, hubo una recuperación en las últimas 10 épocas, lo que sugiere que el modelo podría estar comenzando a ajustarse mejor, aunque aún muy por debajo de los resultados obtenidos con Adam.

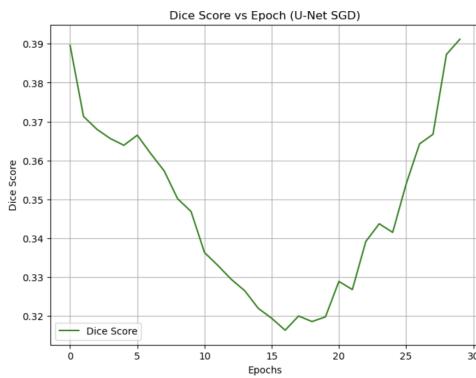


Figura 8: Gráfico de Dice VS Epoch (U-Net SGD)

Más diferencias se pueden seguir observando a la hora de ver los resultados de las predicciones del modelo, donde el valor máximo de Coeficiente de Dice con las imágenes de prueba fue de 0,6339 y el menor fue de 0,1377. Además se puede observar en las imágenes como el modelo no es capaz de formar la máscara correctamente (color rojo), realizando una especie de predicción esparsa por varios píxeles de la imagen, lo que nuevamente indica que el modelo tiene bastantes dificultades cuando se usa el optimizador SGD.

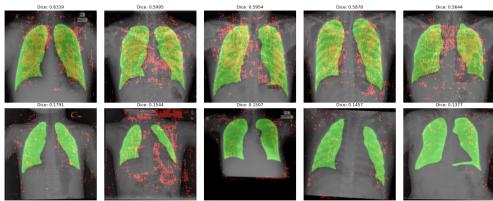


Figura 9: 5 casos fáciles y difíciles (U-Net SGD)

Esta diferencia sugiere que Adam, con su capacidad para adaptarse a diferentes velocidades de aprendizaje para cada parámetro, es mucho más eficiente para este tipo de tarea de segmentación de imágenes, permitiendo que el modelo converja más rápidamente y alcance mejores resultados de segmentación. Por otro lado, SGD, aunque eventualmente muestra una mejora, es significativamente más lento en aprender y ajustarse, lo que se refleja en el bajo coeficiente de Dice y en la lenta reducción de el valor de loss.

Dado que Adam demuestra ser la mejor opción a la hora de elegir el optimizador a utilizar para el modelo U-Net, se realizó la modificación de hyperparámetros utilizando este optimizador. La primera modificación fue en el número de épocas, aumentando el número de 30 a 100. Los resultados fueron similares a los obtenidos, pero el tiempo de ejecución aumentó considerablemente. La segunda prueba fue modificar

la función de pérdida a DiceLoss, lo que provocó que hubiera una leve variación respecto al Coeficiente de Dice en algunas épocas, pero manteniendo buenos resultados en general.

Similar a como fue el caso de U-Net con el optimizador Adam, los resultados del entrenamiento y evaluación del modelo **U-Net++ utilizando el optimizador Adam** demuestran un rendimiento altamente eficaz. A lo largo de las 30 épocas de entrenamiento, se observa una disminución continua y significativa en el valor de loss tanto en el conjunto de entrenamiento como en el de validación.

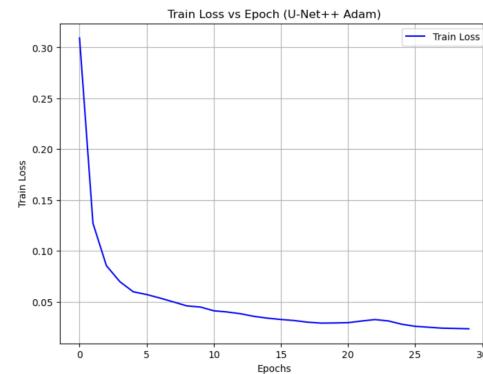


Figura 10: Gráfico de Loss VS Epoch (U-Net++ Adam)

U-Net++ tiende a ofrecer un rendimiento ligeramente superior en términos de precisión, como se refleja en los valores consistentemente altos del Coeficiente de Dice, el cual muestra una mejora rápida en las primeras épocas, estabilizándose alrededor de valores cercanos a 0.96, lo que indica una alta precisión en la segmentación. Esto sugiere que la arquitectura U-Net++ puede manejar mejor la complejidad en la segmentación de estructuras pulmonares en imágenes de radiografías de tórax, lo que podría deberse a la mayor capacidad de representación y a las mejoras en la conectividad dentro de la red proporcionadas por su arquitectura más compleja.

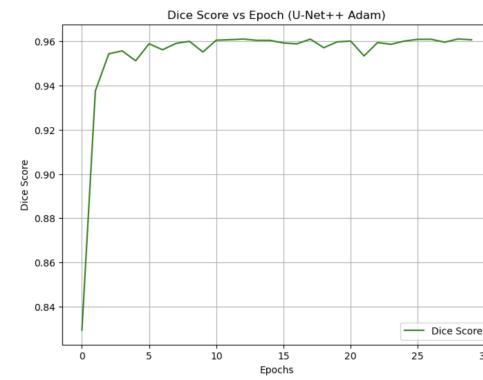


Figura 11: Gráfico de Dice VS Epoch (U-Net++ Adam)

Al comparar los mejores y peores casos, se observa que U-Net++ mantiene una capacidad de segmentación superior en los casos más difíciles en comparación con U-Net. Esto se evidencia en las imágenes con los valores de Dice más bajos, donde U-Net++ sigue logrando una segmentación más precisa en comparación con U-Net, particularmente en áreas donde la estructura pulmonar es más compleja o tiene irregularidades.

Incluso, se puede apreciar que a diferencia de U-Net, con U-Net++, radiografías que tienen particularidades como estar movidas, o tener deformidades, no tienen inconvenientes con U-Net++, obteniendo un alto valor de Coeficiente de Dice, como la imagen que obtuvo un Coeficiente de Dice de 0,9809, contrario a lo que sucedió con U-Net, pero aún así hay algunas imágenes que siguen teniendo dificultades independiente del modelo.

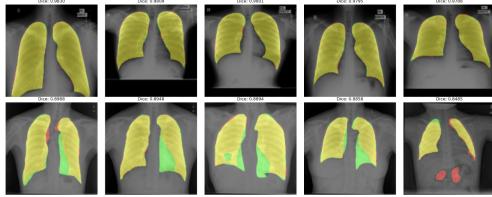


Figura 12: 5 casos fáciles y difíciles (U-Net++ Adam)

El gráfico de barras que muestra los coeficientes de Dice para todas las imágenes de prueba refuerza lo mencionado anteriormente, ya que la variabilidad en el rendimiento es menor con U-Net++. Esto sugiere que U-Net++ es más robusto en la segmentación de diferentes tipos de imágenes, lo que podría ser atribuido a su arquitectura más compleja y el uso de múltiples decoders, que permiten capturar mejor las características multiescalares de las imágenes. A pesar de eso, es importante mencionar que U-Net++ toma mucho más tiempo de entrenar y evaluar, debido a su arquitectura más compleja.

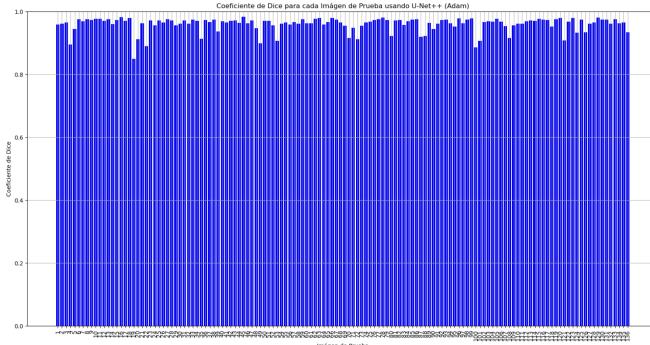


Figura 13: Gráfico de barras de Coeficiente de Dice de imágenes de prueba (U-Net++ Adam)

Respecto al **optimizador SGD con U-Net++**, sucede algo similar que con U-Net, solo que ahora se puede ver más claro que U-Net++ proporciona mejores resultados. En particular, los valores de loss de entrenamiento y evaluación disminuyen de manera constante, lo que indica una mejora en la capacidad del modelo para ajustar los datos.

El Coeficiente de Dice empieza desde un valor bajo (alrededor de 0.33 en la primera época) y llega a un valor de 0.81 al final del entrenamiento. Esto sugiere que el modelo, aunque mejora, sigue teniendo dificultades para segmentar las imágenes de manera efectiva, pero logra obtener mejores resultados que U-Net con el optimizador SGD, ya que U-Net con SGD alcanza un Coeficiente de Dice final que apenas supera los 0.39, mientras que U-Net++ logra más del doble

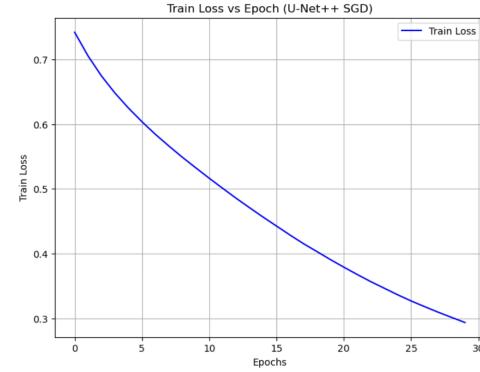


Figura 14: Gráfico de Loss VS Epoch (U-Net SGD)

de este valor, llegando a 0.8189. De la Figura 15 se puede observar que los valores del Coeficiente de Dice crecen desde la primera época hasta la última, sin variaciones, contrario a como sucedió con U-Net.

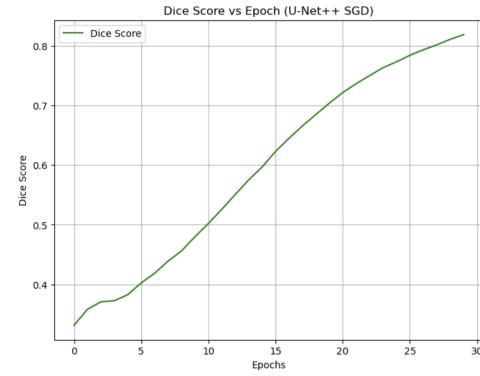


Figura 15: Gráfico de Dice VS Epoch (U-Net++ SGD)

Al analizar las predicciones de U-Net++ utilizando SGD y compararlas con las de U-Net con el mismo optimizador, se observa que U-Net++ tiene un desempeño superior en la segmentación de las imágenes de prueba. Como se puede observar en la Figura 16, las imágenes que logran los mejores Coeficientes de Dice con U-Net++ muestran una segmentación más precisa en comparación con las de U-Net, con menos ruido en las áreas circundantes y una mejor definición de los bordes de los pulmones. Incluso en los casos más difíciles para U-Net++, donde el Coeficiente de Dice es relativamente bajo, el modelo aún logra identificar correctamente las áreas principales de los pulmones, aunque con cierto ruido y errores en las áreas periféricas. Este comportamiento se compara favorablemente con U-Net, que mostró dificultades significativas para manejar estos mismos casos, resultando en segmentaciones mucho más imprecisas y valores de Dice aún más bajos.

A la hora de modificar los hyperparámetros de U-Net++ con el optimizador Adam, sucede algo muy similar con U-Net. Aumentar el número de épocas aumenta significativamente el tiempo de ejecución, pero proporciona buenos resultados, muy similares a los obtenidos con 30 épocas. Cambiar la función de pérdida a DiceLoss da valores muy similares a los obtenidos con la función de pérdida BCEWithLogitsLoss,

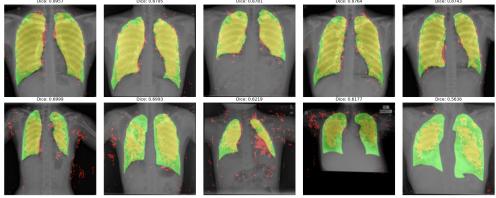


Figura 16: 5 casos fáciles y difíciles (U-Net++ SGD)

aunque DiceLoss logra obtener un valor de loss un poco más bajo.

De los resultados se puede desprender que el mejor optimizador a utilizar tanto para U-Net como U-Net++ es Adam, lo cual tiene sentido, porque combina las ventajas de dos métodos de optimización: AdaGrad, que trabaja bien con características esparsas, y RMSProp, que es ideal para tareas con gradientes ruidosos. Adam ajusta las tasas de aprendizaje para cada parámetro, lo que permite un aprendizaje más rápido y eficiente. En los resultados observados, Adam permitió una convergencia más rápida y mejores valores de Dice, lo que sugiere que su capacidad para adaptar la tasa de aprendizaje a lo largo del entrenamiento ayuda a manejar mejor las variaciones en los datos, conduciendo a segmentaciones más precisas.

Los resultados también demuestran que U-Net++ supera consistentemente a U-Net, tanto en términos del Coeficiente de Dice como en la calidad visual de las segmentaciones. U-Net++ maneja mejor los casos complejos, lo que se refleja en valores de Dice más altos en general, especialmente en los casos difíciles. Esto sugiere que la arquitectura mejorada de U-Net++ es más robusta y eficaz para la segmentación de imágenes, capturando mejor los detalles finos y reduciendo los errores en las segmentaciones. A partir de los resultados, se recomienda utilizar Adam como optimizador para tareas de segmentación de imágenes médicas debido a su mejor rendimiento en términos de rapidez de convergencia y precisión. Además, U-Net++ debería ser la arquitectura preferida sobre U-Net, especialmente en aplicaciones donde la precisión de segmentación es importante, dado que su diseño más complejo proporciona una ventaja significativa en la capacidad de generalización.

#### IV. CONCLUSIONES

Los resultados obtenidos en esta experiencia reflejan la efectividad y las diferencias clave entre las arquitecturas U-Net y U-Net++ en la tarea de segmentación de imágenes médicas. Ambas arquitecturas han demostrado ser herramientas poderosas en la extracción precisa de características en imágenes complejas. A lo largo de los experimentos, se observó que el optimizador Adam ofreció un rendimiento significativamente superior al optimizador SGD, mostrando una convergencia más rápida y resultados más consistentes en términos del Coeficiente de Dice. Este comportamiento puede atribuirse a la capacidad de Adam para ajustar dinámicamente las tasas de aprendizaje, lo que le permite manejar eficientemente los gradientes en problemas de alta dimensionalidad y evitar estancarse en óptimos locales.

La comparación directa entre U-Net y U-Net++ mostró que la arquitectura U-Net++ superó a U-Net en la mayoría de los casos, especialmente en situaciones donde la segmentación era más compleja. U-Net++, con su red densa de conexiones, es capaz de capturar más eficazmente las sutilezas y detalles finos de las imágenes médicas, lo que se traduce en una mayor precisión en la segmentación. Sin embargo, esta mayor capacidad de segmentación también trae consigo un incremento en los requisitos computacionales. U-Net++ es una arquitectura más pesada, tanto en términos de memoria como de tiempo de procesamiento, lo que podría limitar su aplicabilidad en entornos donde los recursos de hardware son limitados o donde se requiere una rápida inferencia.

Un aspecto destacable del análisis realizado es la observación de cómo U-Net++ maneja mejor los casos más difíciles, aquellos donde U-Net muestra un mayor margen de error. Las imágenes que presentaban anatomías o patologías complejas fueron segmentadas con una precisión notablemente mayor por U-Net++, lo que sugiere que esta arquitectura es más robusta frente a la variabilidad en los datos de entrada. Esto es especialmente relevante cuando se considera la diversidad de condiciones que pueden presentarse en un entorno clínico real, donde la capacidad de generalización de un modelo puede marcar la diferencia en su utilidad práctica. A pesar de las ventajas de U-Net++, es importante considerar el equilibrio entre precisión y eficiencia. Existen situaciones donde U-Net podría ser una opción más viable, siempre y cuando se acepte un ligero compromiso en la precisión a cambio de un rendimiento más rápido.

De cara al futuro y considerando los resultados obtenidos, se pueden sugerir varias direcciones para trabajos futuros. Primero, se podría explorar la combinación de U-Net++ con técnicas de aumento de datos y regularización para mejorar aún más la generalización del modelo, especialmente en datos de prueba. Además, dada la efectividad de Adam, se podría experimentar con variantes de Adam o combinarlo con otros optimizadores adaptativos para ver si se pueden obtener mejoras adicionales en la estabilidad y la velocidad de convergencia. También sería valioso realizar un análisis más detallado para identificar casos específicos donde el modelo falla y ajustar la arquitectura y los hiperparámetros, con tal de mejorar los resultados en casos donde los modelos tuvieron complicaciones a la hora de realizar la predicción. La implementación exitosa de estas arquitecturas en entornos clínicos tiene el potencial de mejorar la precisión diagnóstica y también puede acelerar el proceso de toma de decisiones médicas. Modelos como U-Net y U-Net++ representan un avance significativo en la automatización de procesos médicos, y su correcta aplicación podría transformar la práctica médica, haciéndola más eficiente, precisa y accesible.