

Universidad Complutense de Madrid
Facultad de Estudios Estadísticos
Máster Big Data y Data Science



**Detección y segmentación de
tumores cerebrales en imágenes de
resonancia magnética con técnicas
de Deep Learning**

MSc. Carlos Gordillo Olivera

2024

Índice

1	Introducción	2
1.1	Imágenes por resonancia magnética	2
1.2	Tumores cerebrales: generalidades	2
1.3	Motivación del trabajo	3
1.4	Organización del Documento	4
2	Detección de tumores	5
2.1	Descripción del dataset	5
2.2	Preprocesamiento de las imágenes	6
2.3	Modelos	7
2.3.1	Naive bayes	8
2.3.2	Random forest	8
2.3.3	Logistic regression	9
2.3.4	Decision tree	9
2.3.5	KNN	10
2.3.6	Red neuronal convolucional	10
2.4	Resumen de modelos	11
3	Segmentación de tumores	12
3.1	Descripción del dataset	12
3.2	Preprocesamiento de las imágenes	12
3.3	Modelo: UNET	13
4	Puesta en producción y conclusiones	16

Capítulo 1

Introducción

1.1 Imágenes por resonancia magnética

La utilización de imágenes por resonancia magnética (MRI) ha revolucionado el campo de la neurología al proporcionar una visión detallada y no invasiva del cerebro y sus estructuras. La capacidad de obtener imágenes de alta resolución de los tejidos cerebrales en diferentes planos anatómicos ha permitido a los médicos y investigadores diagnosticar y tratar una amplia gama de trastornos neurológicos con mayor precisión y efectividad.

La MRI es especialmente invaluable en el estudio de enfermedades cerebrales como los tumores, los accidentes cerebrovasculares, las enfermedades neurodegenerativas y los trastornos del desarrollo cerebral. Estas imágenes proporcionan información crucial sobre la localización, el tamaño, la extensión y la naturaleza de las anomalías cerebrales, lo que ayuda a los médicos a planificar intervenciones quirúrgicas, a diseñar estrategias de tratamiento y a realizar un seguimiento preciso de la progresión de la enfermedad.

Además, la MRI ofrece la capacidad única de detectar cambios sutiles en la estructura y la función del cerebro que pueden no ser visibles en otros tipos de imágenes médicas. Esto es fundamental para comprender la fisiopatología de las enfermedades cerebrales, identificar biomarcadores tempranos de enfermedades y evaluar la eficacia de los tratamientos.

Para una profundización mayor de los conceptos físicos detrás de esta técnica de imágenes recurrir a la tesis de maestría.

1.2 Tumores cerebrales: generalidades

Se dice que un tumor cerebral es una masa de células que se forma en el tejido cerebral o en ubicaciones cercanas a este, incluyendo los nervios, glándula pituitaria, glándula pineal y las meninges, como se muestra en la Fig. 1.1.

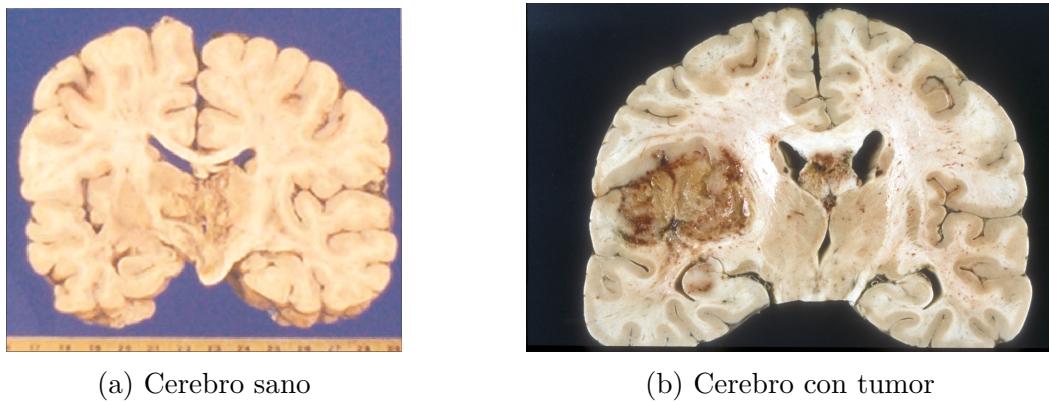


Figure 1.1: Corte coronal de cerebro ex vivo sano y con un tumor.

Los tumores cerebrales se pueden clasificar en primarios o secundarios, dependiendo de si se originan en el propio cerebro o se diseminan desde otra parte del cuerpo (también conocidos como metastásicos). Estas estructuras pueden variar en tamaño, donde algunos, incluso pequeños, pueden provocar síntomas graves y evidentes, mientras que otros de mayor tamaño pueden pasar desapercibidos debido a la falta de síntomas.

En términos generales, algunos tumores primarios son benignos, lo que significa que no son cancerosos. Sin embargo, con el tiempo, pueden crecer y afectar el tejido cerebral circundante al comprimirlo. Por el contrario, los tumores cancerosos pueden desarrollarse de manera rápida y agresiva, invadiendo y destruyendo el tejido cerebral sano a su alrededor.

1.3 Motivación del trabajo

La detección y segmentación de tumores cerebrales representan una tarea de gran importancia en el ámbito médico, según lo indicado por la Organización Mundial de la Salud (OMS). La capacidad de identificar y delimitar con precisión la ubicación y extensión de estos tumores es fundamental para la planificación de intervenciones quirúrgicas efectivas. La resección quirúrgica de los tumores cerebrales depende en gran medida de una detección temprana y una segmentación precisa, lo que puede significar la diferencia entre la vida y la muerte, así como entre la salud y la discapacidad permanente para los pacientes. Además, la detección oportuna de estos tumores permite iniciar el tratamiento adecuado en las etapas iniciales de la enfermedad, lo que puede mejorar significativamente los resultados clínicos y la calidad de vida de los pacientes al evitar complicaciones graves y preservar las funciones cerebrales vitales.

En este contexto, las técnicas de inteligencia artificial y aprendizaje profundo han emergido como herramientas poderosas y prometedoras. La aplicación de algoritmos de inteligencia artificial, como las redes neuronales convolucionales, permite procesar grandes volúmenes de datos de imágenes médicas de manera rápida y precisa, facilitando la identificación automática de características relevantes y la segmentación precisa de los tejidos tumorales. El uso de estas herramientas no solo agiliza el proceso de diagnóstico, sino que también mejora la precisión y fiabilidad de las predicciones, lo que puede conducir a una detección más temprana y un tratamiento más efectivo de los tumores cerebrales. En última instancia, la integración de la inteligencia artificial y las técnicas de aprendizaje profundo en la práctica clínica tiene el potencial de rev-

olucionar el manejo de esta enfermedad, mejorando los resultados clínicos y la calidad de vida de los pacientes.

1.4 Organización del Documento

La organización del documento se estructura en dos partes principales. En la primera parte, se aborda la detección de tumores cerebrales, mientras que la segunda parte se centra en la segmentación de los mismos. Cada una de estas partes se desarrolla en un capítulo independiente, donde se discuten los fundamentos teóricos, las metodologías utilizadas y los resultados obtenidos en cada etapa del proceso.

Tras la presentación de los resultados, se procede a realizar un análisis detallado de las métricas correspondientes, evaluando la efectividad y precisión de los modelos desarrollados en la detección y segmentación de los tumores cerebrales. Este análisis se realiza con el objetivo de validar el rendimiento de los algoritmos y determinar su utilidad en la práctica clínica.

Finalmente, en el último capítulo del documento, se describe el diseño e implementación de una API y una página web interactiva sencilla. Estas herramientas permiten a los usuarios interactuar con los modelos desarrollados, facilitando su uso en entornos clínicos y de investigación. Se detallan las funcionalidades y características de la API y la página web, así como los pasos necesarios para su implementación y utilización efectiva.

Todos los detalles de código, armado de modelos, cálculo de métricas, procesamiento de datos, así como la implementación de la API y front end se encuentra en el Github.

Capítulo 2

Detección de tumores

Link del dataset: <https://www.kaggle.com/datasets/preetviradiya/brian-tumor-dataset/data>

2.1 Descripción del dataset

La metadata informa que el dataset contiene 4600 imágenes en total, divididas en dos clases: *tumor* y *normal* según se observa en la Fig. 2.1. En este contexto va a entenderse a *normal* como *sin tumor*, pero no significa que no existan otras patologías por ejemplo isquemias.

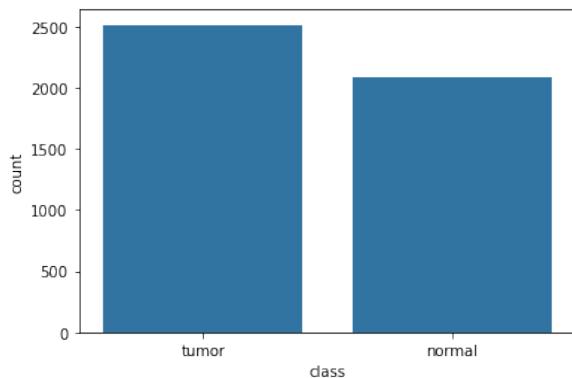


Figure 2.1: Distribución de clases de imágenes del dataset para detección.

En la Fig. 2.2 se observan algunas imágenes del dataset. En particular, las Fig. 2.2a, 2.2b y 2.2c corresponden a pacientes normales mientras que las Fig. 2.2d, 2.2e y 2.2f corresponden a pacientes con tumor cerebral.

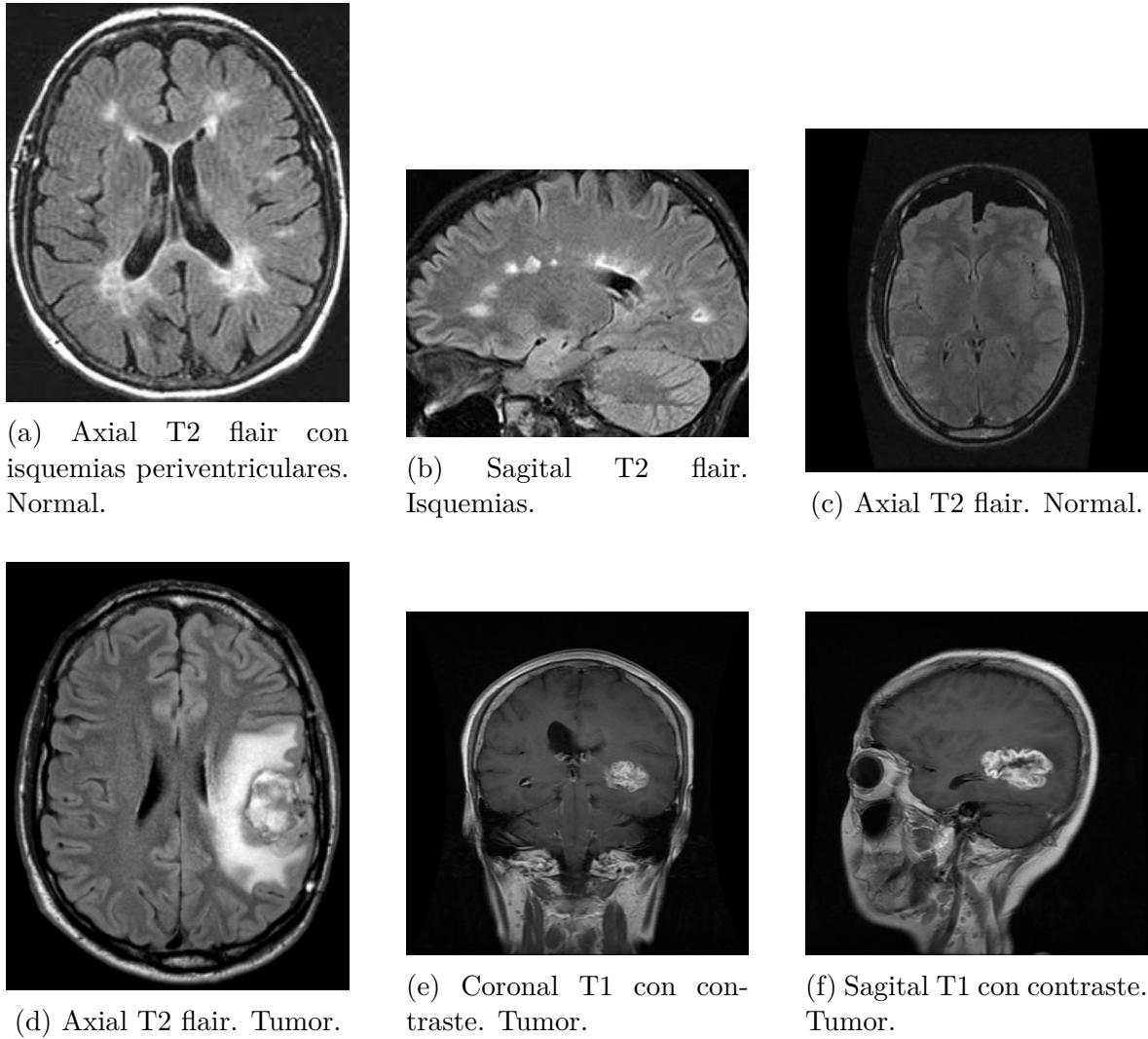


Figure 2.2: Muestra de imágenes del dataset.

Es interesante notar que la Fig. 2.2a es un corte axial T_2 flair de cerebro donde claramente se observa una anomalía del tejido. Las zonas hiperintensas (blancas) que rodean a los ventrículos de forma bilateral corresponden a isquemias típicas de la edad pero no es un tumor, por lo tanto la imagen es clasificada como de paciente sano.

Por otro lado, una patología isquémica también se observa en la Fig. 2.2d, en este caso está rodeando a un tumor. Por lo tanto, esta imagen se clasifica como tumoral.

2.2 Preprocesamiento de las imágenes

Las imágenes disponibles tenían no solo diferentes tamaños sino también diferentes relaciones de aspecto. La Fig. 2.3 muestra la distribución de tamaños, la recta roja corresponde a imágenes que sean cuadradas.

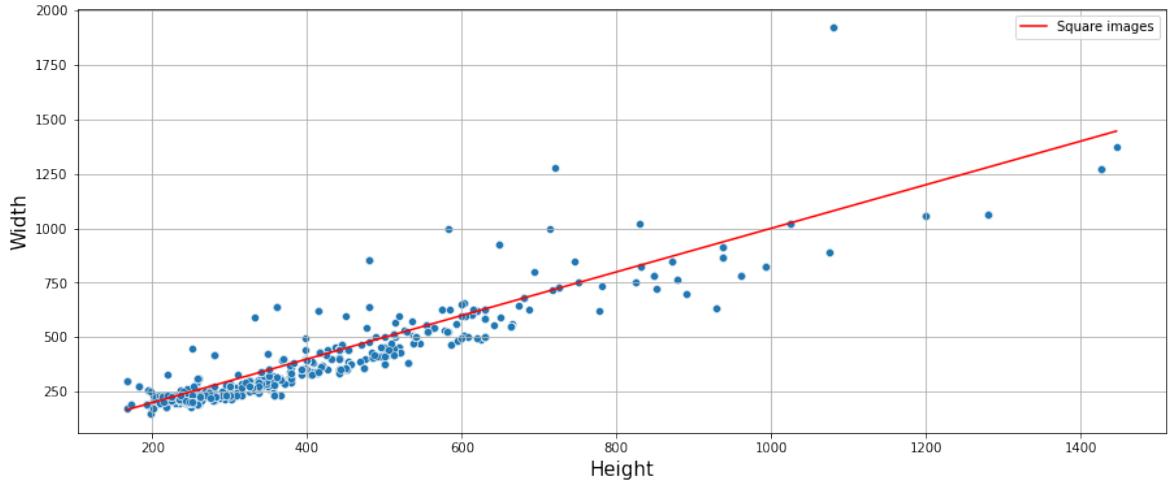


Figure 2.3: Tamaños de las imágenes del dataset

Con el objetivo de normalizar las imágenes, se optó por redimensionarlas a una resolución de 256×256 píxeles.

Además, se aplicó una conversión a escala de grises, dado que esta representación es ampliamente utilizada en el ámbito del diagnóstico médico, garantizando así una compatibilidad estándar para el análisis y la interpretación de las imágenes.

Un último preprocesamiento que se aplicó es normalizar el rango de valores de los píxeles de la imagen, originalmente cada píxel $p_{ij} \in [0, 255]$ y luego de la transformación $p_{ij} \in [0, 1]$.

2.3 Modelos

Antes de aplicar los diferentes modelos se realizó la partición en *train/test* en una proporción 80/20.

Las variables predictoras son los propios píxeles de cada imagen, mientras que el target es binario: tumor/sano.

Inicialmente se aplicaron modelos tradicionales de *machine learning*. Para estos modelos anteriores se realizó un preprocesamiento más, la conversión de las imágenes en formato de matriz (256×256) a vector de dimensión $256 \cdot 256 = 65536$. Los modelos trabajados fueron:

- Naive bayes
- Random Forest Classifier
- Logistic Regression
- Decision Tree Classifier
- KNN

2.3.1 Naive bayes

Este modelo por ser el más simple de todos constituye un baseline. Luego del entrenamiento se obtuvieron la matriz de confusión y curva roc tanto para train como para test, las cuales son mostradas en la Fig. 2.4.

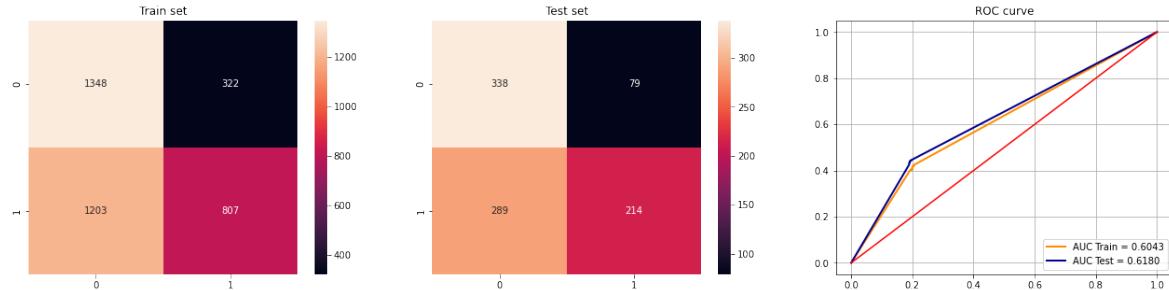


Figure 2.4: Métricas para el modelo Naive-Bayes

En la Tabla 2.1 se detallan las métricas derivadas de la matriz de confusión. Se observa que las métricas son bajas, lo cual es coherente con que es un modelo demasiado simple.

	Accuracy	Precision	Recall	AUC	F_1 score
Train	0.5856	0.7148	0.4015	0.6043	0.5142
Test	0.6	0.7304	0.4254	0.618	0.5377

Table 2.1: Métricas para el modelo Naive-Bayes

2.3.2 Random forest

Luego del entrenamiento se obtuvieron la matriz de confusión y curva roc tanto para train como para test, las cuales son mostradas en la Fig. 2.5.

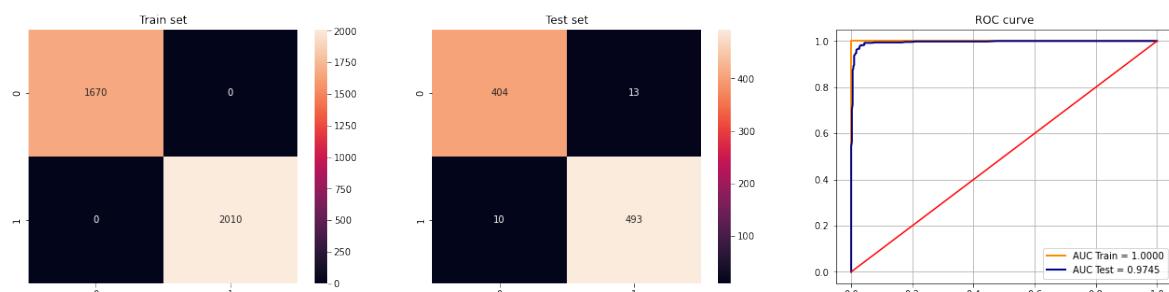


Figure 2.5: Métricas para el modelo Random Forest

En la Tabla 2.2 se detallan las métricas derivadas de la matriz de confusión. Se observa que las métricas son bajas, lo cual es coherente con que es un modelo demasiado simple.

	Accuracy	Precision	Recall	AUC	F_1 score
Train	1.0	1.0	1.0	1.0	1.0
Test	0.975	0.9743	0.9801	0.9745	0.9772

Table 2.2: Métricas para el modelo Random Forest

Es notable que este modelo resulta perfecto para el conjunto de entrenamiento, lo cual haría sospechar de un posible overfitting, pero viendo las métricas correspondientes al conjunto de test esto parecería estar descartado.

2.3.3 Logistic regression

Luego del entrenamiento se obtuvieron la matriz de confusión y curva roc tanto para train como para test, las cuales son mostradas en la Fig. 2.6.

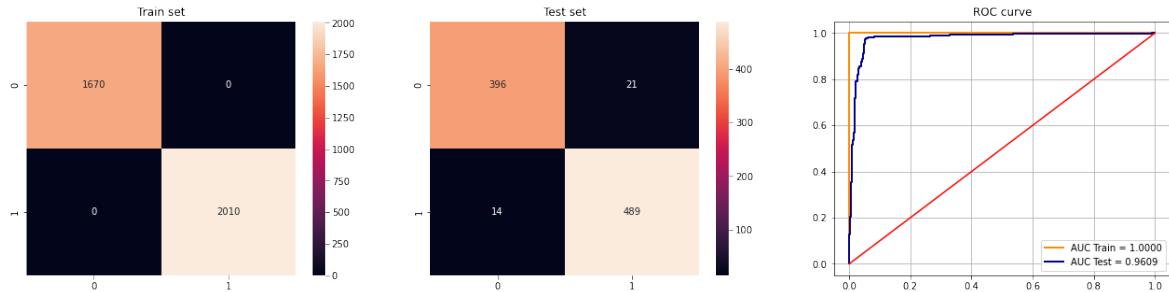


Figure 2.6: Métricas para el modelo logístico

En la Tabla 2.3 se detallan las métricas derivadas de la matriz de confusión.

	Accuracy	Precision	Recall	AUC	F_1 score
Train	1.0	1.0	1.0	1.0	1.0
Test	0.962	0.9588	0.9722	0.9609	0.9654

Table 2.3: Métricas para el modelo logístico

2.3.4 Decision tree

Luego del entrenamiento se obtuvieron la matriz de confusión y curva roc tanto para train como para test, las cuales son mostradas en la Fig. 2.7.

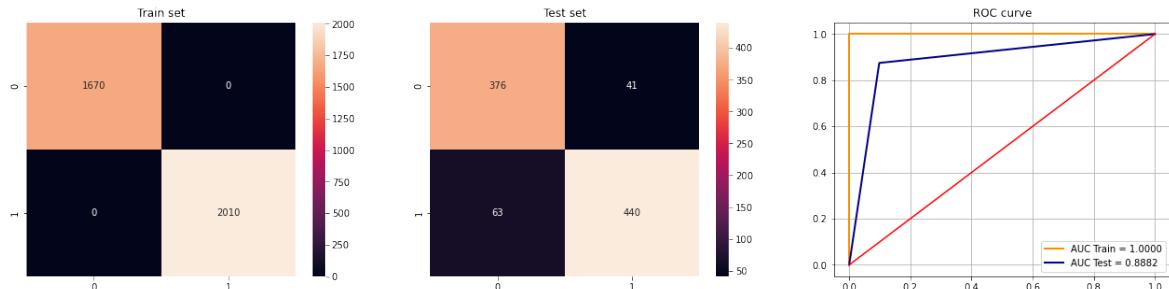


Figure 2.7: Métricas para el modelo de árbol de decisión

En la Tabla 2.4 se detallan las métricas derivadas de la matriz de confusión.

	Accuracy	Precision	Recall	AUC	F_1 score
Train	1.0	1.0	1.0	1.0	1.0
Test	0.887	0.9148	0.8748	0.8882	0.8943

Table 2.4: Métricas para el modelo de árbol de decisión

Comparando las métricas obtenidas para los conjuntos de entrenamiento y prueba, se evidencia claramente que el modelo sufre de sobreajuste. Aunque se podría haber intentado mitigar este sobreajuste mediante la optimización de hiperparámetros, se descartó esta opción debido a la disponibilidad de un modelo superior (random forest) y al considerable aumento en el tiempo de ajuste de dicho modelo.

2.3.5 KNN

Luego del entrenamiento se obtuvieron la matriz de confusión y curva roc tanto para train como para test, las cuales son mostradas en la Fig. 2.8.

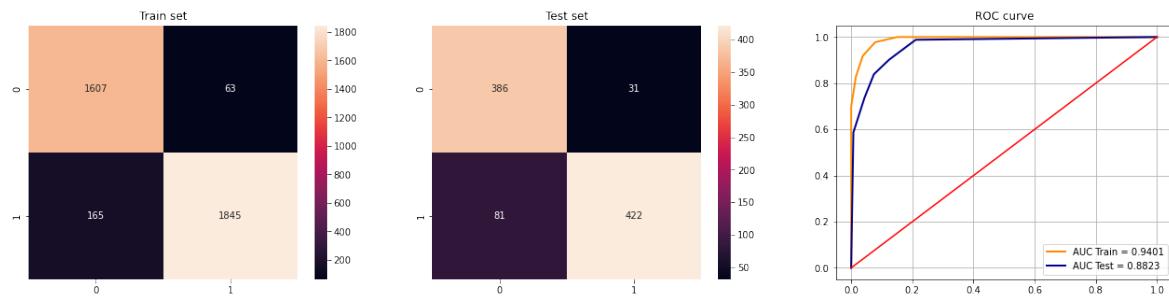


Figure 2.8: Métricas para el modelo KNN

En la Tabla 2.5 se detallan las métricas derivadas de la matriz de confusión.

	Accuracy	Precision	Recall	AUC	F_1 score
Train	1.0	1.0	1.0	1.0	1.0
Test	0.887	0.9148	0.8748	0.8882	0.8943

Table 2.5: Métricas para el modelo KNN

2.3.6 Red neuronal convolucional

Finalmente se aplicó un modelo de *deep learning*, una red neuronal convolucional. Básicamente consta de cuatro bloques *convolucional*, *batch normalization*, *max pooling*, una capa de *flatten* y luego tres bloques de capa *densa* y *dropout* como regularización. El optimizador utilizado fue *adam*, la función de pérdida *binary crossentropy* tomando como métrica *accuracy* dado que el dataset estaba razonablemente balanceado. Se entrenó la red con un *batch size* de 10, *validation split* de 0.2 y con 100 *epochs*.

Antes de calcular las métricas correspondientes de clasificación para este modelo, se procedió a investigar si hubo un overfitting durante el entrenamiento. Para esto se graficó la función de pérdida y el accuracy tanto de train como de test para cada epoch. La gráfica se muestra en la Fig. 3.3.

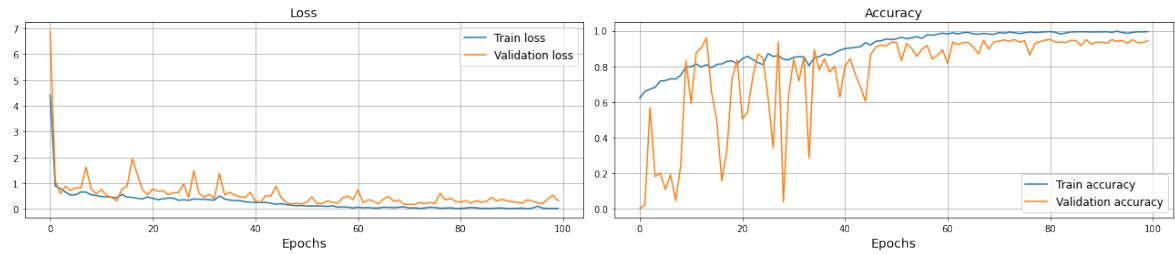


Figure 2.9: Función de pérdida y accuracy para el modelo de red convolucional.

Se observa que en principio no hay un overfitting. Al momento de calcular las predicciones, el modelo entrega una probabilidad de pertenecer a cada clase, por lo tanto se eligió un valor de umbral, el cual fue de 0.59, es decir, aquellas probabilidades mayores a 0.59 serían clasificadas como 1, es decir, con tumor y viceversa. El detalle de estos cálculos se encuentra en el código disponible en el Github. Una vez teniendo este valor umbral se calcularon las métricas, las cuales se muestran en la Tabla 2.6 y en la Fig. 2.10.

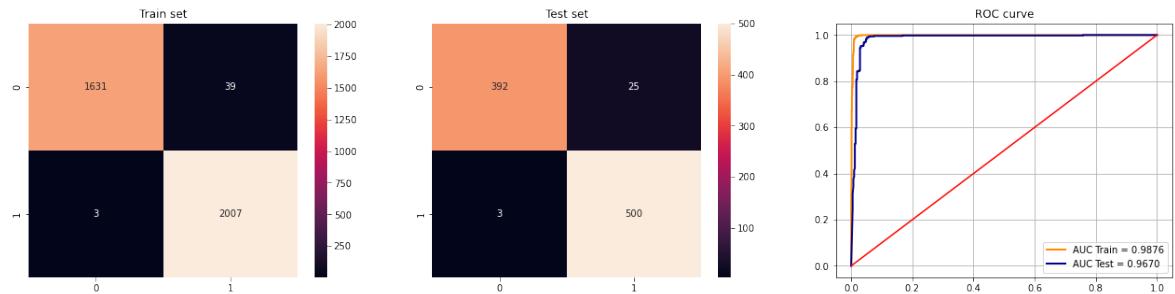


Figure 2.10: Métricas para el modelo de red convolucional

	Accuracy	Precision	Recall	AUC	F_1 score
Train	0.9886	0.9809	0.9985	0.9876	0.9896
Test	0.9696	0.9524	0.994	0.967	0.9728

Table 2.6: Métricas para el modelo de red convolucional

2.4 Resumen de modelos

En términos de métricas, el mejor modelo pareciera ser el Random Forest, sin embargo, considero que el de red neuronal convolucional es mejor porque exhibe un mejor recall.

En el contexto del problema, es importante no solo clasificar correctamente a los verdaderos positivos (pacientes efectivamente con tumores), sino también reducir los falsos negativos (pacientes con tumor pero clasificados como sanos).

Por este motivo, se optó por el modelo de redes neuronales convolucionales.

Capítulo 3

Segmentación de tumores

Link del dataset: <https://www.kaggle.com/datasets/pkdarabi/brain-tumor-image-dataset-semantic-segmentation>

3.1 Descripción del dataset

Este dataset contiene 2146 imágenes con sus respectivas máscaras y está subdividido en conjuntos de entrenamiento (1502 archivos), validación (429 archivos) y test (215).

En la Fig. 3.1 se observa la superposición de una imagen del dataset con su respectiva máscara, indicando la ubicación del tumor.

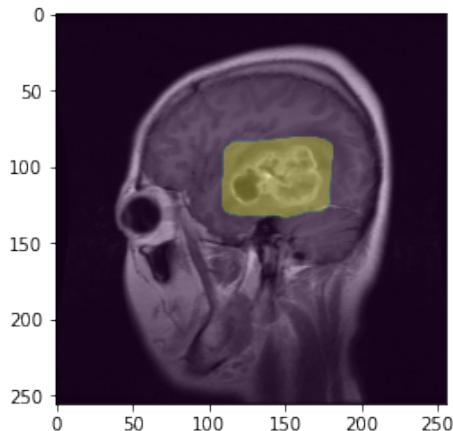


Figure 3.1: Imagen junto con la máscara indicando el tumor.

3.2 Preprocesamiento de las imágenes

Las imágenes disponibles tenían todas el mismo tamaño (640×640). Con el objetivo de normalizar las imágenes, se optó por redimensionarlas a una resolución de (256×256) píxeles.

Se aplicó una conversión a escala de grises, dado que esta representación es ampliamente utilizada en el ámbito del diagnóstico médico, garantizando así una compatibilidad estándar para el análisis y la interpretación de las imágenes.

Un último preprocessamiento que se aplicó es normalizar el rango de valores de los píxeles de la imagen, originalmente cada píxel $p_{ij} \in [0, 255]$ y luego de la transformación $p_{ij} \in [0, 1]$.

Las máscaras eran imágenes binarias del mismo tamaño que la imagen sobre la que segmenta.

3.3 Modelo: UNET

Como modelo de segmentación se utilizó la red ampliamente utilizada UNET, cuya arquitectura se muestra en la Fig. 3.2.

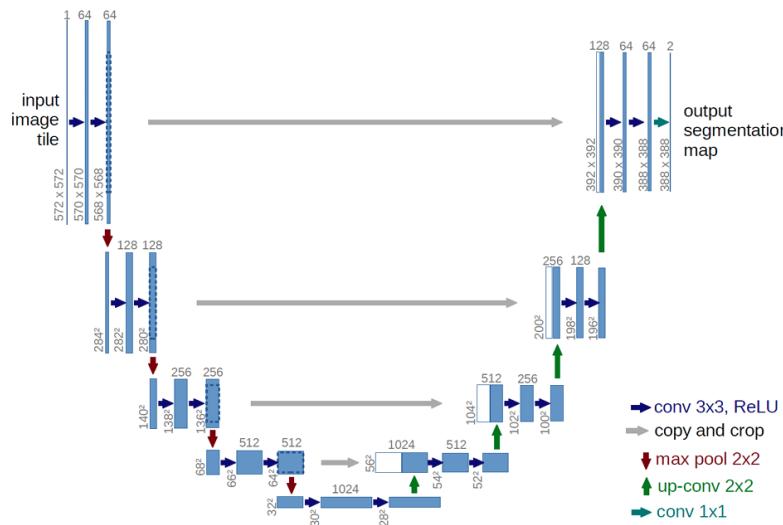


Figure 3.2: UNET

Esta arquitectura consta de un bloque de cuatro encoders, un bloque convolucional y un bloque de cuatro decoders.

El optimizador utilizado fue *adam*, la función de pérdida *Dice loss* tomando como métrica *Dice coefficient*. Se entrenó la red con un *batch size* de 10, *validation split* de 0.2 y con 100 *epochs*. Se implementó un callback de EarlyStopping.

Dado que el target de este modelo es una máscara, una imagen, tenía más sentido tomar como función de pérdida el coeficiente de Dice, el cual se define como:

$$DSC(X, Y) = \frac{2|X \cap Y|}{|X| + |Y|} \quad (3.1)$$

Este coeficiente es igual a 1 si y solo si $X = Y$, y 0 si X e Y son no conexos. Esto codifica matemáticamente la superposición de la máscara obtenida con la máscara target.

Antes de calcular las métricas correspondientes de clasificación para este modelo, se procedió a investigar si hubo un overfitting durante el entrenamiento. Para esto se graficó la función de pérdida y el accuracy tanto de train como de test para cada epoch. La gráfica se muestra en la Fig. 3.3.

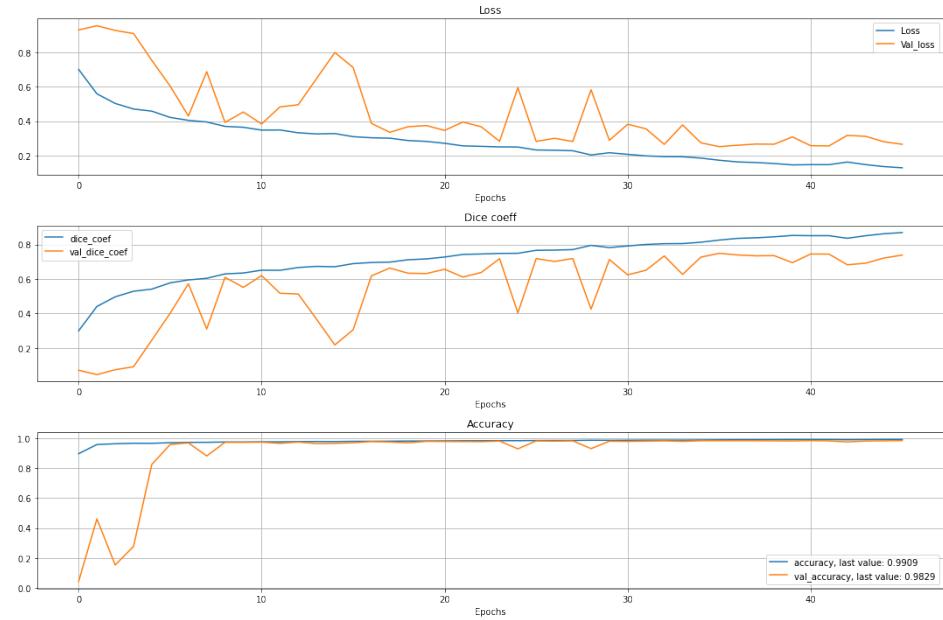


Figure 3.3: Función de pérdidas y accuracy para la red UNET.

Observando las gráficas anteriores se observa que el modelo no overfitea.

A continuación se procedió a calcular las métricas correspondientes. En este caso se calculó el coeficiente de Dice para cada conjunto de datos, los cuales se muestran en la Fig. 3.4.

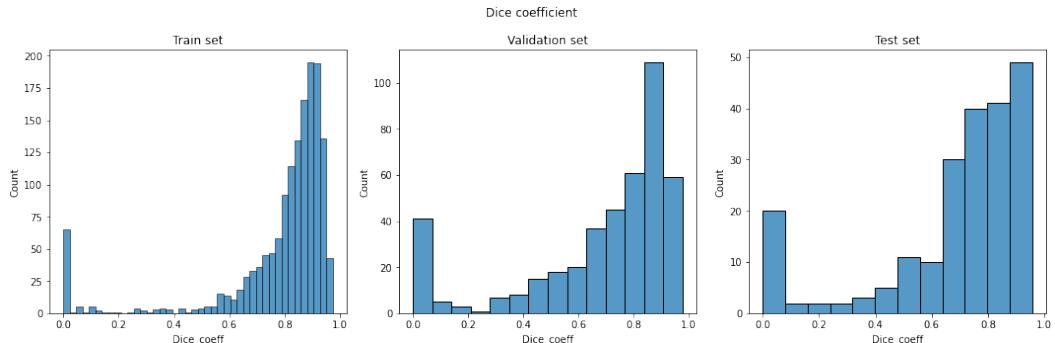
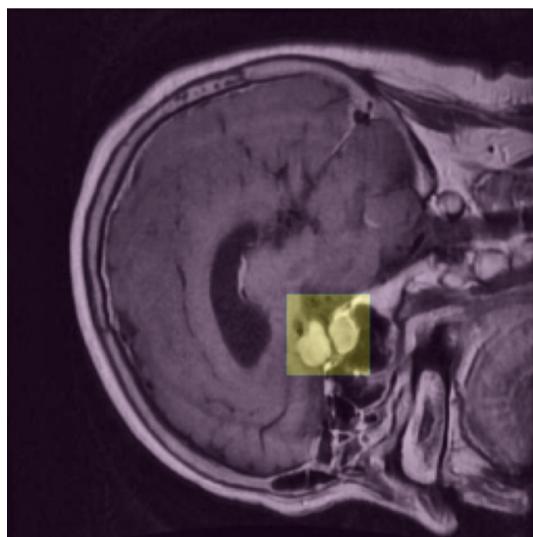
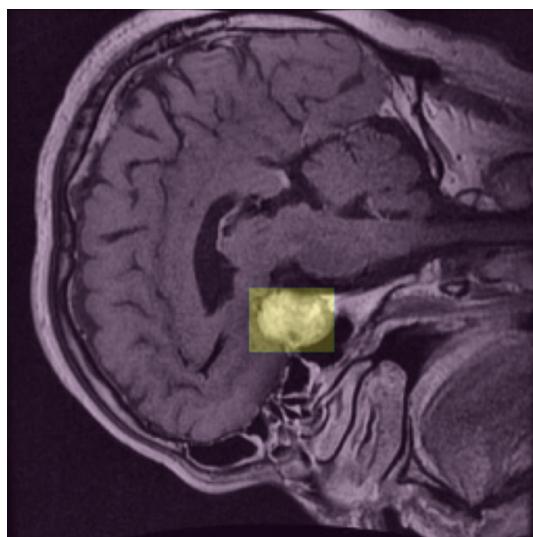
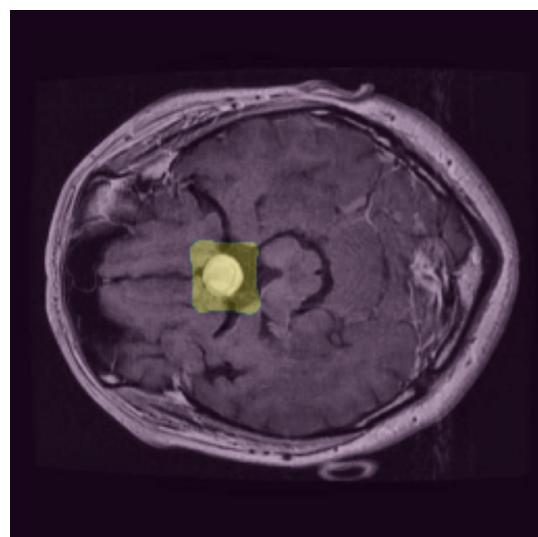
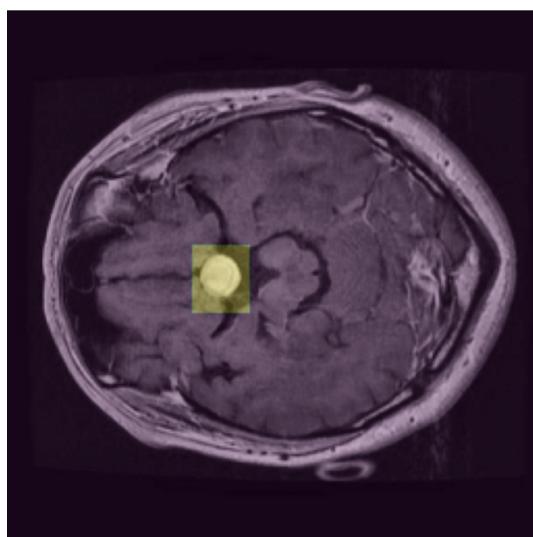


Figure 3.4: Coeficiente de Dice para cada conjunto de imágenes.

Se observa que en todos los casos, el valor máximo de la distribución está cercano a uno, indicando que el modelo funcionó razonablemente bien. Es notable observar que existen algunos coeficientes menores que uno. Esto se explica considerando que las máscaras usadas como target eran perfectamente rectangulares, mientras que las máscaras predichas por el modelo no lo son, sino que intentan adaptarse al tumor copiando su forma, como se observa en la Fig. 3.5.

En el código subido a Github está el detalle de los cálculos realizados, así como un slider para poder ver las imágenes con sus correspondientes máscaras.



Máscaras target

Máscaras predichas

Figure 3.5: Comparación de máscaras dadas y predichas por el modelo

Capítulo 4

Puesta en producción y conclusiones

Habiendo desarrollado y validado los modelos de detección y segmentación de tumores, se procedió a poner en producción ambos modelos. Esto se llevó a cabo en etapas:

- construcción de API para detección,
- construcción de API para segmentación,
- construcción de front end sencillo,
- dockerización.

El flujo de información sería el siguiente:

1. se carga una imagen,
2. la imagen se guarda y se da la opción de cargar otra o procesarla con el primer modelo para detectar un tumor,
3. si el tumor es detectado, se informa y se da la opción de aplicar el modelo de segmentación, mostrando la imagen con la máscara predicha,
4. si el tumor no es detectado y se solicita su segmentación, se informa que no hay tumor a segmentar.

A continuación se muestran imágenes del front end desarrollado y el procesamiento en la Fig. 4.1.

El diseño del front end y back end responden al Modelo Vista Controlador usando FastAPI. Los detalles están en el Github.

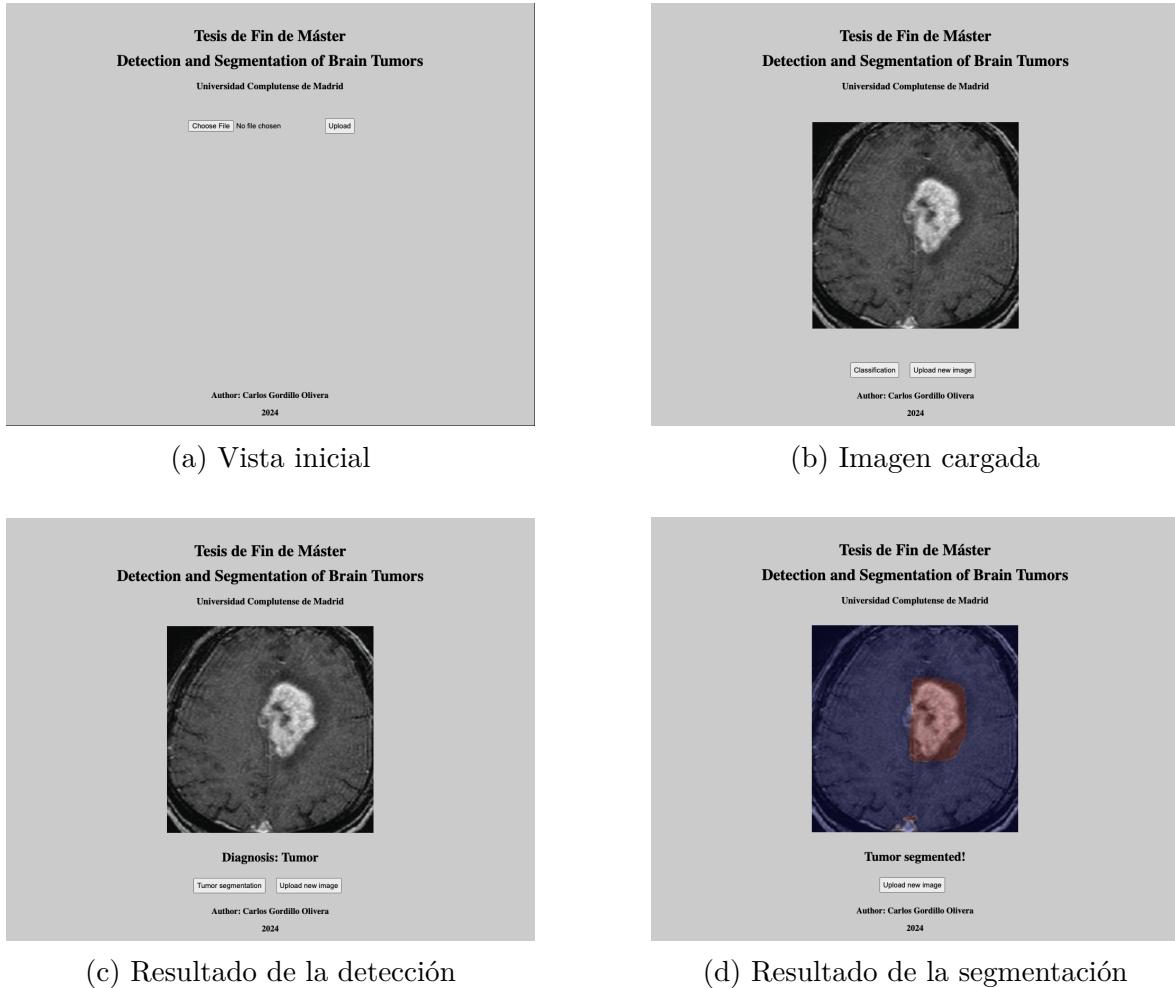


Figure 4.1: Flujo de información

Como conclusiones finales, se logró desarrollar un modelo de detección de tumores cerebrales. Dicho modelo fue evaluado y se descartó overfitting y presentó excelentes métricas de desempeño.

Se desarrolló también un modelo de segmentación de tumores, es decir, no únicamente detectar la presencia de tumores sino determinar la ubicación de los mismos. Se lo evaluó descartando overfitting y presentó buenas métricas.

Finalmente se desarrolló una API junto con un front end con el objetivo de poner en producción ambos modelos sumado a la visualización por parte del usuario.

Como trabajo a futuro podría considerarse agregar un label más al target que identifique el tipo de imagen. Por ejemplo, en algunos casos el modelo segmentó como tumor a los ventrículos laterales en imágenes potenciadas en T_2 , al ser el líquido brillante en estas imágenes, el algoritmo posiblemente los confundió con tumores en imágenes T_1 con contraste, donde también el tumor se ve brillante.

Podría considerarse también aplicar data augmentation sobre la segmentación tumoral, pero no es tan sencillo porque no solo habría que realizar transformaciones sobre las imágenes usadas para entrenar, sino que la máscara target debería sufrir las mismas transformaciones.