Detección de tumores cerebrales mediante U-Net y Yolo

Alejandro Puche García

alejandro.puche@alumnos.upm.es

Abstract

Este estudio evalúa y compara el desempeño de modelos de aprendizaje profundo para la detección de tumores cerebrales en imágenes de resonancia magnética (IRM). En particular, se analizaron cuatro variantes de la arquitectura U-Net y un modelo YOLO v11n, entrenados y probados en un conjunto de datos compuesto por 2,146 imágenes, divididas en conjuntos de entrenamiento (1,502 imágenes), validación (215 imágenes) y test (429 imágenes). Se generaron máscaras binarias a partir de las bounding boxes para el uso de los modelos U-net. El desempeño de los modelos se evaluó mediante la métrica de Intersección sobre Unión (IoU) binaria, con resultados que oscilaron entre 0.48 y 0.54 para los modelos U-Net. También se analizaron las capacidades de detección de YOLO y se compararon con los resultados de segmentación. Para hacer esta comparación se usó la métrica mAP sobre un umbral de 0.5 en la métrica IoU. El modelo Yolo arrojó mejores resultados sobre las métricas, sin embargo, visualizando las imágenes se puede ver como los modelos U-net ofrecían mejores resultados en determinados casos.

Keywords: MRI craneal, detección, segmentación, Yolo, U-Net.

1 Introduction

La detección temprana y precisa de tumores cerebrales es un desafío crucial en la medicina moderna debido a las implicaciones directas sobre la planificación del tratamiento y el pronóstico del paciente. Con los avances recientes en inteligencia artificial y aprendizaje profundo, los modelos de segmentación y detección, como U-Net y YOLO, han demostrado ser herramientas prometedoras en la identificación automatizada de patologías en imágenes médicas. Este trabajo presenta un análisis comparativo del desempeño de varios modelos U-Net y de YOLO para la detección de tumores cerebrales, utilizando un dataset compuesto por 2,146 imágenes de resonancia magnética. Estas resonancias incluyen bounding boxes que señalan donde se encuentra el tumor.

2 Estado del arte

La detección y segmentación de tumores cerebrales ha sido un área de investigación activa en los últimos años, especialmente con la incorporación de técnicas de aprendizaje profundo. Estas metodologías han demostrado ser significativamente más precisas, eficientes y escalables en comparación con los enfoques tradicionales basados en técnicas manuales y métodos semi-automáticos.

Cabe destacar BraTS, un dataset de segmentación de resonancias magnénicas de 3 dimensiones sobre el que se construyen algunas competiciones donde se usan distintos métodos para obtener los mejores resultados posibles en esta tarea. Se han llegado a obtener resultados de 0.85 de IoU sobre este dataset.

El modelo YOLO (You Only Look Once) se ha desta-

cado en tareas de detección en tiempo real debido a su capacidad para procesar imágenes completas en una sola etapa. Aunque tradicionalmente se ha aplicado en dominios generales como la visión por computador, las versiones más recientes han sido adaptadas para imágenes médicas mediante ajustes en sus anchors y estrategias de entrenamiento. En el contexto de tumores cerebrales, YOLO ofrece una alternativa para la localización rápida de regiones de interés, aunque con ciertas limitaciones en la precisión de los bordes en comparación con los modelos de segmentación.

3 Metodología

En este estudio se implementaron y compararon dos enfoques principales para la detección y segmentación de tumores cerebrales en imágenes de resonancia magnética (IRM): modelos basados en la arquitectura U-Net y un modelo YOLO.

3.1 Preprocesamiento de los datos para U-Net

Los datos contaban con 2146 imágenes extraidas de resonancias magnéticas. Además de estas imágenes, el dataset incluye un archivo json en el que, por cada imagen, se incluyen las coordenadas de los vértices de la bounding box que contiene al tumor.
Lo primero que se ha hecho es la división del dataset en un conjunto de entrenamiento con 1502 imágenes, uno de validación con 215 y uno de test con 429.
Para adaptar las imágenes a la arquitectura U-net se han creado máscaras binarias que continen 0 en la parte de fuera de la bounding box y 1 dentro. Para la creación de estas máscaras, se ha usado el método fillpoly de la librería CV2. Las máscaras generadas se han ido almacenando en una carpeta con el mismo nombre que la imagen correspondiente.
Para el procesado de estas imágenes y sus máscaras se ha creado una clase DataGen que hereda de la clase Sequence de Keras. Esta clase recibe la dirección de la carpeta de las imágenes y de las máscaras, el tamaño del batch, el tamaño de las imágenes y el json. Mediante el método get-item se va devolviendo los batches correspondientes de las imágenes normalizadas y escaladas y sus correspondientes máscaras.

3.2 Modelo U-Net

Los modelos U-Net son un tipo de autoencoders convolucionales que incluyen conexiones residuales entre el encoder y el decoder. El modelo U-Net cuenta con una arquitectura simétricas basada en bloques convolucionales. Cada bloque convolucional incluye dos capas convolucionales con el mismo número de filtros y función de acticación ReLU y capas batch normalization debido a la profundidad del modelo. En el encoder se usan capas Conv2d y maxpooling para reducir la dimensión de la imagen. Sin embargo, en el decoder se usan capas Conv2d transpose y capas concatenate que unen el resultado de estas capas con los resultados de las capas del encoder (conexiones residuales).
En cuanto a la función de coste usada para el entrenamiento del modelo se ha usado $\text{Dice Loss}(P, G) = 1 - \frac{2 \cdot |P \cap G|}{|P| + |G|}$ que mide la superposición entre la máscara creada por el modelo y la máscara real. La métrica usada para evaluar los modelos y comparar los distintos U-Net es la intersección sobre la unión (IoU).

3.3 Modelo Yolo

Para la detección de tumores, se empleó la versión YOLOv11n de la librería Ultralytics. El modelo Yolo (You Only Look Once) es un modelo usado para la detección entrenado con distintos tipo de objetos. En este caso, para reentrenar el modelo con los datos del dataset hace falta darles una estructura específica para la cual se usa la función convert-and-organize-coco-to-yolo. Además de organizar los archivos de la manera necesaria hace falta crear un archivo yml en el que se ponen los directorios donde se encuentran los distintos archivos de train, val y test y el número de categorías de objetos a detectar, en este caso una. Al entrenar con esta librería se generan una serie de métricas como puede ser mAP con distintos valores de IoU. Esta métrica será utilizada para medir la efectividad de Yolo frente a los modelos U-Net.

El modelo Yolo no da como output una máscara binaria como hacían los modelos U-Net, sino que, da como output las coordenadas de la bounding box que contiene al tumor y la probabilidad de contener un tumor.

4 Experimentos

Para evaluar el desempeño de los modelos propuestos en la detección y segmentación de tumores cerebrales, se realizaron una serie de experimentos divididos en dos etapas principales: entrenamiento de modelos U-Net y entrenamiento de un modelo YOLO. A continuación, se detallan los procedimientos y configuraciones experimentales.

4.1 Modelos U-Net

Para intentar conseguir el mejor modelo U-Net, se han probado distintas arquitecturas. Todas las arquitecturas se han entrenado usando Dice loss como función de coste, early stopping sobre el loss de validación y IoU como métrica. La arquitectura usada está basada en los bloques convolucionales explicados en el apartado de modelos. Todos los modelos tienen como entrada imágenes de 256 por 256 píxeles.

4.1.1 Primer modelo U-Net

Este modelo es el más pequeño de todos los que se han probado. El encoder y el decoder cuentan con 3 bloques convolucionales de 32, 64 y 128 filtros. El cuello de botella de este modelo cuenta con 256 filtros. La mejor IoU de este modelo se obtuvo en el epoch 35 y fue de 0.4972.

4.1.2 Segundo modelo U-Net

Este modelo es muy similar al primero, tiene también 3 bloques convolucionales tanto en el encoder como en el decoder. Sin embargo, el número de filtros de cada bloque es de 64, 128 y 256. El cuello de botella tiene 512 filtros.
Este modelo obtuvo un resultado parecido al primer

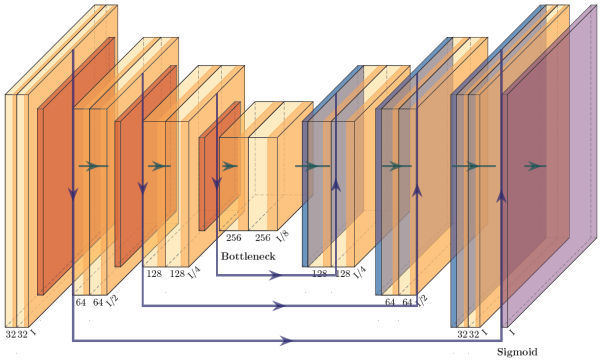

Figure 1: Arquitectura del primer modelo

modelo en cuanto a IoU sobre el conjunto de validación. El mejor IoU lo obtuvo en el epoch 38 y fue de 0.4901.

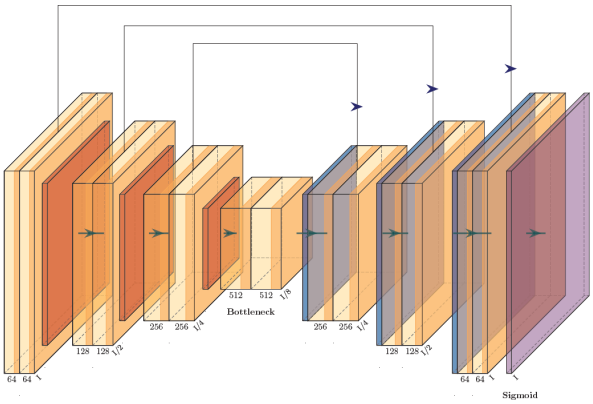

Figure 2: Arquitectura del segundo modelo

4.1.3 Tercer modelo U-Net

Para intentar aumentar la efectividad del modelo en este modelo se usa un bloque convolucional más tanto en el encoder como en el decoder. Es decir, este modelo cuenta con 4 bloques convolucionales de 64, 128, 256, 512 filtros en el encoder y en el decoder. Y un cuello de botella de 1024 filtros.
Este es el modelo que mostró mejores resultados,

obteniendo en el epoch 25 un IoU de 0.5465 sobre el conjunto de validación. Sin embargo, según aumentaban los epoch se podía apreciar un cierto overfitting siendo el IoU del modelo sobre el conjunto de entrenamiento bastante mayor al que ofrecía sobre el conjunto de validación.

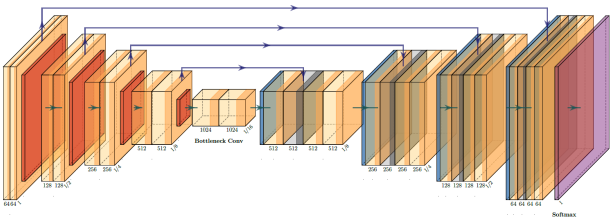

Figure 3: Arquitectura del tercer modelo

4.1.4 Cuarto modelo U-Net

Para intentar evitar el overfitting del tercer modelo, se ha usado la misma arquitectura pero añadiendo una capa dropout en el cuello de botella de 0.3. Este dropout se ha puesto en el cuello de botella, puesto que, tras probar a ponerlo en otras capas es el modelo que mejores resultados ha dado. A pesar de esto, el modelo ofrece peores resultados que el tercero. En el epoch número 20 tuvo un IoU sobre el conjunto de validación de 0.5229.

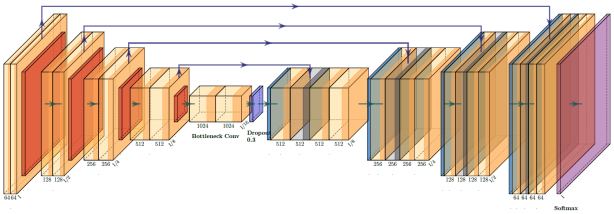

Figure 4: Arquitectura del cuarto modelo

4.2 Yolo

El modelo Yolo se reentrenó usando un modelo Yolo 11n de la librería ultralytics. Para su entrenamiento se usaron imágenes de 640 por 640 y se entrenó durante 100 epochs.

En la siguiente gráfica se puede ver como evoluciona la métrica mAP con distintos umbrales de IoU:

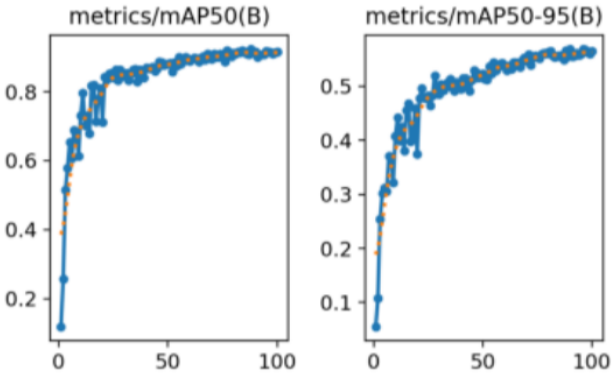

Figure 5: mAP de Yolo en el entrenamiento

5 Resultados

En este apartado se evaluarán usando distintas métricas los modelos U-Net y Yolo. Además, se comparán los distintos modelos usando distintas métricas como IoU y mAP. Todos los modelos se evaluarán sobre el conjunto de test.

5.1 Comparación de los modelos U-net

Para la comparación de los distintos modelos U-Net la principal métrica para medir la eficacia del modelo es la IoU evaluando en el conjunto de test. En esta métrica el modelo que mejores resultados dio fue el tercer modelo que dio 0.54, seguido del cuarto que obtuvo 0.52. El primero, a pesar de ser más pequeño que el segundo modelo, obtuvo una IoU de 0.50, mientras que el segundo obtuvo 0.47.
En la métrica accuracy los resultados obtenidos siguen el mismo orden que en la IoU. Sin embargo, la métrica accuracy en estos modelos no es tan representativa, puesto que, las máscaras estaban compuestas principalmente por ceros (fondo). Por tanto, un modelo que diese máscaras de todo ceros tendría

una accuracy bastante elevada. Para la evaluación del modelo también se han mostrado algunos ejemplos de predicciones hechos sobre el conjunto de test. En estos ejemplos se puede ver como el tercer modelo es el que mejor hace las predicciones.

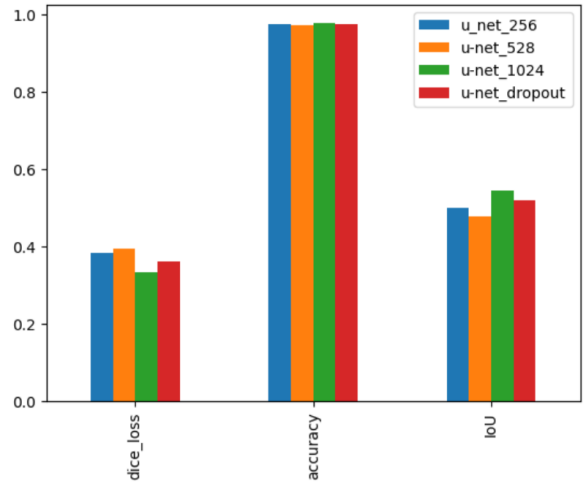

Figure 6: Métricas de los distintos modelos U-Net

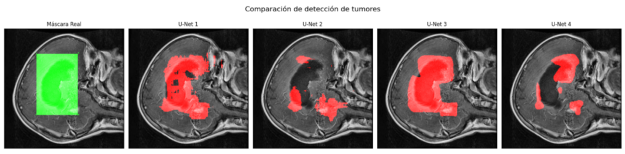

Figure 7: Predicciones de los modelos U-Net

5.2 Comparación de Yolo con los modelos U-Net

Para la comparación de U-Net con Yolo la métrica IoU no es la más apropiada. Puesto que, el resultado de Yolo siempre son bounding boxes, por tanto, siempre tiende a tener más solapamiento con la bounding box de la etiqueta. Sin embargo, lo que se busca con estos modelos no es la detección exacta de la bounding box de la etiqueta, sino que, se busca saber la zona en la cual está ubicado el tumor. Por tanto, una métrica que puede ser más apropiada es el mAP sobre la IoU. Esta métrica lo que mide es en qué proporción de las predicciones el IoU está por encima de un cierto umbral. Es decir, si el umbral de IoU que se usa es 0.5, la métrica mAP dirá qué proporción de las predicciones han tenido un IoU entre la predicción y la etiqueta de más del 50%. Se ha usado el umbral 0.5 puesto que se ha considerado que si una predicción tiene un IoU mayor que 0.5 la detección del tumor ha sido exitosa.
Usando esta métrica Yolo obtiene los mejores resultados con un mAP de 0.865, seguido del tercer modelo que obtiene 0.69, después el cuarto con 0.65, el primero con 0.61 y por último el segundo que obtiene 0.58. A pesar de que los resultados obtenidos por

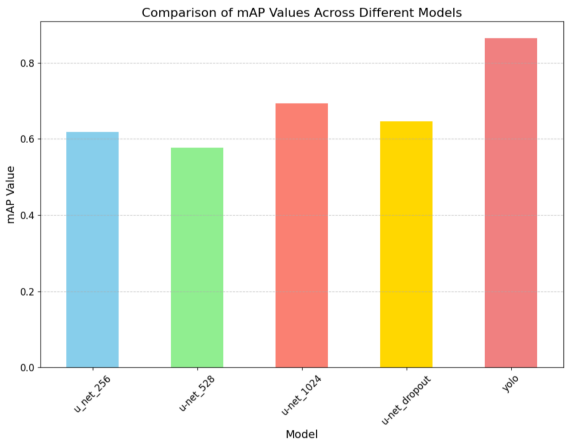

Figure 8: Comparación del mAP

Yolo son mejores teniendo en cuenta un mAP sobre IoU de 0.5, si bajamos el umbral de IoU los resultados obtenidos por los modelos U-Net son bastante mejores, mientras que los obtenidos por Yolo no aumentan demasiado. Además, mostrando las predicciones de todos los modelos, las predicciones de Yolo y el tercer modelo U-net suelen ser bastante parecidas.

6 Conclusiones

Tanto el tercer modelo U-net como Yolo han mostrado buenos resultados a la hora de hacer la de-

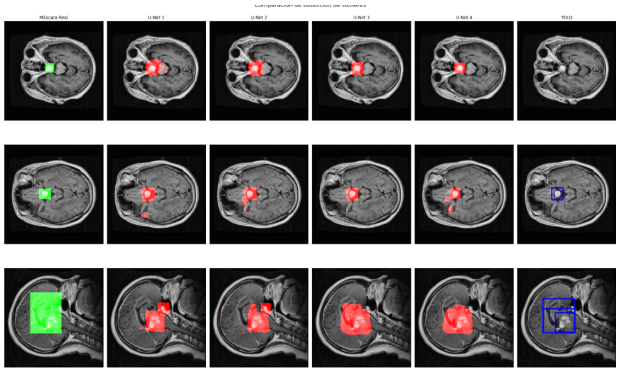

Figure 9: Predicciones de los distintos modelos

tección. Si bien, mirando únicamente las métricas Yolo está claramente por encima, se ha visto como los modelos U-net ofrecen mejores resultados para algunas imágenes. Cabe destacar que el tamaño del dataset era bastante pequeño, esto hace que los modelos U-Net tengan peores resultados, puesto que, el modelo yolo utilizado ya estaba entrenado con distintos tipos de imágenes. De hecho probablemente el modelo yolov11n ya hubiese sido entrenado con imágenes de tumores.

Una ventaja de los modelos U-Net frente a Yolo es que estos modelos podrían ser reentrenados en datasets de segmentación. Probablemente, reentrenando estos modelos con dataset de segmentación permitiría conseguir mejores resultados que un modelo que solo hubiese sido entrenado con esos datos.

En cambio, el uso de yolo también da ventajas. La principal es que indica la probabilidad de que haya tumor o no. Esto puede ser útil a la hora de aplicar estos modelos en casos reales, puesto que, este porcentaje puede utilizarse para hacer una clasificación binarias entre resonancias con tumor y sin tumor.

References

[1] B. H. Menze et al., "The Multimodal Brain Tumor Image Segmentation Benchmark (BRATS)," in IEEE Transactions on Medical Imaging, vol. 34, no. 10, pp. 1993-2024, Oct. 2015, doi: 10.1109/TMI.2014.2377694.

[2] Ghaffari, M., Sowmya, A., Oliver, R. (2019). Automated brain tumor segmentation using multimodal brain scans: a survey based on models submitted to the BraTS 2012–2018 challenges. IEEE reviews in biomedical engineering, 13, 156-168.

[3] Kang, M., Ting, C. M., Ting, F. F., Phan, R. C. W. (2023, October). RCS-YOLO: A fast and high-accuracy object detector for brain tumor detection. In International Conference on Medical Image Computing and Computer-Assisted Intervention (pp. 600-610). Cham: Springer Nature Switzerland.

[4] Allah, A. M. G., Sarhan, A. M., Elshennawy, N. M. (2023). Edge U-Net: Brain tumor segmentation using MRI based on deep U-Net model with boundary information. Expert Systems with Applications, 213, 118833.

[5] Baldovino, R. G., Vidad, A. J. P., Abastillas, R. P. B., Bugtai, N. T., Dadios, E. P., Vicerra, R. R. P., ... Roxas Jr, N. R. (2024). Comprehensive analysis on Ultralytics-supported YOLO models for detection and recognition of large office objects for indoor navigation. Procedia Computer Science, 246, 3851-3858.