

# **Clasificación Automática de Tumores Cerebrales mediante el Uso de Redes Neuronales basadas en Vision Transformer**

Juan Manuel Medina Ramos

Tutor

Javier Sánchez Pérez

Las Palmas de Gran Canaria  
Junio 2023

# Índice

Índice de Figuras .....	4
Índice de Tablas.....	5
Resumen.....	6
Abstract .....	7
Capítulo 1. Introducción .....	8
1.1. Motivaciones.....	9
1.2. Objetivos .....	10
1.3. Organización del documento .....	10
Capítulo 2. Estado del arte .....	12
2.1. Aplicaciones de aprendizaje automático en medicina.....	12
2.2. Estado del arte de los clasificadores de tumores cerebrales.....	13
Capítulo 3. Planificación y desarrollo del proyecto .....	16
3.1. Metodología de trabajo y planificación inicial .....	16
3.2. Explicación del desarrollo del proyecto .....	17
Capítulo 4. Estudio de los tumores cerebrales.....	21
4.1. Definiciones de tumor y cáncer .....	21
4.2. Tipos de tumores cerebrales.....	21
4.3. Métodos de Diagnóstico .....	22
4.3.1. Tomografía Axial Computarizada .....	23
4.3.2. Resonancia Magnética .....	23
Capítulo 5. Herramientas.....	26
5.1. Lenguaje de programación.....	26
5.2. Entorno de pruebas y servicio de almacenamiento.....	26
5.3. Librerías.....	27
Capítulo 6. Conjuntos de datos .....	28
6.1. Conjunto de datos de tumores cerebrales de Figshare .....	28
6.2. Conjunto de datos de tumores cerebrales de Kaggle .....	29
6.3. Combinación de los conjuntos de datos .....	31
Capítulo 7. Redes neuronales.....	33
7.1. <i>Transformers</i> .....	34
7.1.1. <i>Vision Transformer</i> .....	35

7.1.2.	<i>Data-Efficient Image Transformer</i> .....	36
7.1.3.	<i>Swin Transformer</i> .....	37
Capítulo 8.	Configuración de los experimentos.....	39
8.1.	Configuración de hiperparámetros .....	39
8.1.1.	Algoritmo de Optimización .....	40
8.1.2.	Funciones de activación y pérdida .....	40
8.1.3.	Comparativa de hiperparámetros.....	41
8.2.	Métricas empleadas.....	42
Capítulo 9.	Evaluación de los resultados obtenidos.....	44
9.1.	Precisión media de los modelos.....	44
9.1.1.	Resultados obtenidos por la arquitectura ViT.....	47
9.1.2.	Resultados obtenidos por la arquitectura DeiT .....	48
9.1.3.	Resultados obtenidos por la arquitectura Swin .....	49
9.1.4.	Conclusiones basadas en la exactitud y el error .....	50
9.2.	Evaluación del rendimiento por clases .....	51
9.3.	<i>Precision, Recall</i> y <i>F1 Score</i> de los modelos .....	54
9.4.	Comparación con el estado del arte .....	55
9.5.	Aportaciones científicas .....	58
Capítulo 10.	Conclusiones.....	60
Anexo:	competencias específicas.....	61
Bibliografía	.....	62

# Índice de Figuras

Figura 4-1: Imágenes MRI de distintos tipos de tumores cerebrales primarios .....	22
Figura 4-2: Tipos de imágenes MRI según el plano anatómico.....	24
Figura 4-3: Tipos de imágenes MRI según los valores de TE y TR .....	25
Figura 6-1: Muestras del conjunto de datos de Figshare.....	29
Figura 6-2: Muestras del conjunto de datos de Kaggle.....	30
Figura 7-1: Estructura de un <i>Transformer</i> .....	35
Figura 7-2: Estructura de ViT.....	36
Figura 7-3: Estructura de DeiT.....	37
Figura 7-4: Estructura de Swin .....	38
Figura 9-1: Gráficas de aprendizaje y error de ViT.....	48
Figura 9-2: Gráficas de aprendizaje y error de DeiT.....	49
Figura 9-3: Gráficas de aprendizaje y error de Swin .....	50
Figura 9-4: Matrices de confusión de DeiT224 .....	53
Figura 9-5: Matrices de confusión de SwinS224 .....	54

# Índice de Tablas

Tabla 3-1: Fases y tareas inicialmente definidas.....	17
Tabla 3-2: Fases y tareas desarrolladas.....	20
Tabla 6-1: Estructura del conjunto de datos de Kaggle .....	31
Tabla 6-2: Estructura del conjunto de datos elaborado.....	32
Tabla 7-1: características de los modelos empleados.....	34
Tabla 8-1: Configuraciones de hiperparámetros.....	42
Tabla 9-1: Precisión media sin aumento de datos .....	46
Tabla 9-2: Precisión media con aumento de datos .....	46
Tabla 9-3: Tasas de exactitud por clase.....	51
Tabla 9-4: Números de aciertos por clase.....	52
Tabla 9-5: Porcentajes medios de precisión, recuperación y <i>F1 score</i> .....	55
Tabla 9-6: Comparativa de los conjuntos de datos.....	57
Tabla 9-7: Comparación de resultados con respecto a otros autores .....	58

# Resumen

El proyecto, denominado “Clasificación automática de tumores cerebrales mediante el uso de redes neuronales basadas en Vision Transformer”, tiene como principal objetivo el desarrollo y estudio de un clasificador de tumores cerebrales basado en la topología *Transformer*, que sea capaz de identificar distintos tipos de tumores cerebrales y de determinar si un cerebro no presenta masa cancerígena.

Se quiere desarrollar este sistema, para facilitar y aumentar la velocidad de la detección de este tipo de tumores, ya que la rapidez en el diagnóstico es clave para incrementar las posibilidades de supervivencia del paciente.

En conclusión, el proyecto busca aportar una herramienta rápida y eficaz que ayude en la lucha contra el cáncer.

# Abstract

The Project, called “Clasificación automática de tumores cerebrales mediante el uso de redes neuronales basadas en Vision Transformer”, aims to develop and study a brain tumor classifier using Transformer neural network, it is capable of identifying different types of brain tumors and determining whether a brain not has a cancerous mass.

This system is being developed to facilitate and increase the speed of detection of this type of tumor because the tempo in diagnosis is essential to increasing the patient’s chances of survival.

In conclusion, the project aims to provide a tool to help in the fight against a disease with as high a mortality rate as cancer.

# Capítulo 1. Introducción

Pocas enfermedades han dado tanto que hablar como los tumores, su complejidad y su alta tasa de mortalidad han hecho que sean una de las afecciones más conocidas por la población. Incluso hoy en día, con todos los avances que ha realizado el ser humano en el sector sanitario, siguen causando millones de muertes en todo el mundo. Sin embargo, la tasa de mortalidad ha ido disminuyendo gracias a los diversos tratamientos que han ido surgiendo con el transcurso de los años. Para que estos sean efectivos, la velocidad en el diagnóstico es fundamental y comenzar la aplicación del plan de tratamientos en una etapa temprana de la enfermedad hace que la probabilidad de recuperación del paciente aumente significativamente. Sobre todo, la rapidez y la precisión en el diagnóstico es clave en las variantes más mortales, como es en el caso de los tumores cerebrales.

La imagen es uno de los medios más utilizados por el equipo médico para la realización de diagnósticos. Desde el uso de radiografías para visualizar la gravedad y localización de una fractura de un hueso, hasta el empleo de ecografías para el seguimiento de un feto durante el periodo de embarazo. En el caso de los tumores cerebrales, se suelen utilizar imágenes generadas por resonancia magnética o *Magnetic Resonance Imaging* (MRI) debido a su alto índice de detalle. De esta forma, se puede determinar el tipo de tumor, sus dimensiones y su localización.

Según [1], la visión por computador o visión artificial es uno de los principales campos de la inteligencia artificial. Esta permite a los sistemas, a través del uso de diversas técnicas y herramientas, extraer, procesar e interpretar imágenes digitales o vídeos. La Visión por computador intenta simular la capacidad que tienen los seres humanos de analizar y comprender la información visual que captan del mundo exterior, lo que les permite realizar tareas de gran complejidad como la detección de patrones, objetos, entre otras. En estos últimos años, se han empezado a implantar este tipo de técnicas de forma masiva, ya que son capaces de extraer información relevante de datos visuales de entrada de forma automática y, con una gran rapidez y precisión. Durante una gran cantidad de tiempo, el paradigma de los sistemas de Visión por computador ha estado dominado por las redes neuronales convolucionales o *Convolutional Neural Network* (CNN). Sin embargo, en estos últimos años ha empezado a emerger una nueva arquitectura denominada *Transformer*. Aunque, las redes de tipo *Transformer* originalmente se desarrollaron con la intencionalidad de resolver tareas de procesamiento del lenguaje natural, diversos proyectos han demostrado el gran rendimiento que pueden ofrecer esta clase de modelos para la resolución de problemas del campo de la visión artificial.

Una de las principales tareas de la visión artificial es la clasificación de imágenes. Las máquinas desarrolladas para la resolución de este tipo de problemas son capaces de etiquetar y reconocer un objeto o una determinada característica relevante presente en una imagen de entrada. En la medicina, el amplio uso de elementos visuales como tomografías o resonancias magnéticas para la detección de enfermedades, como los tumores cerebrales, y la imperiosa necesidad de realizar diagnósticos rápidos y precisos, hacen de esta rama de la ciencia de la salud un sector con un gran potencial aplicativo para este tipo de herramientas.

Como se ha comentado anteriormente, la velocidad en el diagnóstico de los tumores cerebrales es fundamental para el aumento de las posibilidades de supervivencia del enfermo. Diversos factores, como el estrés o la enorme carga de trabajo a los que suelen estar sometidos los sanitarios, pueden provocar errores fatales que conlleven a falsos negativos. Este proyecto intenta proporcionar una herramienta rápida y eficaz para que, a través del uso de *Transformers*, el proceso de diagnóstico se automatice y de esta forma se convierta en una tarea más sencilla.

En este proyecto, se exponen una serie de *Transformers* que tienen la capacidad de detectar un total de tres tipos diferentes de tumores cerebrales primarios, los cuales son los siguientes: meningiomas, gliomas y tumores pituitarios. Asimismo, los modelos son capaces de distinguir cuando un paciente no presenta ninguna clase de tumor cerebral. Para el entrenamiento de las redes, se ha empleado un conjunto de datos (*dataset*) elaborado a partir de dos colecciones de imágenes MRI, uno proveniente de Kaggle y otro de Figshare, que muestran la estructura cerebral de diversos pacientes. La colección resultante contiene las siguientes clases: meningiomas, gliomas, tumores pituitarios y cerebros sanos. Aunque las imágenes del conjunto son de 512x512, el tamaño utilizado para el entrenamiento de los modelos ha sido de 224x224.

Se han empleado una serie de modelos de las arquitecturas *Vision Transformer (ViT)*, *Data-Efficient Image Transformer (DeiT)* y *Shifted window Transformer (Swin)* previamente entrenados para intentar resolver el problema de clasificación planteado. El entrenamiento de las redes se ha efectuado haciendo uso de algunas técnicas, como transferencia de aprendizaje o aumento de datos. Una vez completado el proceso de aprendizaje de cada uno de los modelos, se ha llevado a cabo una evaluación de los resultados obtenidos a través de una serie de métricas y representaciones visuales. Para el análisis del rendimiento de las redes se han utilizado las siguientes métricas: *accuracy*, *precision*, *recall* y *F1 Score*. Con respecto a estas dos primeras medidas, existe un pequeño problema a nivel lingüístico, ya que el lenguaje español no recoge dos términos, como sí lo hace el inglés, para diferenciar ambos conceptos. Por lo que, el autor ha decidido usar la palabra “precisión” como traducción directa de *accuracy* y de *precision*. Por último, se ha hecho una comparativa entre los resultados obtenidos y los alcanzados por investigaciones anteriores.

## 1.1. Motivaciones

Se calcula que, de manera global, durante el año 2020 aproximadamente 308.102 personas fueron diagnosticadas con algún tipo de tumor cerebral primario o tumor de la médula espinal y 251.329 personas fallecieron a causa de estos motivos [2].

La velocidad y la precisión del diagnóstico son factores fundamentales para la reducción de la tasa de mortalidad de los tumores cerebrales. La aplicación de los diversos tratamientos en etapas tempranas de la enfermedad hace que las posibilidades de supervivencia aumenten considerablemente. Sin embargo, la detección de tumores cerebrales conlleva una gran cantidad de dificultades: la complejidad de la enfermedad, el gran abanico de afecciones que presentan síntomas parecidos y el requerimiento de conocimientos profundos en el campo para el correcto uso de las diversas técnicas utilizadas en el diagnóstico, entre otras.

Este proyecto no solo intenta proporcionar una herramienta eficaz, veloz y precisa para la realización de diagnósticos de tumores cerebrales, sino que también busca automatizar el proceso de detección de la enfermedad con el objetivo de aliviar la carga de trabajo del cuerpo médico y facilitar dicho proceso para aquellos profesionales con menor nivel de experiencia.

Asimismo, este trabajo procura ser un punto de apoyo para futuros proyectos de investigación de la misma índole que intenten emplear redes *Transformer* u otras técnicas de aprendizaje profundo (*Deep Learning*), mostrando el grado de viabilidad que tiene el uso de estas tecnologías dentro del sector sanitario, las ventajas que tienen con respecto a otras aplicaciones y los resultados obtenidos.

## 1.2. Objetivos

El objetivo principal es la de estudiar y comparar el rendimiento de diversos modelos ViTs, DeiTs y Swins a la hora de clasificar distintos tipos de tumores cerebrales primarios a partir de imágenes MRI. Estos también deben ser capaces de discernir aquellas muestras en las que no se presenta la enfermedad. Asimismo, se ha buscado maximizar los resultados de las arquitecturas mediante una búsqueda exhaustiva de los conjuntos de hiperparámetros y mecanismos de entrenamiento adecuados para cada una de las redes.

Se ha elaborado un conjunto de datos a partir de dos colecciones usualmente empleadas por otros autores en investigaciones de la misma índole. Todo ello con la meta de entrenar, validar y testear los modelos con el mayor número de secuencias MRI posible. El volumen contiene las siguientes clases: glioma, meningioma, tumor pituitario y no tumor.

## 1.3. Organización del documento

Para facilitar la lectura del lector, se ha realizado un breve resumen de cada uno de los capítulos que componen el documento:

- **Capítulo 2 – Estado del Arte:** sección en la que se realizará un barrido histórico de diversas aplicaciones de aprendizaje automático (*Machine Learning*) desarrolladas con fines médicos. Asimismo, se comentará la evolución sufrida por el paradigma de la clasificación de imágenes y la aparición de diversos proyectos que han aprovechado estas metodologías para la detección de tumores cerebrales primarios a partir de secuencias MRI.
- **Capítulo 3 – Planificación y desarrollo del proyecto:** se comentará el marco de trabajo utilizado, cada una de las fases que componen el proyecto y las tareas realizadas.
- **Capítulo 4 – Estudio de los tumores cerebrales:** apartado en el que se explicará terminología básica en lo que respecta a tumores cerebrales y los métodos de diagnóstico más usados.
- **Capítulo 5 – Herramientas:** se indicará y comentará el lenguaje de programación, las librerías y entornos de trabajo empleados.

- **Capítulo 6 – Conjuntos de imágenes:** se abordarán las diferentes fuentes de las que se han tomado las imágenes MRI y la estructura del conjunto de datos resultante de la combinación de estas colecciones.
- **Capítulo 7 – Redes neuronales:** sección en la que se explicarán las redes de tipo *Transformers* entrenadas, así como su funcionamiento.
- **Capítulo 8 – Configuración de los experimentos:** apartado en el que se comentará la metodología de entrenamiento usada y las métricas empleadas para la evaluación de los resultados obtenidos.
- **Capítulo 9 – Evaluación de los resultados obtenidos:** capítulo en el que se evaluarán los niveles de desempeño obtenidos por los modelos durante la clasificación, se realizará una comparativa de los resultados conseguidos con respecto al estado del arte y se comentarán las aportaciones a nivel científico que proporciona este proyecto.
- **Capítulo 10 – Conclusiones:** sección en la que el autor indicará toda la información de valor que puede ser extraída de este proyecto.
- **Anexo – Competencias Específicas:** se abordarán todas las competencias cubiertas por este trabajo.

# Capítulo 2. Estado del arte

Desde el surgimiento del aprendizaje automático, se han desarrollado una gran cantidad de aplicaciones basadas en este concepto para la resolución de tareas en diferentes áreas, como la industrial, la deportiva o la automovilística. Este tipo de aplicaciones han tenido también un gran éxito en el sector médico, ya que permiten a los sanitarios disponer de herramientas que les ayuden a procesar los datos recogidos de un paciente, como sus síntomas o elementos visuales tomadas de este, para poder determinar su condición médica. En esta sección, se comentarán diversas aplicaciones de aprendizaje automático desarrolladas con fines médicos. Además, se abordarán algunas técnicas empleadas por otros investigadores para la detección de tumores cerebrales por imágenes MRI y los resultados obtenidos por estos en términos de precisión (*accuracy*).

## 2.1. Aplicaciones de aprendizaje automático en medicina

El diagnóstico de enfermedades constituye uno de los procesos más críticos e importantes llevados a cabo por los cuerpos sanitarios. Un buen diagnóstico es fundamental para poder comprender y determinar la condición médica de un determinado paciente. Gracias a la información extraída durante este proceso, los especialistas pueden establecer una serie de tratamientos que buscan eliminar o mitigar los efectos de la enfermedad. Por lo tanto, es esencial que la información obtenida sea precisa. Asimismo, la velocidad en el diagnóstico es de suma importancia, ya que la detección en una fase temprana de una cierta afección permite aplicar en el paciente un plan de tratamiento que impida la evolución de la patología. En ciertas enfermedades, como los tumores, el hecho de empezar a aplicar dicha estrategia de tratamiento en una primera etapa de la enfermedad provoca un aumento significativo de su efectividad, lo que incrementa las posibilidades de supervivencia del afectado. La creación de herramientas para la automatización del diagnóstico ha sido una meta ampliamente perseguida en las últimas décadas, ya que permiten reducir considerablemente la carga de trabajo del equipo sanitario e incluso pueden llegar a ser más precisas que los propios especialistas.

El desarrollo de esta clase de sistemas no es realmente una novedad surgida en estos últimos años. La primera aplicación creada para la automatización de diagnósticos fue desarrollada por Edward Shortliffe en su tesis doctoral a principios de los años 70, denominada Mycin [3]. Según [4], Mycin era un sistema experto basado en reglas de causa y efecto, capaz de detectar las bacterias causantes de una determinada patología y de recomendar el tratamiento más adecuado que debería seguir el paciente. Su porcentaje de acierto era de casi un 70 %, mayor que la de los médicos no especialistas en la materia. Sin embargo, el sistema no se usó de forma masiva debido a los problemas legales que podrían enfrentar los hospitales si Mycin realizase algún diagnóstico incorrecto.

A mitad de la década de los 80s, el Hospital General de Massachusetts lanzó al mercado el sistema de apoyo de decisiones clínicas DXplain [5]. Dxplain era una aplicación que generaba una lista de posibles diagnósticos a partir de los síntomas sufridos por el paciente [6].

El surgimiento del aprendizaje profundo causó una gran revolución en el mundo de la inteligencia artificial. En 2007, IBM desarrolló un supercomputador capaz de responder preguntas formuladas en lenguaje natural denominado Watson [5] [7]. Watson estaba basado en DeepQA, un software que haciendo uso de procesamiento de lenguaje natural y otras herramientas, era capaz de generar rápidamente posibles respuestas a una determinada cuestión planteada [5]. Esta tecnología empezó a utilizarse en el año 2013 por el *Memorial Sloan Kettering Cancer Center* como un sistema de soporte médico capaz de recomendar diferentes opciones de tratamiento para pacientes con cáncer de pulmón [8]. En el año 2015, se desarrolló Pharmabot que consistía en un *chatbot* conversacional cuya finalidad era la de prescribir medicamentos genéricos para niños y ofrecer información relacionada con estos [5] [9]. A principios de 2017, la FDA o *U.S. Food and Drug Administration* aprobó por primera vez el uso de una IA denominada Arterys, la cual fue desarrollada para ayudar a los médicos en el diagnóstico de enfermedades cardíacas [10].

Durante estos últimos años, ha habido un creciente aumento en el desarrollo de aplicaciones basadas en aprendizaje profundo. El equipo de investigadores de MIT lanzó un comunicado en 2019, en donde explicaban los resultados obtenidos por un modelo que era capaz de predecir el cáncer de mama con 5 años de antelación de su desarrollo [11]. En septiembre del año 2020, un equipo de investigadores del *IEEE Open Journal of Engineering in Medicine and Biology* elaboró un sistema capaz de diagnosticar casos de Covid-19 mediante la voz de los pacientes con una precisión de más del 98 % [12].

## 2.2. Estado del arte de los clasificadores de tumores cerebrales

Antes de la aparición de las redes CNN, el paradigma del aprendizaje automático estaba dominado por clasificadores tradicionales como SVM o KNN. Diversos estudios, como el de Kavin Kumar et al. [13], han utilizado este tipo de métodos junto a técnicas de extracción de características para la clasificación de tumores cerebrales en imágenes generadas por resonancia magnética. Los modelos fueron entrenados con un conjunto de imágenes MRI pequeño, que contenían secuencias de tumores cerebrales y de cerebros sanos. El clasificador SVM fue capaz de distinguir con un 95 % de precisión los cerebros que presentaban un tumor con respecto a los que no. Mientras que, el clasificador KNN tuvo un rendimiento menor llegando a un máximo de 80 % de precisión.

A partir del año 2012, con el surgimiento del modelo AlexNet, han surgido diversas redes neuronales convolucionales que han dominado el panorama de la clasificación de imágenes, como VGGNet, Resnet o GoogleLeNet [14].

Özlem Polat et al. [15], en su paper denominado *Classification of brain tumors from MR images using deep transfer learning*, efectuaron una comparativa del rendimiento que tenían algunas redes famosas como VGG16, VGG19, ResNet50 y DenseNet21, a la hora de clasificar distintos tipos de tumores cerebrales (glioma, meningioma y tumor pituitario). Las CNNs fueron entrenadas mediante transferencia de aprendizaje y se optó por utilizar un conjunto de datos proveniente de la plataforma Figshare, con más de 3 mil imágenes de resonancias de cerebros. El modelo que obtuvo un mayor rendimiento fue ResNet50 con una precisión del 99.02 %.

Marco Antonio Gómez Guzmán et al. [16] realizaron una comparativa similar, pero con las redes ResNet50, InceptionV3, InceptionResNetV2, Xception, MobileNetV2 y EfficientNetB0. Los modelos eran capaces de clasificar hasta 3 tipos de tumores cerebrales primarios y podían percibir cuando un cerebro no presentaba la enfermedad. Se usó *Transfer Learning* para el entrenamiento de las redes. InceptionV3 logró el mejor resultado con un 97.12 % de exactitud media.

Algunos artículos como [17] propusieron sus propias redes CNN para la detección de tumores cerebrales en secuencias MRI, en vez de emplear arquitecturas ya diseñadas. En este caso, los autores lograron producir un modelo capaz de reconocer distintas clases de tumores cerebrales primarios, además de poder identificar aquellos cerebros que no contenían la enfermedad, con una exactitud del 99 %. Ali Mohammad Alqudah et al. [18] desarrollaron una red convolucional personalizada que alcanzó una alta tasa de exactitud con un 99.19 %.

M.Sharma et al. [19] propusieron un modelo híbrido entre SVM y CNN con una exactitud del 98.50 % a la hora de clasificar tumores cerebrales benignos y malignos en imágenes MRI. Para la extracción de las características de las muestras, se empleaba el modelo CNN. Mientras que, la clasificación la realizaba una SVM en base a esta información.

En el año 2020, la publicación de Alexey Dosvitsky et al. [20] causó una gran revolución en el sector, ya que demostró que los *Transformers* puros no solo podían ser empleados para la realización de tareas NLP, sino que también para la resolución de problemas de clasificación de imágenes. El modelo en cuestión fue denominado por los autores como *Vision Transformer*.

Sudhakar Tummala et al. [21] llevaron a cabo un estudio en el que se comprobaba el rendimiento que podían alcanzar los modelos ViTB16, ViTB32, ViTL16 y ViTL32 al momento de categorizar distintos tipos de tumores cerebrales en secuencias MRI. El entrenamiento se realizó con diferentes resoluciones de imágenes (224x224 y 384x384) y se emplearon técnicas de transferencia de aprendizaje. Los investigadores observaron que el modelo individual con mejor rendimiento fue ViTL32 con un 98.21 % de precisión, el cual fue entrenado con imágenes de 384x384. Sin embargo, el modelo que mejores resultados proporcionó fue una combinación de estos 4 *Transformers*. Este fue entrenado con imágenes de 384x384 y alcanzó un 98.7 % de precisión.

Los autores de [22] desarrollaron LCDeiT, una variante del modelo ViT que mediante una estrategia estudiante-profesor intenta reducir el número de datos necesarios para el entrenamiento. La red fue entrenada con dos conjuntos de datos, uno proveniente de Figshare denominado *Brain Tumor Dataset* y otro llamado BraTS2021, el cual está más enfocado para tareas de segmentación. Ambos volúmenes contenían imágenes de secuencias en las que aparecían 3 tipos de tumores cerebrales primarios: meningioma, glioma y tumor pituitario. El modelo demostró un gran rendimiento identificando los distintos tumores. Tanto con el conjunto de Figshare, logrando un 98.73 % de exactitud, como con BraTS2021, alcanzando un 97.86 % de precisión.

D. Reyes et al. [23] entrenaron una serie de arquitecturas CNN mediante transferencia de aprendizaje y aumento de datos. Los autores hicieron uso de dos conjuntos de datos para la realización del entrenamiento, uno proveniente de Kaggle denominado *Brain Tumor Classification (MRI)* y otro de la plataforma Figshare. Los modelos EfficientNet, ResNet y

MobileNet fueron los que presentaron un mejor comportamiento durante la clasificación de las muestras de las dos colecciones.

J. Sánchez et al. [24] entrenaron diversos modelos de la arquitectura EfficientNet, haciendo uso de mecanismos de transferencia de aprendizaje y *fine-tuning*. Los dos conjuntos de datos seleccionados para la ejecución de los procesos de aprendizaje fueron *Brain Tumor Classification (MRI)* de Kaggle y *Brain Tumor Dataset* de Figshare. Ambas colecciones disponen de más de 3000 muestras cada una. El modelo EfficientNetB0 entrenado mediante el conjunto de Figshare fue el más preciso con un 98.4 %.

# Capítulo 3. Planificación y desarrollo del proyecto

Dos factores fundamentales para el correcto desarrollo de cualquier proyecto son la realización de una buena planificación inicial que abarque las distintas fases en las que se estructurará este y la selección de una correcta metodología de trabajo que permita readaptar la planificación inicial. En este capítulo, se procede a explicar el marco de trabajo escogido, la planificación inicial redactada y, las diferentes fases y tareas efectuadas durante el desarrollo del proyecto.

## 3.1. Metodología de trabajo y planificación inicial

Para el desarrollo del proyecto se ha empleado la metodología Scrum. Como bien se menciona en [25] [26], Scrum es un marco de trabajo ágil, ampliamente usado en el desarrollo de proyectos en el sector tecnológico por sus buenos resultados. Se basa en la realización de iteraciones (*Sprints*) de corta duración, de 2 a 4 semanas, en las que se ejecutan una serie de tareas de diferentes prioridades, con la finalidad de obtener un resultado final denominado incremento. Antes del comienzo de cada iteración, el equipo verifica si es necesario añadir nuevos requisitos para ser implementadas en futuros *Sprints* y se llega a un consenso con respecto a la selección de las tareas a efectuar en la siguiente iteración. Dicha selección, se suele realizar en base al nivel de prioridad de cada tarea. Durante el desarrollo de la iteración, el equipo de desarrollo se reúne diariamente para conocer el estado en el que se encuentra cada una de las tareas. Estas reuniones suelen ser de una corta duración, entre 10 a 15 minutos como máximo. Una vez finalizado el *Sprint*, se planifica una reunión en el que se presentan los resultados obtenidos y, según estos, se adaptan los objetivos para futuras iteraciones. Asimismo, el equipo lleva a cabo un análisis profundo de la metodología empleada durante el *Sprint* para identificar posibles mejoras que permitan optimizar y mejorar el proceso. La metodología define dentro de los equipos de trabajo hasta un total de 3 roles:

- *Product Owner*: define los requisitos necesarios que tiene que cumplir el producto a desarrollar, intentando aumentar y maximizar el valor de este.
- *Scrum Master*: es el encargo de asegurar que la metodología se está llevando correctamente e intenta eliminar posibles obstáculos o impedimentos que dificulten la realización de los incrementos.
- Equipo de desarrollo: son los encargados de desarrollar los incrementos.

En el caso de este proyecto, el equipo Scrum está compuesto por dos integrantes: el tutor y el estudiante. El tutor ha desempeñado los roles de *Product Owner* y *Scrum Master*, mientras que el alumno ha cumplido el rol del Equipo de desarrollo. Cabe destacar que previamente al comienzo de la primera iteración, el equipo Scrum mantuvo una reunión en la que se definió las distintas fases con las que iba a contar el proyecto a realizar y una serie de tareas genéricas que deberían ser llevadas a cabo. Estas fases y tareas se encuentran redactadas en la Tabla 3-1.

Fases	Tareas (nombre y descripción)
Estudio previo y Análisis	Tarea 1.1: Investigación y aprendizaje sobre conceptos de <i>Machine Learning</i> y de las librerías necesarias
	Tarea 1.2: Estudio sobre los conceptos médicos abarcados en el problema en cuestión
Diseño, Desarrollo e Implementación	Tarea 2.1: Desarrollo de los diferentes modelos de redes neuronales
Evaluación, Validación y Prueba	Tarea 3.1: Chequeo de los resultados obtenidos por los modelos
	Tarea 3.2: Evaluación de los resultados obtenidos y elección del modelo final
Documentación y Presentación	Tarea 4.1: Desarrollo del documento que se usará en la exposición
	Tarea 4.2: Práctica de la exposición
	Tarea 4.3: Realización de la exposición

Tabla 3-1: Fases y tareas inicialmente definidas.

Para la realización del proyecto, se emplearon iteraciones de una duración de 3 semanas cada una. Previamente al comienzo de cada *Sprint*, el equipo Scrum llevaba a cabo una reunión en la que se definían nuevas tareas a efectuar, se indicaba su grado de prioridad y se redefinían las ya existentes. Además, se seleccionaban las tareas a llevar a cabo en el siguiente *Sprint*. En cada una de las iteraciones, se podían realizar tareas de distintas fases si el equipo Scrum lo consideraba oportuno. Diariamente, el alumno dedicaba un corto periodo de tiempo para verificar el estado de las actividades y cuales le quedaban por realizar. Tras finalizar el periodo de 3 semanas, se planificaba una reunión entre el alumno y el tutor para observar, verificar y evaluar los resultados obtenidos del *sprint*. Durante este encuentro, se decidía según lo ocurrido en la iteración, si era necesario modificar o replanificar los objetivos marcados para futuros *sprints*. Asimismo, se reflexionaba sobre posibles cambios en la forma de trabajar del alumno, que pudiesen mejorar y maximizar el rendimiento de este.

### 3.2. Explicación del desarrollo del proyecto

En la Tabla 3-2, se pueden visualizar las actividades llevadas a cabo durante el desarrollo del trabajo. A continuación, se procede a explicar con mayor nivel de detalle las tareas realizadas en cada una de las fases:

- **Estudio previo y Análisis:** en primer lugar, se realizó un estudio profundo de distintos conceptos de aprendizaje automático y de aprendizaje profundo relacionados con el entrenamiento de redes neuronales. Esta primera tarea fue fundamental para poder entender cómo se lleva a cabo este, distintas metodologías, los elementos que lo conforman (hiperparámetros) y posibles problemas que pueden surgir durante el proceso de aprendizaje. Además, se buscaron todas las herramientas necesarias (lenguaje de programación, entorno de desarrollo, etc.) para el desarrollo del proyecto. Una vez asimilado estos conceptos, se efectuó una investigación exhaustiva con respecto a diversas aplicaciones de aprendizaje automático desarrolladas con fines médicos y sobre el estado del arte de los clasificadores de tumores cerebrales. Posteriormente, se llevó a cabo una exploración dentro el paradigma actual de las redes de tipo *Transformer*, con el fin de determinar aquellos modelos que podrían ser de mayor interés para ser entrenados. Las arquitecturas *Transformer* seleccionadas fueron las siguientes: *Vision Transformer*, *Data-Efficient Image Transformers* y *Swin Transformer*. La topología ViT fue escogida debido a ser la precursora en lo que respecta al uso de *Transformers* en el campo de la visión artificial. Otros de los motivos para su selección, fue provocado por los buenos resultados que obtuvieron este tipo de modelos en artículos anteriores relacionados con el desarrollo de clasificadores de tumores cerebrales. DeiT y Swin fueron seleccionados, dado que son topologías basadas en ViT que incorporan ciertas mejoras con respecto a su predecesor. Por lo que, el analizar el impacto de estas mejoras podría enriquecer de forma significativa el valor del proyecto. Para poder realizar el entrenamiento de las redes, se tuvieron que buscar una serie de librerías que permitiesen la construcción de estos *Transformers*. Debido a la falta de recursos computacionales, se optó por entrenar los 3 modelos más pequeños de cada una de las arquitecturas presentes en las librerías. Por último, se realizó un estudio relacionado con los tumores cerebrales y los métodos más comunes empleados para su diagnóstico.
- **Diseño, Desarrollo e Implementación:** gracias a la información extraída durante la investigación del estado del arte, se pudo observar como la mayoría de los autores empleaban dos conjuntos de datos en específico. El primero denominado *Brain Tumor Classification (MRI)* proveniente de Kaggle y el segundo llamado *Brain Tumor Dataset* originario de Figshare. Por lo que, se optó por combinar ambos conjuntos de datos para tener, de esta forma, una gran variedad de imágenes para el entrenamiento de los modelos. Para poder elaborar el conjunto de datos, se tuvo que llevar a cabo un tratamiento especial para el redimensionado de algunas imágenes de la colección de Kaggle. Una vez finalizado este, se procedió a estructurar y a organizar el conjunto de datos, el cual fue dividido en 3 lotes: entrenamiento, validación y testeo. Las dos últimas tareas realizadas en esta fase tuvieron que ver con el desarrollo del código para la realización del entrenamiento, la generación de representaciones visuales que permitan visualizar más fácilmente los resultados y la construcción de los *Transformers*.
- **Evaluación, validación y prueba:** para la ejecución de los entrenamientos se empleó siempre la misma metodología, explicada a continuación. En primer lugar, se efectuaban una serie de pruebas de ensayo y error para hallar la configuración de hiperparámetros

óptima para el entrenamiento de los modelos de una determinada arquitectura. Una vez encontrado este, se procedía a realizar el entrenamiento de estas redes utilizando exclusivamente transferencia de aprendizaje. Posteriormente, se ejecutaba una segunda ronda de entrenamientos en los que se usaba tanto transferencia de aprendizaje como aumento de datos. Tras finalizar el proceso de aprendizaje de todas las arquitecturas, se llevó a cabo un análisis de los resultados obtenidos por los modelos en base a diferentes métricas y se seleccionaron aquellos *Transformers* que alcanzaron mejores niveles de desempeño.

- Documentación: desarrollo de la memoria del proyecto, en donde se explican todas las fases explicadas con anterioridad.

Fases	Tareas (nombre y descripción)
Estudio previo y Análisis	Tarea 1.1: Investigación y aprendizaje sobre conceptos de aprendizaje automático y librerías necesarias
	Tarea 1.2: Realización del estado del arte
	Tarea 1.3: Estudio relacionado con el funcionamiento de los <i>Transformers</i> y búsqueda de librerías que permitan su implementación
	Tarea 1.4: Estudio sobre los conceptos médicos abarcados en el problema en cuestión
Diseño, Desarrollo e Implementación	Tarea 2.1: Búsqueda de fuentes de datos
	Tarea 2.2: Tratamiento de las imágenes
	Tarea 2.3: Diseño y elaboración del conjunto de datos
	Tarea 2.4: Desarrollo del código para el entrenamiento y visualización de resultados
	Tarea 2.5: Implementación de los modelos pre-entrenados
Evaluación, Validación y Prueba	Tarea 3.1: Búsqueda de los conjuntos de hiperparámetros óptimos para los ViTs
	Tarea 3.2: Entrenamiento de los ViTs mediante transferencia de aprendizaje
	Tarea 3.2: Entrenamiento de los ViTs mediante transferencia de aprendizaje y aumento de datos

	Tarea 3.3: Búsqueda de los conjuntos de hiperparámetros óptimos para los DeiTs
	Tarea 3.4: Entrenamiento de los DeiTs mediante transferencia de aprendizaje
	Tarea 3.5: Entrenamiento de los DeiTs mediante transferencia de aprendizaje y aumento de datos
	Tarea 3.6: Búsqueda de los conjuntos de hiperparámetros óptimos para los Swins
	Tarea 3.7: Entrenamiento de los Swins mediante transferencia de aprendizaje y aumento de datos
	Tarea 3.8: Entrenamiento de los Swins mediante transferencia de aprendizaje y aumento de datos
	Tarea 3.9: Evaluación de los resultados obtenidos y elección de las propuestas
Documentación	Tarea 4.1: Desarrollo de la memoria

*Tabla 3-2: Fases y tareas desarrolladas.*

# Capítulo 4. Estudio de los tumores cerebrales

En esta sección, se ha efectuado un estudio profundo de conceptos relacionados con los tumores cerebrales, con la finalidad de comprender cuestiones tan fundamentales como: qué son los tumores, qué tipos de tumores cerebrales existen y cuáles son los métodos de diagnóstico más empleados para su detección. Es de suma importancia el hecho de asimilar con cierto grado de profundidad las respuestas a estas cuestiones para poder comprender, de esta manera, el problema abarcado.

## 4.1. Definiciones de tumor y cáncer

La humanidad se ha visto afectada por el cáncer desde tiempos inmemorables. Papiros egipcios escritos sobre el año 1600 a.C. ofrecen las primeras descripciones de la enfermedad, cuyas fuentes se piensa que datan en torno al 2500 a.C. [27].

Según la Sociedad Española de Oncología Médica (SEOM) [28], el cáncer sigue siendo una de las afecciones con mayor tasa de mortalidad en el año 2022 y se espera un incremento del número de fallecimientos en los próximos años. La organización estima que para el año 2040 la enfermedad cause más de 16 millones de muertes a nivel mundial, lo que supondría un aumento de más de 6 millones de fallecidos con respecto a 2020.

El término cáncer hace referencia a una amplia gama de enfermedades que se caracterizan por el desarrollo de células anormales, que se propagan, expanden y crecen en cualquier lugar del organismo sin control alguno [29]. Mientras que, las células sanas se dividen de manera controlada y tras un determinado periodo de tiempo mueren, las células infectadas se fragmentan incontroladamente y pierden la capacidad de morir [29].

Un tumor consiste en una protuberancia o masa anormal de tejido corporal, generada por una multiplicación y división desmesurada de un conjunto de células [30]. Los tumores pueden presentar o no células cancerígenas. Los tumores que contienen células anormales se denominan malignos, mientras que los que no presentan masa cancerígena se denominan benignos [29].

## 4.2. Tipos de tumores cerebrales

Un tumor cerebral consiste en una protuberancia anormal cuya ubicación se encuentra dentro del cerebro [31]. Los tumores que se hayan originado en el propio cerebro se denominan primarios, mientras que los nombrados como secundarios o metastásicos son masas atípicas, cuyo nacimiento yace en otra parte del cuerpo, y se han expandido al cerebro [31].

Los tumores cerebrales primarios son categorizados según la región específica del cerebro en la que se ha desarrollado y su nivel de agresividad. La agresividad del tumor es clasificada en una escala de cuatro grados, siendo considerados los grados más altos como los más mortales [32]. Para el entrenamiento de los modelos, se han recopilado una serie de imágenes generadas por resonancia magnética que muestran tumores localizados en distintas partes del cerebro, con el objetivo de estos que sean capaces de distinguir distintos tipos de tumores cerebrales primarios en base a su ubicación. El conjunto de datos está compuesto por los siguientes tipos de tumores cerebrales primarios:

- Glioma: tipo de tumor cerebral cuyo origen aún se desconoce, aunque se especula que su nacimiento se origina debido al crecimiento excesivo de las células gliales [33]. Las células gliales se encuentran dentro del sistema nervioso y realizan funciones principalmente de soporte, que son fundamentales para el correcto funcionamiento del cerebro [34].
- Meningioma: se trata de una clase de tumor mayormente no canceroso cuyo desarrollo comienza en las meninges [33]. Las meninges consisten en una serie de membranas cuya función principal es la protección del cerebro y la médula espinal [35].
- Tumor pituitario: como bien se menciona en [36], un tumor pituitario es una protuberancia anormal, comúnmente benigna, que se origina en la glándula pituitaria. La glándula pituitaria se trata de un órgano esférico de tamaño minúsculo cuya ubicación se encuentra en la base del cerebro. Los tumores pituitarios suelen provocar un aumento o reducción de las hormonas segregadas por la glándula, que controlan ciertas funciones corporales importantes.

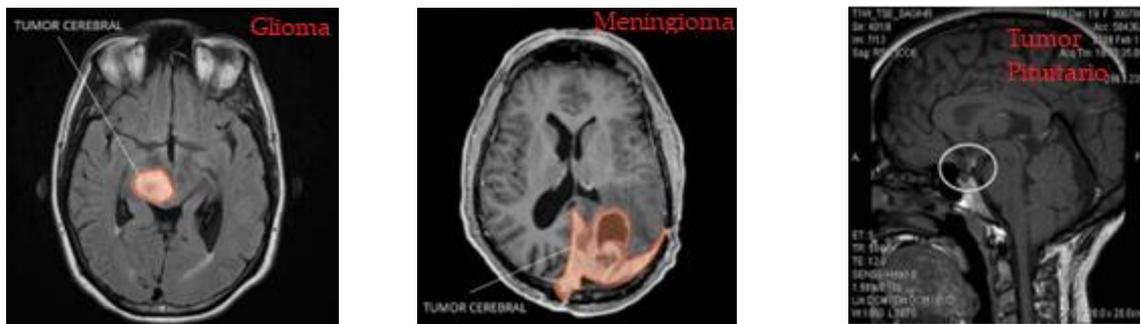


Figura 4-1: Imágenes MRI de distintos tipos de tumores cerebrales primarios.

Fuentes originales: American Society of Clinical Enconlogy [37], DEI [38].

### 4.3. Métodos de Diagnóstico

El diagnóstico es un proceso clave en la identificación de tumores cerebrales. Un correcto diagnóstico muestra información de gran relevancia sobre las características del tumor, como por ejemplo su tamaño, ubicación o la velocidad de propagación. Los datos recopilados por las pruebas de diagnóstico son empleados por el equipo médico para la determinación y selección del tratamiento más adecuado que debe seguir el paciente [39]. Asimismo, características como la velocidad y precisión de las pruebas de diagnóstico son fundamentales para la identificación

del tumor en las primeras fases de desarrollo, ya que durante este periodo la eficacia a la hora de aplicar los tratamientos correspondientes aumenta significativamente con respecto a fases más avanzadas de la enfermedad.

Normalmente, el proceso de diagnóstico suele comenzar cuando el paciente comenta a su médico de cabecera que ha empezado a padecer ciertos síntomas relacionados con la enfermedad, como migrañas, convulsiones, pérdida parcial o completa de la visión, etc. [40]. Una vez que el médico de cabecera haya llevado a cabo una evaluación exhaustiva del enfermo, si este confirma sus sospechas, lo destinaría a un neurólogo para la realización de una serie de pruebas de diagnóstico [39]. El primer paso efectuado por el neurólogo es la ejecución de un examen neurológico completo, usado por el especialista para observar la posible región del cerebro en la que se ubicaría el tumor [40]. Posteriormente, se emplean una serie de pruebas radiológicas utilizadas para visualizar las características físicas del tumor, su ubicación y posibles áreas que esté afectando [41].

#### **4.3.1. Tomografía Axial Computarizada**

Una Tomografía axial computarizada (TAC) consiste en una máquina conectada a un dispositivo informático, que permite la generación de imágenes del cerebro en forma de cortes desde diferentes perspectivas, a través de la emisión de rayos X [41]. Según la fundación Dacer [39], este suele ser el primer método de diagnóstico mediante imagen utilizado por los neurólogos, debido a sus rápidos resultados que facilitan la tarea de identificar posibles anomalías albergadas en el cerebro del paciente. Asimismo, los especialistas usan habitualmente tomografías para la recopilación de información que les permita determinar la siguiente prueba a efectuar.

#### **4.3.2. Resonancia Magnética**

La Resonancia Magnética es un mecanismo utilizado para la obtención de imágenes precisas de la estructura interna del cerebro por cortes. Su funcionamiento se basa en el uso de ondas de radiofrecuencia, campos magnéticos generados por imanes de gran magnitud y un sistema informático encargado de la generación de la imagen [42] [43]. Las imágenes se crean a partir de la medición de la energía emitida y la duración del estímulo a la que están sometidos los tejidos que se quieren visualizar [44]. La máquina usada en este tipo de pruebas consiste en un tubo largo y estrecho, abierto por ambos lados, en el que en su interior se recuesta al paciente [40].

Es la prueba por excelencia empleada por los neurólogos para la detección de tumores debido a su gran precisión y al increíble nivel de detalle de los resultados. Para la obtención de mejores imágenes, el equipo médico puede utilizar técnicas más avanzadas de MRI, como la espectroscopia o la perfusión cerebral por resonancia magnética [40].

El resultado obtenido por una resonancia magnética depende tanto del plano anatómico como de los factores físicos usados para captar la imagen. Los planos anatómicos empleados

dentro de este tipo de pruebas afectan directamente al campo de coordenadas de la imagen y son los siguientes:

- Axial: plano horizontal al suelo que divide el cuerpo en dos mitades, una superior y una inferior [45].
- Coronal: plano vertical al suelo que divide al paciente en dos mitades, una superior y una anterior [45]
- Sagital: plano vertical al suelo usado para dividir el cuerpo en dos mitades, una izquierda y una derecha [46].

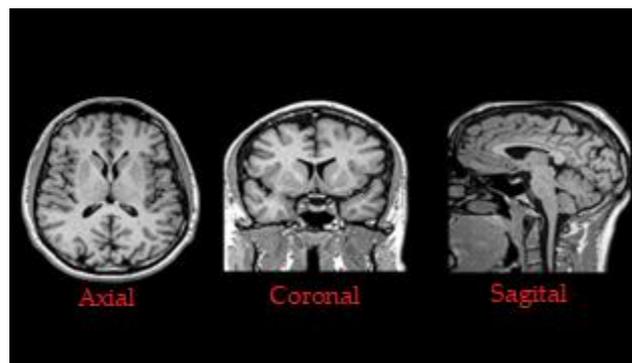


Figura 4-2: Tipos de imágenes MRI según el plano anatómico. Fuente original: Imaios [44].

Por otro lado, los factores físicos afectan directamente a la intensidad y al contraste presentado en la imagen [43]. Algunos de los factores que intervienen en la generación de imágenes MRI son los siguientes:

- *Time echo* (TE): hace referencia al tiempo transcurrido entre la ejecución del pulso de radiofrecuencia y la identificación de la señal de resonancia magnética, la cual es originada en los tejidos examinados [46].
- *Repetition Time* (TR): consiste en el periodo de tiempo transcurrido entre dos ondas de 90 grados [47].

Dependiendo del rango de valores de *Time echo* y *Repetition Time* empleados por la resonancia magnética, se pueden obtener diferentes tipos de representaciones visuales, siendo las más comunes las imágenes ponderadas en T1 y en T2 [48]. Una resonancia magnética genera imágenes ponderadas en T1 o *T1-Weighted* cuando emplea tiempos de TE y TR cortos [49]. Este tipo de imágenes permite al equipo médico diferenciar entre los tejidos grasos y los no grasos debido al fuerte contraste [46]. Si la resonancia magnética utiliza valores de TE y TR largos, esta conseguirá imágenes ponderadas en T2 o *T2-Weighted* [49]. Esta clase de secuencias facilita la identificación de los tejidos con una alta concentración de agua, como el líquido cefalorraquídeo, frente a los que contienen una menor cantidad de esta, como los huesos [46].

Asimismo, otra clase de imagen MRI muy usada es la denominada *Fluid Attenuated Inversion Recovery* (Flair). Para su creación se emplean valores muy elevados de *Time echo* y *Repetition Time*, mucho mayores que los utilizados para la creación de imágenes ponderadas en T1 y T2

[48]. Este tipo de imágenes posibilitan la visualización de posibles daños superficiales del cerebro mediante la supresión del líquido cefalorraquídeo [51].

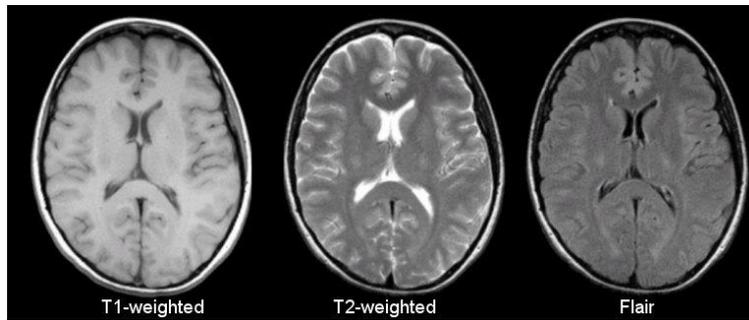


Figura 4-3: Tipos de imágenes MRI según los valores de TE y TR. Fuente original: Case [49].

# Capítulo 5. Herramientas

Para el desarrollo del proyecto, se han empleado una serie de herramientas que han permitido llevar a cabo este satisfactoriamente. A continuación, se procederán a explicar el lenguaje de programación, el entorno de pruebas, el servicio de almacenamiento y las librerías usadas para la realización de este.

## 5.1. Lenguaje de programación

Para la implementación, entrenamiento y visualización de los resultados obtenidos por cada uno de los modelos se ha utilizado el lenguaje de programación Python. Según la documentación del mismo lenguaje [52], Python es un lenguaje de código abierto que presenta una sintaxis sencilla para el usuario, la cual ocasiona que la curva de aprendizaje sea menor con respecto a otros lenguajes, y un competente sistema de programación de objetos. Python contiene un extenso repertorio de librerías que pueden ser empleadas en números ámbitos. Asimismo, el lenguaje presenta una gran versatilidad, ya que este puede ser ejecutado en diversos sistemas operativos [53].

En la actualidad, Python es uno de los lenguajes de programación más usados en el campo del aprendizaje automático, debido a la numerosa cantidad de librerías programadas expresamente para la resolución de tareas dentro de esta área. Dichas librerías permiten suavizar el nivel de complejidad presentado por ciertos conceptos matemáticos empleados en el desarrollo de aplicaciones de aprendizaje automático[54].

## 5.2. Entorno de pruebas y servicio de almacenamiento

El entorno de pruebas empleado para el desarrollo del proyecto ha sido Google Colab. La misma empresa desarrolladora [55], explica que Google Colab o Google Colaboraty consiste en un entorno de pruebas alojado en la nube basado en Jupyter Notebook, que permite a los usuarios programar, ejecutar y guardar código de manera arbitraria en Python a través del navegador. Colab es una herramienta muy utilizada para el desarrollo y entrenamiento de modelos de aprendizaje profundo, ya que permite acceder de forma gratuita a recursos informáticos, como unidades de procesamiento gráfico (GPUs). A través de este entorno, se han desarrollado una serie de cuadernos encargados de realizar las tareas necesarias para llevar a cabo el entrenamiento de los modelos de manera satisfactoria.

Para el almacenamiento del proyecto se ha usado Google Drive. Google Drive es un servicio gratuito de almacenamiento en la nube, que permite a los usuarios acceder a sus archivos de manera segura desde cualquier parte del mundo [56]. Drive ha permitido disponer al usuario de una gran cantidad de almacenamiento, que le ha sido necesaria para guardar todos los elementos relacionados con el proyecto, como los cuadernos, el conjunto de datos o los resultados obtenidos.

### 5.3. Librerías

TensorFlow es una librería desarrollada por el equipo de Google Brain que permite una ejecución eficiente de cálculos matemáticos a través del uso de recursos informáticos, como GPUs o unidades de procesamiento tensorial (TPUs) [57]. Esta herramienta ha sido utilizada para poder aprovechar la GPU ofrecida por Colab y, de esta forma, facilitar la realización de los entrenamientos. Asimismo, TensorFlow proporciona un *framework* para el desarrollo y entrenamiento de redes neuronales denominado Keras, el cual es muy usado debido a su sencillez y fácil manejabilidad para los usuarios [57]. Keras ha sido empleado para la construcción y entrenamiento de los modelos.

Numpy consiste en una librería de Python enfocada principalmente en el cálculo numérico y en el análisis de grandes conjuntos de datos, que permite la creación de una clase de objetos denominados arreglos (*arrays*) [58]. Los arreglos facilitan la representación y el procesamiento de una misma colección de datos, ya que la librería otorga la posibilidad de representar una misma colección de datos en dimensiones ilimitadas y su velocidad de procesamiento es mucha más alta que el de las listas [58]. Asimismo, la librería incorpora un conjunto de funciones que incrementan considerablemente la eficiencia en las operaciones realizadas entre diferentes arreglos [58].

La visualización de los resultados obtenidos por los diferentes modelos durante el proceso de aprendizaje se ha efectuado mediante las librerías Matplotlib y Scikit-Learn. Matplotlib es la librería de Python por excelencia para la visualización de datos a través de distintos tipos de gráficas, como histogramas, gráfica de líneas (*box plots*), etc. [59]. Scikit-learn es una librería de Python que contiene una gran cantidad de funciones y algoritmos especialmente diseñadas para el campo del análisis predictivo de datos, como clasificadores o modelos de selección [60].

Para la importación de los *Transformers* se han empleado las siguientes librerías:

- Vit-keras: consiste en un repositorio público subido a la plataforma Github que permite implementar alguno de los modelos descritos en el artículo *An Image Is Worth 16x16 Words: Transformers For Image Recognition At Scale* entrenados mediante el conjunto de datos ImageNet-2012 [61]. Los modelos ViT fueron descargados de esta librería.
- TensorFlow-hub: se trata de un repositorio de TensorFlow que incorpora una gran cantidad de modelos pre-entrenados y facilita la implementación de estos [62]. Mediante el uso de este repositorio, se cargaron algunos modelos DeiT que fueron previamente entrenados mediante el conjunto de datos ImageNet1k [64].
- Tfswin: librería de Github que proporciona un conjunto de modelos *Swin Transformers*, los cuales fueron entrenados mediante ImageNet1k [65].

Por último, cabe destacar la utilización de algunas librerías como Os, librería de Python que incorpora ciertas funcionalidades para la manipulación de archivos [66], o Import-ipynb, módulo usado para la importación del código de uno o varios cuadernos Jupyter [67].

# Capítulo 6. Conjuntos de datos

Una de las partes más importantes en todo proyecto de aprendizaje profundo es la selección del conjunto de datos para el entrenamiento de los modelos. Un buen volumen tiene que agrupar una gran cantidad de muestras distintas que deben reunir una amplia porción de patrones o características representativas de los elementos a clasificar. Todo ello con la finalidad de garantizar que el modelo, una vez haya finalizado el proceso de entrenamiento, sea capaz de reconocer esos rasgos en nuevos conjuntos de datos y, de esta forma, poder explotar todo su potencial rendimiento. Los conjuntos de datos poco generalistas pueden dar lugar a significativos problemas de sobreajuste en el proceso de aprendizaje, causando que la red sea incapaz de abordar eficazmente la tarea asignada.

Por este motivo, para el entrenamiento, evaluación y testeo de los *Transformers*, se ha decidido unificar 2 conjuntos de datos que agrupan imágenes MRI de distintos tipos de tumores cerebrales. A continuación, se procede a explicar toda la información de cierta relevancia relacionada con los volúmenes (origen, número de imágenes por clase, etc.) para, finalmente, comentar como está compuesto el conjunto de datos originado de la unificación de ambas colecciones. Dicho conjunto resultante ha sido el utilizado para llevar a cabo el entrenamiento de los modelos.

## 6.1. Conjunto de datos de tumores cerebrales de Figshare

Figshare consiste es una plataforma de libre acceso que proporciona a investigadores de todo el mundo un repositorio en donde puedan almacenar cualquier clase de información resultante de sus investigaciones, facilitando a otros investigadores su citación, acceso y utilización en sus propios proyectos [68] [69]. En esta plataforma se encuentra el conjunto *Brain Tumor Dataset* [70], una colección de imágenes de libre acceso y utilización publicado por Jun Cheng en el año 2017, que en la actualidad cuenta con más de 150 mil descargas y hasta un total de 129 citaciones en otros artículos científicos. El volumen de secuencias está compuesto por 3064 imágenes ponderadas en T1 de tres clases de tumores cerebrales primarios: glioma, meningioma y tumor pituitario, cuyas dimensiones son de 512x512. Las imágenes provienen de 233 pacientes diferentes y el número de secuencias por clase es el siguiente: 708 meningiomas, 1426 gliomas y 930 tumores pituitarios.

Los datos del conjunto de imágenes se encuentran estructurados en 4 carpetas que almacenan 766 archivos Matlab (*.mat*) cada una. Cada archivo guarda los siguientes atributos:

- *Label*: indica la clase a la que pertenece la imagen.
- *Image*: imagen del tumor.
- *PID*: identificador del paciente.
- *Tumor Border*: vector de coordenadas que indica la posición en la imagen del tumor.

Como se puede observar en la Figura 6-1, aunque todas las imágenes son ponderadas en T1, el plano anatómico puede diferir entre diferentes muestras (axial, sagital y coronal).

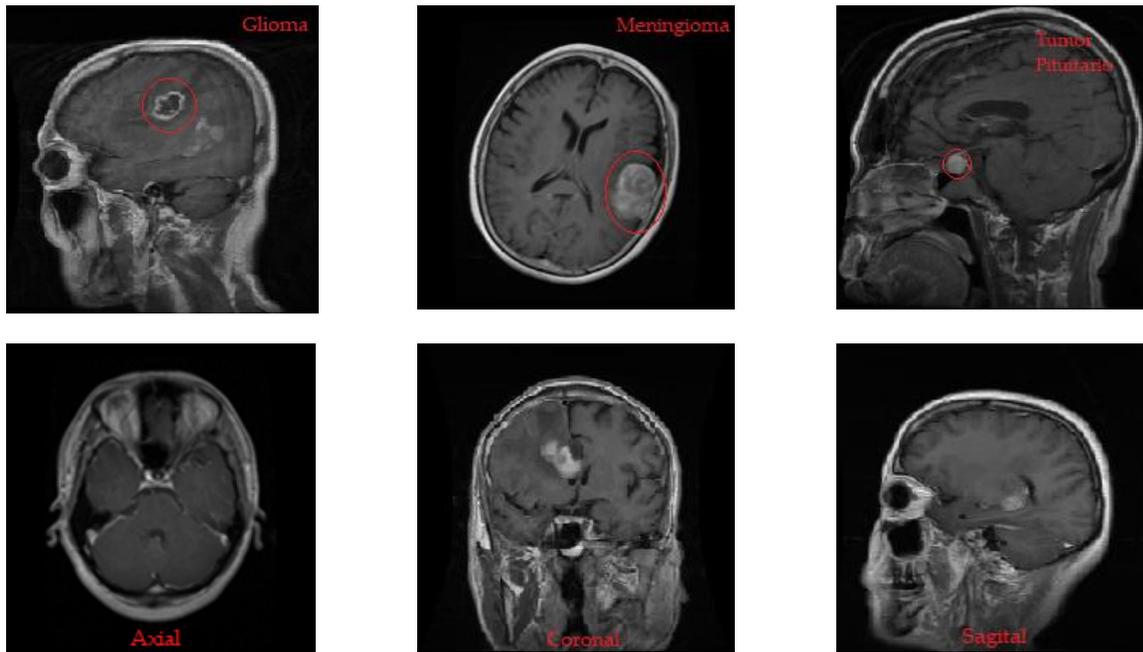


Figura 6-1: Muestras del conjunto de datos de Figshare.

## 6.2. Conjunto de datos de tumores cerebrales de Kaggle

Kaggle es una plataforma gratuita creada en el año 2010 y comprada en 2017 por Google, que cuenta con una de las mayores comunidades colaborativas del sector de la ciencia de datos y del aprendizaje automático [71]. Es una herramienta que permite a los usuarios compartir recursos, como conjuntos de datos o códigos, que puedan ser de ayuda para el desarrollo de otros proyectos similares [72]. En la plataforma existen también las denominadas competiciones, en donde especialistas de todo el mundo compiten para obtener la mejor solución a un problema concreto [73].

*Brain Tumor Classification (MRI)* [73] es un conjunto de datos almacenado en Kaggle que cuenta con más de 200 mil visualizaciones y 28 mil descargas, cuyos autores son Sartaj Bhuvaji, Ankita Kadam, Prajakta Bhumkar, Sameer Dedge, Navoneel Chakrabarty y Swati Kanchan. La colección fue elaborada con la finalidad de proporcionar un punto de partida para la creación de herramientas basadas en *Deep Learning*, que fuesen capaces de mejorar el rendimiento de los métodos manuales de detección de tumores cerebrales. Este contiene imágenes MRI de 3 tipos de tumores cerebrales: glioma, meningioma y tumor pituitario. Asimismo, el *dataset* cuenta con un cuarto tipo de secuencias en las que el cerebro del paciente no alberga ninguna clase de tumor. Las imágenes de la colección no solo difieren con respecto a los elementos presentados en cada una de estas cuatro clases, sino que también las imágenes de un mismo tipo varían dependiendo del plano anatómico (axial, coronal o sagital) y de los valores de TE y TR usados para la creación de cada una de las secuencias. El conjunto de datos almacena imágenes MRI de tipo *T1-Weighted*, *T2-Weighted* y *Flair*.

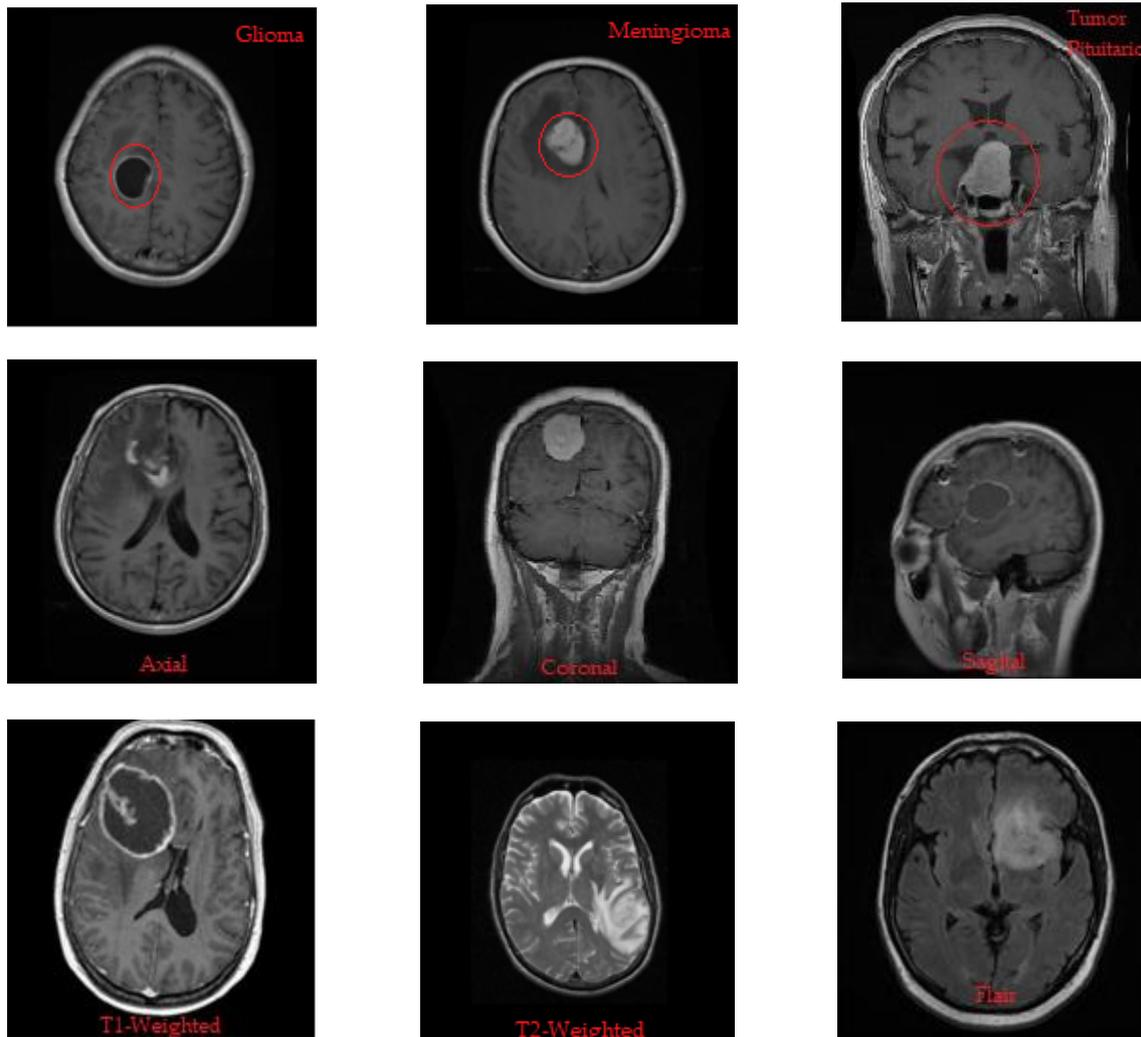


Figura 6-2: Muestras del conjunto de datos de Kaggle.

El conjunto de datos incluye 2 carpetas principales (*testing* y *training*) cuyo nombre hace referencia a la finalidad con la que se van a utilizar las imágenes que almacenan. Cada carpeta se encuentra dividida en 4 subcarpetas, una para cada clase, en donde se encuentran guardadas las secuencias. El conjunto de datos incluye un total de 3264 imágenes en formato JPG, las cuales fueron divididas por los autores originales de la siguiente forma:

- La carpeta de entrenamiento está compuesta por 2900 imágenes. Por clases, hay un total de 826 secuencias de gliomas, 822 de meningiomas, 395 de cerebros sanos y 827 de tumores pituitarios.
- Se almacenan 394 imágenes para el testeo final. Por clases, hay un total de 100 secuencias de gliomas, 115 de meningiomas, 105 de cerebros sanos y 74 de tumores pituitarios.

La mayoría de imágenes tienen un tamaño de 512x512. Sin embargo, existen algunas muestras que no siguen estas dimensiones.

Clases	Entrenamiento	Test	Total
<b>Glioma</b>	826	100	926
<b>Meningioma</b>	822	115	937
<b>No Tumor</b>	395	105	500
<b>Tumor Pituitario</b>	827	74	901
<b>Total</b>	2870	394	3264

Tabla 6-1: Estructura del conjunto de datos de Kaggle.

### 6.3. Combinación de los conjuntos de datos

Como ya se ha comentado, el conjunto de datos empleado para el desarrollo del proyecto es una combinación de las dos colecciones explicadas con anterioridad. Durante el proceso de su elaboración, se encontraron ciertas dificultades para poder realizarlo satisfactoriamente. Es cierto que la gran mayoría de las imágenes de los dos conjuntos tienen un mismo tamaño, el cual es de 512x512. Sin embargo, el principal problema viene provocado por la existencia de algunas muestras en la colección de Kaggle que no seguían este estándar. Por lo tanto, se tuvo que efectuar un proceso manual de redimensionamiento para cada una de las imágenes que no tuviesen un tamaño de 512x512. En primer lugar, se tuvieron que detectar todas aquellas imágenes que no tuviesen un tamaño de 512x512. Posteriormente, se identificaba la dimensión de mayor nivel de cada una de las muestras y se ajustaba a una magnitud de 512. Por último, se centraba cada imagen y se rellenaba con píxeles negros aquella dimensión cuyo tamaño era inferior a 512.

Una vez completado el procedimiento descrito en el anterior párrafo, ya se pudo crear sin impedimentos el conjunto de datos. El volumen de muestras está estructurado en dos carpetas, una para entrenamiento y la otra para testeo. Cada una de las carpetas está dividida en 4 subcarpetas (glioma, meningioma, no tumor y tumor pituitario), en donde en cada una se almacenan las imágenes de su respectiva clase. Para el desarrollo del proceso de aprendizaje de los modelos, se ha optado por dividir la colección de datos en 3 *sets* o conjuntos:

- Conjunto de entrenamiento: datos utilizados para el entrenamiento del modelo. El 80 % de las imágenes del conjunto se han empleado para este propósito y se han obtenido de la carpeta de entrenamiento.
- Conjunto de validación: conjunto utilizado para la evaluación del rendimiento del modelo después de cada época. Un 10 % de las imágenes han sido destinadas para este propósito y se han recogido de la carpeta de entrenamiento.
- Conjunto de testeo: lote empleado para comprobar el rendimiento del modelo después del finalizar el entrenamiento. Se ha utilizado un 10 % de las imágenes para la realización de esta tarea, las cuales se recogieron de la carpeta de testeo.

En la Tabla 6-2, se puede observar el número total de imágenes de cada clase utilizadas dentro de los conjuntos descritos.

<b>Clases</b>	<b>Entrenamiento</b>	<b>Validación</b>	<b>Test</b>	<b>Total</b>
<b>Glioma</b>	1881	236	235	2352
<b>Meningioma</b>	1321	165	163	1649
<b>No Tumor</b>	402	50	48	500
<b>Tumor Pituitario</b>	1470	183	188	1841
<b>Total</b>	5074	634	634	6342

*Tabla 6-2: Estructura del conjunto de datos elaborado.*

# Capítulo 7. Redes neuronales

El entrenamiento de redes neuronales supone en una gran cantidad de ocasiones un total quebradero de cabeza para los investigadores o profesionales que se dedican a este sector. La limitación de recursos computacionales suele ser una de las principales restricciones que impiden el correcto desarrollo del proceso de aprendizaje de un modelo. Sobre todo, dicho obstáculo es más notable en el entrenamiento de aquellas redes que cuentan con un gran volumen de pesos, como es en el caso de los *Transformers* seleccionados para el desarrollo del clasificador. Por lo que, para superar esta dificultad, se emplearon modelos pre-entrenados previamente con un conjunto de datos de gran magnitud y que poseen ya de por sí un gran rendimiento. Estos fueron adaptados al problema en cuestión mediante *Transfer Learning*. La transferencia de aprendizaje consiste en la adición de una nueva capa oculta y de salida a los modelos, los cuales serán los elementos sujetos que serán entrenados durante el proceso de aprendizaje. Los pesos de las arquitecturas se congelan durante el entrenamiento, ya que su misión es la de transmitir sus conocimientos a las nuevas capas añadidas.

Para el desarrollo del proyecto, se escogieron una serie de modelos de las siguientes arquitecturas:

- *Vision Transformer*: primer *Transformer* puro desarrollado para la resolución de tareas de visión por computador. De esta arquitectura, se seleccionaron los modelos ViTB16, ViTB32 y ViTL16, los cuales se tratan de dos modelos bases y uno largo. El número que identifica a la red hace referencia al tamaño de las porciones (*patches*) en las que se troceará la imagen de entrada.
- *Data-Efficient Image Transformer*: arquitectura basada en ViT, la cual integra un mecanismo denominado destilación que busca reducir los datos necesarios para el entrenamiento de *Transformers*. Los modelos utilizados de esta arquitectura han sido DeiT224, DeiT224 y DeiT224, los cuales se tratan de los modelos diminuto (*tiny*), pequeño (*small*) y base (*base*). Dichas redes solo pueden ser entrenados con imágenes de 224x224.
- *Swin Transformer*: topología que propone ciertas mejoras con respecto a la arquitectura ViT, como el *patch merging*. Los modelos entrenados han sido SwinT224, SwinS224 y SwinB224, los cuales son las redes *tiny*, *small* y *base*. Al igual que los modelos de la anterior arquitectura, estos solo pueden ser entrenados con imágenes de 224x224.

Modelos	Top-1 Accuracy	Dataset	Parámetros	Resolución de imagen
VitB16	0.8453	ImageNet2012	86M	384
VitB32	0.8166	ImageNet2012	88M	384
VitL16	0.8447	ImageNet2012	303M	384
DeiT224	0.7220	ImageNet1k	5M	224
DeiT224	0.7980	ImageNet1k	22M	224
DeiT224	0.8180	ImageNet1k	86M	224
SwiT224	0.8120	ImageNet1k	28M	224
SwiT224	0.8320	ImageNet1k	50M	224
SwiT224	0.8350	ImageNet1k	88M	224

Tabla 7-1: características de los modelos empleados. Fuente original: Github [74] [75] [76].

## 7.1. Transformers

Una red de tipo *Transformer* es un modelo que es capaz de identificar mediante el contexto las relaciones existentes entre una serie de datos secuenciales entrantes e interpretar el significado de estos. Esta arquitectura fue presentada por Ashish Vaswani et al. [77] en el artículo *Attention is all you need* en el año 2017 y se emplea principalmente en el campo del procesamiento del lenguaje natural (NLP), para la traducción, predicción de texto y generación de resúmenes [78] [79]. La aparición los *Transformers* ha causado una completa revolución en el paradigma de este campo de la inteligencia artificial, provocando que las arquitecturas más dominantes a lo largo de estos años, como RNN o LSTM, sean totalmente reemplazadas [79]. Diversas herramientas, como ChatGPT, el cual se trata de un *chatbot* muy conocido por el público en general debido a su sorprendente rendimiento a la hora de realizar tareas complejas, utilizan *Transformers* [81].

Según [80] [82], los *Transformers* se basan esencialmente en el uso de módulos de atención propia (*Self-Attention*), que permiten la paralización en el tratamiento de los datos entrantes y en una serie de codificadores-decodificadores. El texto es transformado en diversos vectores numéricos que contienen el valor de la palabra representada y su posición en la sentencia. Los módulos de atención identifican el contexto de una frase a través del cálculo de una matriz, la cual muestra las dependencias o relaciones existentes entre las diferentes palabras que la componen. Los codificadores (*encoders*) se encargan del procesamiento del texto de entrada mediante un *Multi-Head Attention*. Este consiste en una serie de módulos de atención que proporcionan diferentes interpretaciones de la frase, una red neuronal *fully-connected* y una serie de capas de normalización. A través de estos elementos, los codificadores son capaces de calcular un vector de contexto fijo de la frase entrante. Los decodificadores (*decoders*) son los responsables de generar el texto de salida. Estos van elaborando paso por paso cada una de las palabras de la oración final, aprovechando como entrada la frase desarrollada hasta ese momento. Los decodificadores están compuestos por dos *Multi-Head Attention*, una red neuronal *fully-connected* y tres capas de normalización. El primer mecanismo de *Multi-Head*

*Attention*, también denominado *Masked Attention*, detecta el contexto de la sentencia elaborada en el paso anterior. La segunda capa usa este contexto junto al vector generado por los codificadores para decidir en qué lugar o lugares de la frase de entrada debería centrarse el decodificador para generar la nueva palabra. La salida de esta capa se pasa a una red neuronal que calcula un vector de probabilidades de tamaño igual al abecedario del lenguaje utilizado por el modelo. Cada celda del vector representa una palabra y contiene un valor entre 0 a 1. El decodificador escribirá la palabra que mayor valor probabilístico tenga, es decir, que más cerca este de 1.

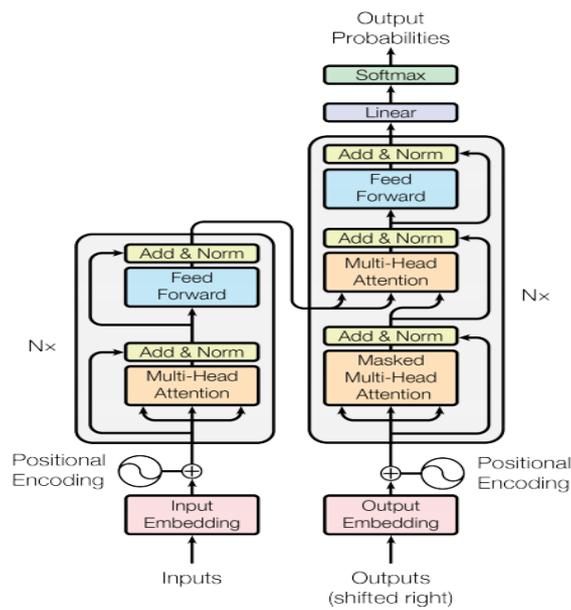


Figura 7-1: Estructura de un Transformer. Fuente original: Ashish Vaswani et al. [78].

### 7.1.1. Vision Transformer

Actualmente, los *Transformers* no se están utilizando solo para realizar tareas de procesamiento de texto, sino que también se están empleando dentro del campo de la visión por computador. El artículo de Alexey Dosovitskiy et al. [20] *An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale* publicado en 2020, muestra como los denominados *Vision Transformer* también son capaces de resolver problemas relacionados con la clasificación de imágenes. En el año 2021, el equipo de Google Research y Google Brain publicó [83], artículo en el que se explicaba una serie de ventajas que poseían este tipo de redes con respecto a las CNN.

Según se explica en [84] [85], el funcionamiento del modelo ViT propuesto por Dosovitskiy consiste en un bloque de codificadores, empleados para la extracción de características de la imagen, y una red *MultiLayer Perceptron* (MLP), la cual es denominada como *MLP Head* y es la encargada de la clasificación. Estos están compuestos por un *Multi-Head Attention*, dos capas de normalización y una red MLP. En primer lugar, la imagen es dividida en porciones, las cuales se convierten en vectores numéricos. A estos vectores se les inyecta cierta información

correspondiente a la posición que tenían dentro de la imagen. Al *Multi-Head Attention* se le pasa un vector numérico especial denominado token de clase y todas las porciones transformadas anteriormente. El mecanismo de atención identifica una serie de dependencias entre las diferentes zonas de la imagen, que va recopilando dentro del token de clase. De esta forma, el token almacena una representación global de la imagen, lo que permite al modelo comprenderla mejor en su conjunto. Posteriormente, el token de clase se pasa como entrada a la red MLP del codificador, que es la encargada de la extracción de las características más relevantes. Finalmente, estos rasgos extraídos son mandados a la *MLP Head*, la cual lleva a cabo la clasificación de la imagen en base a esta información.

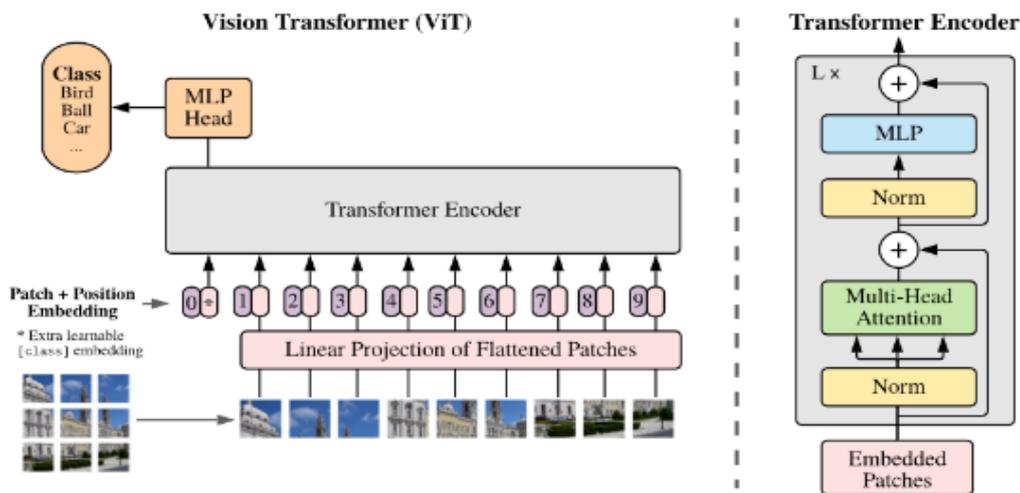


Figura 7-2: Estructura de ViT. Fuente original: Github [76].

### 7.1.2. Data-Efficient Image Transformer

*Data-Efficient Image Transformer* es un modelo basado en la arquitectura de ViT desarrollado por investigadores de Facebook, publicado por Hugo Touvron et al. [85] en el año 2020.

En [86] [87], se menciona que el principal problema presentado por ViT es el gran número de imágenes y de recursos necesarios para llevar a cabo el entrenamiento de esta clase de redes. DeiT propone un mecanismo de entrenamiento basado en la destilación, que provoca que este modelo necesite un conjunto de imágenes mucho menor que su antecesor para la obtención de resultados competitivos. La destilación se basa principalmente en el uso de una red neuronal profesora, cuya misión es la de transferir sus conocimientos a una red estudiante. En este caso, los investigadores utilizaron un modelo CNN como profesor, ya que esta clase de arquitectura necesita una menor cantidad de datos para ser entrenada, y una red de tipo *Transformer* como estudiante. Al igual que sucede en ViT, al codificador de DeiT se le pasan una serie de porciones de la imagen y un token de clase. Otra innovación que fue presentada por el equipo de Facebook fue la introducción de un nuevo token denominado token de destilación (*distillation token*). Este vector numérico es empleado por el *Transformer* para comprobar las diferencias entre las predicciones realizadas por el estudiante y el profesor. De esta forma, el modelo se va ajustando a los conocimientos del CNN.

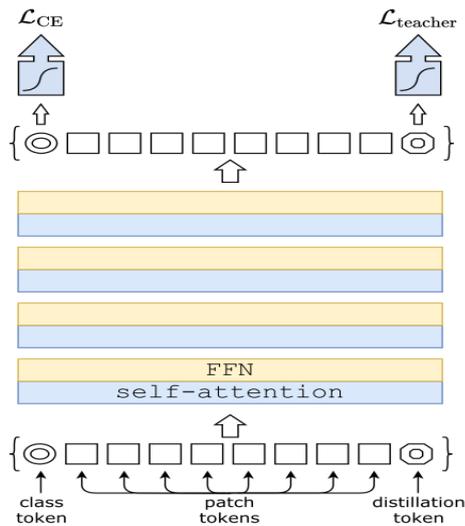


Figura 7-3: Estructura de DeiT. Fuente original: Meta AI [86].

### 7.1.3. Swin Transformer

*Swin Transformer* o *Shifted window Transformer* es un modelo basado en ViT publicado por investigadores de Microsoft en el año 2021 en el artículo *Swin Transformer: Hierarchical Vision Transformer using Shifted Windows* [88].

En [89] [90], se mencionan diferentes novedades implementadas por Swin con respecto a su antecesor, como los mapas de características jerárquicos o el desplazamiento de ventanas de atención. Este modelo emplea una serie de codificadores puestos de forma jerárquica. El codificador con menor rango es el que recibe como entrada las porciones de la imagen convertidas en vectores numéricos. En cada una de las entradas del resto de codificadores se encuentra un mapa de característica que aplica un mecanismo denominado *patch merging*. La finalidad de esta técnica es la de fusionar y combinar el tensor resultante del codificador anterior, con el fin de pasar un número menor de porciones al codificador del siguiente nivel. Uno de los principales problemas que tiene el ViT original es la enorme cantidad de recursos computacionales que requiere el mecanismo de *Multi-Head Attention*, sobre todo en aquellas imágenes de gran dimensión. Por ello, los investigadores decidieron dividir cada bloque del *Transformer* en dos secciones, las cuales usan módulos de atención modificados. En la primera sección, el *Multi-Head Attention* emplea 4 ventanas de atención para cada una de las regiones de la matriz. Cada ventana identifica las dependencias y relaciones existentes entre los píxeles de su zona en cuestión sin tener en cuenta el resto de regiones. En la segunda sección, las ventanas son desplazada para que, de esta manera, puedan identificar nuevos enlaces. De esta forma, la carga computacional se reduce drásticamente.

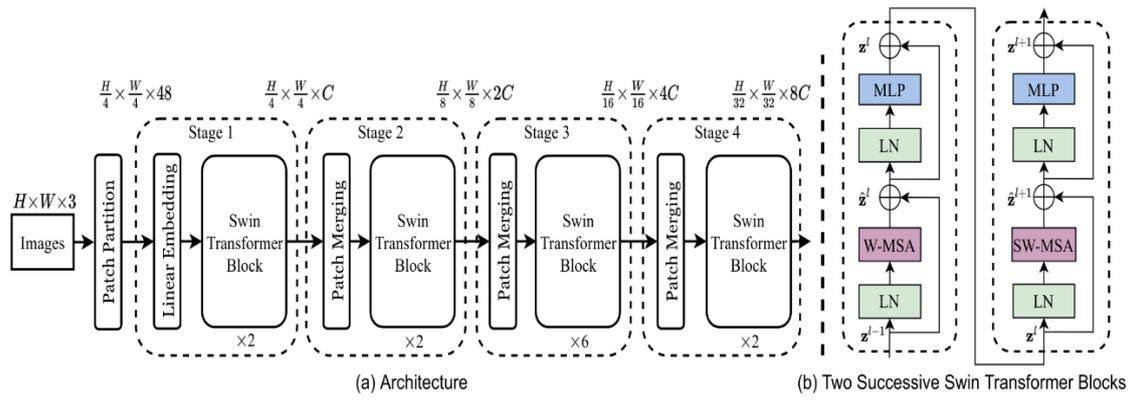


Figura 7-4: Estructura de Swin. Fuente original: Medium [90].

# Capítulo 8. Configuración de los experimentos

Como ya se ha mencionado anteriormente, se empleó la transferencia de aprendizaje como mecanismo para facilitar el entrenamiento de los *Transformers*. Algunos modelos también fueron entrenados usando aumento de datos para aumentar la variedad del conjunto de imágenes empleado en el entrenamiento de los modelos. El aumento de datos crea nuevas muestras mediante la aplicación de una serie de operaciones a las secuencias ya existentes en la colección. Este provoca una reducción considerable de posibles sobreajustes que puedan ser producidos durante el proceso de aprendizaje. Por lo tanto, se entrenaron un total de 18 modelos: 9 utilizando exclusivamente transferencia de aprendizaje y otros 9 que han sido entrenados mediante el empleo tanto de transferencia de aprendizaje y aumento de datos. Las operaciones utilizadas para la generación de nuevas imágenes, a través de aumento de datos, fueron las siguientes: rotaciones horizontales, rotaciones aleatorias de entre -10 % a 10 % y *zooms* aleatorios de entre un -20 % a un 20 %.

Cabe destacar que las imágenes del conjunto de datos tenían un tamaño de 512x512. Dichas imágenes fueron reescaladas a un tamaño de 224x224, ya que ciertos modelos solo podían ser entrenados con muestras de estas dimensiones. Los valores de los píxeles de las imágenes se encontraban en un rango de entre 0 a 255. Para la facilitación del proceso de entrenamiento, se optó por reescalar los valores en un intervalo de entre -1 a 1.

A continuación, se procede a explicar la configuración de hiperparámetros empleada en cada uno de los entrenamientos y las métricas utilizadas para la evaluación de los resultados obtenidos, una vez terminado el proceso de aprendizaje.

## 8.1. Configuración de hiperparámetros

Según [91], los hiperparámetros son las variables que marcan la configuración del entrenamiento de un modelo y son modificadas manualmente por el usuario. Un ejemplo de hiperparámetro sería el denominado tasa de aprendizaje (*learning rate*), el cual marca la magnitud de la actualización de los pesos por parte del optimizador. Una buena configuración de estos valores es fundamental para el correcto aprendizaje del modelo. No existe una forma precisa para el establecimiento de estos valores por lo que se deben ajustar manualmente mediante la realización de diversas pruebas experimentales. Otros hiperparámetros que son fundamentales en el proceso de aprendizaje se encuentran abordados en [92] [93] [94] y son los siguientes:

- Algoritmo de optimización: se encarga de actualizar los valores de los pesos a través del cálculo del gradiente de la función de error para cada parámetro.
- Función de activación: función utilizada en cada neurona que marca si esta es activada o no según los valores que tome de entrada.

- Función de pérdida: función usada para calcular el error cometido en cada iteración por la red.
- Número de neuronas y de capas: se refiere al número de capas que tendrá el modelo y el número de neuronas que tiene cada capa.
- Época (*Epoch*): número de iteraciones que marca las veces que se va a entrenar y validar la red neuronal.
- *Batch size*: marca el tamaño del lote de datos que se va a emplear cada vez que se entrene el modelo.
- *Dropout*: porcentaje de la red neuronal cuyos pesos no son entrenados durante cada entrenamiento.

En los siguientes apartados, se procede a explicar los hiperparámetros seleccionados para cada una de las arquitecturas.

### 8.1.1. Algoritmo de Optimización

El algoritmo de Adam es uno de los optimizadores más utilizados para el entrenamiento de modelos destinados para la resolución de problemas de visión por computador. Como se explica en [95] [96], Adam se originó en el año 2014, pero no fue presentado por primera vez hasta el 2015 en la conferencia *ICLR 2015*. Dicha conferencia se trata de una de las más conocidas por los especialistas en aprendizaje profundo. El optimizador se denomina Adam debido a que adapta la magnitud de la tasa de aprendizaje para cada uno de los pesos a partir de las medias de los primeros y segundos momentos de los gradientes. Adam combina los beneficios proporcionados por los optimizadores AdaGrad y RMSProp, buscando de esta manera una mayor eficiencia que la proporcionada por el algoritmo del gradiente descendente estocástico. El algoritmo utiliza la siguiente configuración de parámetros:

- *Alpha*: se trata de la tasa de aprendizaje usada por el optimizador.
- *Beta1* ( $\beta_1$ ): hace referencia al factor de decaimiento utilizado para el cálculo de la media del primer momento.
- *Beta2* ( $\beta_2$ ): hace referencia al factor de decaimiento usado para el cálculo de la media del segundo momento.
- *Epsilon*: parámetro de ínfimo valor numérico empleado para evitar divisiones entre 0.

Para la realización del proceso de entrenamiento, se utilizaron los valores por defecto dados por Keras, excepto la tasa de aprendizaje que varía dependiendo del modelo a entrenar. La configuración de parámetros por defecto presentada por Keras es la siguiente:  $\text{Alpha}=0.001$ ,  $\beta_1=0.9$ ,  $\beta_2=0.999$  y  $\text{Epsilon}=10^{-7}$ .

### 8.1.2. Funciones de activación y pérdida

*Rectifier Linear Unit* (ReLU) es una función de activación lineal muy usada por modelos de aprendizaje profundo en sus capas ocultas, debido a su gran velocidad y a la reducción de posibles problemas de saturación del gradiente [97]. Si el valor de entrada es negativo, la función

siempre va a devolver 0. Cuando el valor de entrada es positivo, el valor de salida va a ser igual al de entrada. La función ReLU sigue la siguiente forma [98]:

$$f(x) = \begin{cases} 0, & \text{si } x < 0 \\ x, & \text{si } x > 0 \end{cases}$$

*Softmax* es la función de activación más empleada por los modelos destinados a la clasificación de múltiples clases, ya que es capaz de convertir un vector numérico de longitud K en un vector de la misma dimensión con valores que oscilen entre [0,1] y cuya suma total sea igual a 1 [99]. *Softmax* es utilizada en las capas finales de una red para calcular un vector de probabilidades cuya longitud sea igual al número total de clases a clasificar [98]. Es decir, cada posición del vector resultante representa una clase del problema abarcado. Para realizar la clasificación, el modelo selecciona la clase cuyo valor probabilístico sea mayor. En el caso del problema planteado en este proyecto, el número de clases presentes en el conjunto de datos es igual a 4. Por lo que, el tamaño del vector de probabilidades generado por la capa oculta de los modelos será equivalente a dicha longitud. La función *Softmax* sigue la siguiente fórmula, la cual fue obtenida de [98]:

$$\sigma(\vec{z})_i = \frac{e^{z_i}}{\sum_{j=1}^K e^{z_j}}$$

*Categorical Cross-Entropy* es una función de pérdida altamente utilizada para el cálculo del error cometido por un modelo de clasificación múltiple. Consiste en una combinación entre la función de activación *Softmax* y la función de pérdida *Cross-Entropy* [99]. La fórmula empleada por *Categorical Cross-Entropy* es la siguiente [100]:

$$CE = - \sum_{i=1}^{i=N} y_i \cdot \log(\hat{y}_i)$$

### 8.1.3. Comparativa de hiperparámetros

Como ya se ha comentado con anterioridad, no existe ningún tipo de método empírico y exacto para calcular la configuración óptima de hiperparámetros para un determinado modelo. Por lo que, el ajuste de los hiperparámetros se suele llevar a cabo mediante sucesivas pruebas de ensayo y error. Intentar entrenar distintos modelos puede resultar una tarea muy complicada a la hora de encontrar una configuración que sea óptima y común, ya que ciertos conjuntos de hiperparámetros pueden dar resultados excelentes en ciertas arquitecturas y en otras puede suceder lo opuesto. Este mismo problema pudo ser observado durante el desarrollo del proyecto, en donde un mismo ajuste de hiperparámetros daba resultados sobresalientes para los modelos ViT y DeiT. Sin embargo, cuando se empleó esta misma configuración para los entrenamientos de las redes Swin, se detectaron problemas de subajuste que impedían que los modelos llegasen a su máxima capacidad de rendimiento. Por este principal motivo, se decidió

utilizar para el proceso de aprendizaje de los Swins un ajuste de hiperparámetros distinto, el cual eliminara las incidencias encontradas e incrementase los niveles de desempeño de estos.

Como la principal prioridad del proyecto era desarrollar un clasificador cuya eficacia y precisión sean las mayores posibles, el autor optó por buscar aquellas configuraciones que maximicen el rendimiento de los *Transformers* antes que intentar hallar unos valores comunes de hiperparámetros que limiten el potencial de ciertas arquitecturas. En la Tabla 8-1, se puede visualizar una comparativa entre los diferentes ajustes empleados para el entrenamiento de cada uno de los modelos. Cabe aclarar que las funciones de activación y pérdida usadas para cada uno de los entrenamientos, así como el optimizador, fueron las descritas en los anteriores puntos.

Modelos	Tamaño de la Capa Oculta	Tasa de Aprendizaje	Número de Épocas	Batch Size	Dropout
VitB16	2048	0.0001	30	48	0.3
VitB32	2048	0.0001	30	48	0.3
VitL16	2048	0.0001	30	48	0.3
DeiT224	2048	0.0001	30	48	0.3
DeiT524	2048	0.0001	30	48	0.3
DeiT824	2048	0.0001	30	48	0.3
SwimT224	512	0.001	30	32	0.3
SwimS224	512	0.001	30	32	0.3
SwimB224	512	0.001	30	32	0.3

Tabla 8-1: Configuraciones de hiperparámetros.

## 8.2. Métricas empleadas

Una de las principales partes de todo proyecto relacionado con el desarrollo de aplicaciones de *Deep Learning*, es la evaluación de los resultados obtenidos en los entrenamientos de los modelos. Dicha evaluación se suele llevar a cabo a través de una serie de métricas que ofrecen una visualización general o profunda del rendimiento de cada una de las redes, durante las distintas fases del proceso de aprendizaje. El uso de métricas permite observar la evolución que sigue el desempeño de un cierto modelo a lo largo del entrenamiento y detectar posibles problemas de sobreajustes. Para el cálculo de las métricas se hace uso de los siguientes valores, explicados en [101]:

- *True Positive* (TP): número de muestras en las que se indica correctamente la aparición de ciertas propiedades o rasgos.
- *False Positive* (FP): número de muestras en las que se indica erróneamente la aparición de ciertas propiedades o rasgos.

- *True Negative* (TN): número de muestras en las que se indica correctamente la falta de presencia de ciertas propiedades o rasgos.
- *False Negative* (FN): número de muestras en las que se indica erróneamente la falta de presencia de ciertas propiedades o rasgos.

Para representar visualmente estos valores, se ha optado por utilizar matrices de confusión. A continuación, se indican las métricas que fueron utilizadas para la evaluación de los resultados:

- *Accuracy*: La precisión (*accuracy*) es una de las métricas más empleadas, ya que permite conocer el nivel de precisión de un modelo a la hora de clasificar los distintos elementos de un conjunto de datos correctamente [102]. Dicha métrica se calcula como la división entre el número de predicciones correctas y el número total de predicciones [103]:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

- *Precision*: La precisión (*precision*) es una métrica fundamental para la evaluación del rendimiento de un modelo, debido a que visualiza el porcentaje de detecciones positivas verdaderas con respecto a todas las predicciones positivas hechas [104]. La precisión se calcula de la siguiente forma [103]:

$$Precision = \frac{TP}{TP + FP}$$

- *Recall*: La recuperación (*recall*) es una métrica de rendimiento que estima el porcentaje de detecciones positivas correctas con respecto al número total de muestras que realmente son positivas [105]. El cálculo del *recall* se realiza de la siguiente manera [104]:

$$Recall = \frac{TP}{TP + FN}$$

- *F1 Score*: métrica de gran utilidad para la evaluación general del rendimiento de un modelo, ya que combina los valores de precisión y recuperación mediante el cálculo de su media armónica [107]:

$$F1\ Score = 2 \times \frac{Precision \times Recall}{Precision + Recall}$$

# Capítulo 9. Evaluación de los resultados obtenidos

Una vez terminada la fase de entrenamiento, se ha realizado una evaluación profunda de los resultados obtenidos por cada uno de los modelos. Para ello, se han empleado una serie de métricas que permiten estimar los niveles de desempeño alcanzados por estos. Las arquitecturas han sido evaluadas a través de las siguientes medidas, las cuales fueron explicadas con anterioridad: *accuracy*, *precision*, *recall* y *F1 score*. Además, se han empleado diversas representaciones visuales como matrices de confusión y gráficas que facilitan la visualización de los resultados. Finalmente, se ha llevado a cabo una comparativa entre los resultados obtenidos por los modelos propuestos y los alcanzados por otros autores, y una explicación sobre las aportaciones científicas proporcionadas por este proyecto para futuras investigaciones.

## 9.1. Precisión media de los modelos

La precisión (*accuracy*) se ha utilizado principalmente para observar la evolución del rendimiento de los modelos durante el transcurso del entrenamiento. Para facilitar su visualización, se han empleado gráficas que muestran el nivel de precisión y pérdida en cada una de las épocas al llevar a cabo la clasificación de los elementos del conjunto de entrenamiento y validación. Asimismo, se ha hecho uso de esta métrica para calcular el desempeño global de los modelos durante el proceso de entrenamiento, validación y testeo.

En la Tabla 9-1, se pueden observar los resultados obtenidos por aquellos modelos que fueron entrenados sin realizar un aumento de datos. Estas redes de tipo *Transformer* alcanzaron un grado de precisión sobresaliente durante el proceso de entrenamiento, superando en todos los casos el 94 % de precisión. La arquitectura DeiT fue la que mejor rendimiento logró con este lote de muestras, siendo en concreto el modelo DeITS224 el que mejor desempeño demostró con un 100 % de precisión. DeITT224 y DeITL224 no se quedaron atrás mostrando una precisión mayor al 99.90 %. Las redes Swins también ofrecieron unos resultados excelentes, no bajando del umbral del 97.50 %. Los modelos SwinT224 y SwinS224 lograron una precisión mayor del 99.30 %, quedándose estos muy cerca de los valores alcanzados por los DeITs. La arquitectura ViT fue la que peor rendimiento mostró, ya que ningún modelo pudo superar la barrera del 97 % de precisión.

Durante la fase de validación, también se registraron valores magníficos. La cifra de menor magnitud fue la registrada por ViTB16 con un 91.48 %. Los modelos Swins y DeITs fueron los que obtuvieron los resultados más destacados. En concreto, SwinT224 y SwinS224 destacaron por encima del resto con un 98.11 % y un 97.48 % de precisión, respectivamente. El *Transformer* DeITB224 también alcanzó un impresionante grado de rendimiento en lo que concierne a esta métrica, llegando a un 96.69 %. De nuevo, las redes ViTs fueron las menos precisas. El modelo con mejor desempeño fue el modelo ViTB32 con un 93.53 %.

En cuanto al grado de precisión logrado durante la etapa de testeo, los dominadores volvieron a ser las arquitecturas DeiT y Swin. Los modelos que dieron un mejor rendimiento fueron DeiT224 con un 97.16 % y Swin224 con un 96.21 %. El resto de las redes también mostraron un desempeño más que destacable, superando estas la barrera del 90 %.

Como se puede apreciar en la Tabla 9-2, los modelos entrenados mediante aumento de datos obtuvieron resultados mucho más pobres. En este caso, las redes DeITs fueron las que peores tasas de precisión alcanzaron durante el proceso de entrenamiento. El único modelo de esta arquitectura que pudo superar el umbral del 90 % fue DeITB224. La arquitectura ViT logró sorprendentemente un mejor grado de desempeño que DeiT. Dos de los tres *Transformers* de este tipo lograron un nivel de precisión superior al 90 %, siendo ViTL16 el que mejor rendimiento consiguió con un 91.98 %. Se puede observar fácilmente que los *Transformers* Swins fueron los que mejores valores consiguieron, ya que los tres modelos registraron números que sobrepasaron el 91 % de precisión. Específicamente, las redes SwinS224 y SwinB224 lograron los mejores resultados, llegando a un 93.48 % y un 92.10 %, respectivamente.

En la etapa de validación, las cifras en lo que refiere a la precisión variaron en gran magnitud entre los diferentes modelos. Dicha fluctuación se hace presente sobre todo en los resultados mostrados por las redes ViTs. Mientras que ViTL224 alcanzó un notable nivel de desempeño logrando un 92.11 % de precisión, ViTB224 fue el peor de los nueve modelos, ya que registró una tasa del 86.12 %. Una vez más, los Swins superaron en términos de rendimiento al resto de redes. Sin embargo, la diferencia de resultados no fue tan marcada y amplia como la mostrada durante el periodo de entrenamiento. Dos de sus modelos se quedaron cerca del 90 % de precisión y SwinS224 fue la red más precisa con un 92.11 %.

Durante la fase de testeo, se visualizaron resultados bastante pobres por parte de todos los modelos. Cabe destacar la mediocridad de los valores registrados por los DeITs, ya que incluso una de las redes de esta arquitectura fue incapaz de superar el 85 % de precisión. SwinS224 fue el modelo más preciso con un 90.53 %. El resto de *Transformers* se quedaron atrás con respecto a este modelo, puesto que no pudieron superar el umbral del 90 % de precisión.

Sin Aumento de Datos			
Modelos	Training Accuracy	Validation Accuracy	Test Accuracy
ViTB16	94.76 %	91.48 %	90.54 %
ViTB32	97.32 %	93.53 %	93.84 %
ViTL16	94.92 %	92.59 %	92.27 %
DeiTT224	99.92 %	94.79 %	<b>97.16 %</b>
DeiTS224	<b>100.00 %</b>	95.43 %	95.90 %
DeiTB224	99.94 %	96.69 %	95.74 %
SwinT224	99.55 %	<b>98.11 %</b>	96.21 %
SwinS224	99.35 %	97.48 %	95.42 %
SwinB224	97.50 %	95.26 %	93.53 %

Tabla 9-1: Precisión media sin aumento de datos.

Con Aumento de Datos			
Modelos	Training Accuracy	Validation Accuracy	Test Accuracy
ViTB16	88.33 %	86.12 %	86.90 %
ViTB32	90.68 %	89.27 %	87.70 %
ViTL16	91.98 %	<b>92.11 %</b>	89.59 %
DeiTT224	88.55 %	87.70 %	87.85 %
DeiTS224	87.39 %	86.44 %	84.38 %
DeiTB224	90.40 %	90.38 %	88.33 %
SwinT224	92.10 %	89.75 %	89.59 %
SwinS224	<b>93.48 %</b>	<b>92.11 %</b>	<b>90.53 %</b>
SwinB224	91.58 %	89.75 %	88.64 %

Tabla 9-2: Precisión media con aumento de datos.

Como ya se ha abordado anteriormente, las tasas de precisión obtenidas por los modelos entrenados sin aumento de datos fueron ampliamente superiores a la de las redes que fueron entrenadas empleando dicho método para la ampliación de colecciones de muestras. Por lo tanto, debido a esta enorme diferencia en términos de rendimiento, se ha decidido analizar exclusivamente con mayor nivel de profundidad las arquitecturas que fueron entrenadas sin aumento de datos.

En los siguientes puntos, se procederá a realizar un análisis exhaustivo de las gráficas de aprendizaje y de pérdida de los modelos, de cada una de las topologías, que mejores resultados

lograron en el periodo de testeo. Tanto en la gráfica de aprendizaje como en la de pérdida, se visualiza la evolución del nivel de rendimiento de una determinada red en cada una de las épocas durante las etapas de entrenamiento y validación. El objetivo principal del análisis es la detección de posibles problemas de sobreajuste o subajuste generados a lo largo del proceso de aprendizaje.

El sobreajuste (*overfitting*) ocurre cuando la red memoriza el conjunto de datos utilizado para el entrenamiento en vez de aprender sus rasgos característicos, por lo que esta será incapaz de distinguir los patrones relevantes de nuevos datos [108].

El subajuste (*underfitting*) se produce cuando un modelo es entrenado con un conjunto de datos muy pequeño, lo que provoca que este no consiga aprender todo el conocimiento necesario para tener un buen rendimiento [108].

Para la identificación de este tipo de problemas, se suele observar el comportamiento presentado por el modelo en las gráficas de aprendizaje y pérdida. Si las gráficas tienen demasiadas fluctuaciones significa que la red sufre de sobreajuste y si estas se asemejan a una recta con poca pendiente, lo más probable es que padezca de subajuste.

### **9.1.1. Resultados obtenidos por la arquitectura ViT**

ViTB32 fue el modelo de la arquitectura ViT que mejores resultados obtuvo con un nivel de precisión del 97.32 % durante el entrenamiento, un 93.53 % en la fase de validación y un 93.84 % en la etapa de testeo.

Como se puede apreciar en la Figura 9-1, los valores de precisión de validación son superiores con respecto a los de entrenamiento en la mayoría de las épocas. Dicha diferencia es sobre todo más notoria durante el comienzo del proceso de aprendizaje, reduciéndose esta de gran manera una vez superada la época número 20. En la gráfica de pérdida, se puede visualizar esta misma diferencia de valores, siendo bastante menor la pérdida lograda en la fase de validación que en la de entrenamiento durante las primeras etapas. La disparidad entre ambas magnitudes no se vio decrementada hasta casi alcanzar la época 25. El hecho que puede haber ocasionado esta desigualdad inicial en ambas gráficas puede deberse al *Dropout*, ya que a la hora de entrenar la red va a dejar inactiva un porcentaje concreto de su estructura. Sin embargo, durante la validación esta limitación no se encuentra presente debido a que se utiliza la totalidad del modelo para la clasificación.

Inicialmente, tanto la tasa de precisión alcanzada en el entrenamiento como en la validación se encuentran por debajo del 90 %. Con el transcurso de las épocas, estos valores se incrementan ligeramente, logrando sobrepasar dicho umbral antes de la época 10 en el caso de la validación y no siendo superada hasta casi llegar a la época número 15 en la fase de entrenamiento. Los niveles de precisión logrados en la época 30, en ambos casos, se encuentran bastante cercanos al 95 %. Si se observa el comportamiento del modelo en la gráfica de aprendizaje, se puede apreciar como las cifras de entrenamiento son más estables que las de validación, ya que presentan un crecimiento mucho más ligero.

Este mismo patrón de comportamiento se puede apreciar en la gráfica de pérdida, por el hecho de que la función de entrenamiento presenta una suavidad mayor que la de validación. Asimismo, en ambas representaciones se puede percibir un ligero subajuste del modelo, puesto que las funciones de entrenamiento y validación no logran en ninguno de los periodos estabilizarse. Por lo que, podría ser necesario modificar el conjunto de hiperparámetros añadiendo un mayor número de épocas para el proceso de aprendizaje o un aumento sutil de la tasa de aprendizaje para solucionar este problema. Aun así, el rendimiento alcanzado por este modelo es bastante notable.

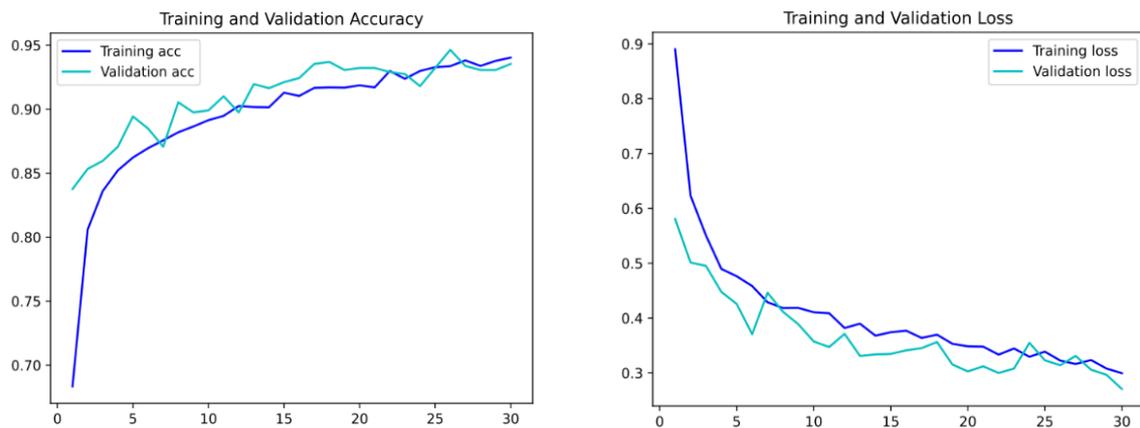


Figura 9-1: Gráficas de aprendizaje y error de ViT.

### 9.1.2. Resultados obtenidos por la arquitectura DeiT

En el caso de la arquitectura DeiT, el modelo que mejor desempeño demostró fue DeiT224 con un porcentaje de precisión del 99.92 % en el entrenamiento, un 94.79 % durante la validación y un 97.16 % en el testeo. La principal diferencia que se puede visualizar con respecto a ViTB32, es la permanente varianza entre los niveles de precisión relativos al periodo de entrenamiento y al de validación. La función de validación se encuentra constantemente por debajo de la de entrenamiento, la cual no logró superar la barrera del 95 % de precisión. En cambio, la mayoría de las cifras registradas a lo largo del entrenamiento se encuentran cercanas al 100 % de precisión. Lo mismo ocurre en la gráfica de pérdida, en donde las magnitudes más pequeñas fueron obtenidas por el modelo a lo largo del periodo de entrenamiento.

Tanto en la gráfica de pérdida como en la de aprendizaje, se puede apreciar como las funciones de entrenamiento y de validación logran estabilizarse una vez pasadas las 20 épocas. Como en el caso de ViTB32, los valores de precisión y de pérdida aumentan de forma ligera a lo largo del entrenamiento. Sin embargo, las cifras registradas de estas métricas durante la fase de validación sufren un mayor grado de inestabilidad.

Por todos estos motivos, se puede concluir que el modelo presenta algo de sobreajuste que podría ser solucionado mediante una variación de ciertos hiperparámetros, como el *batch size* o la tasa de aprendizaje. A pesar de este pequeño problema de sobreajuste, el rendimiento alcanzado por esta red es más que sobresaliente.

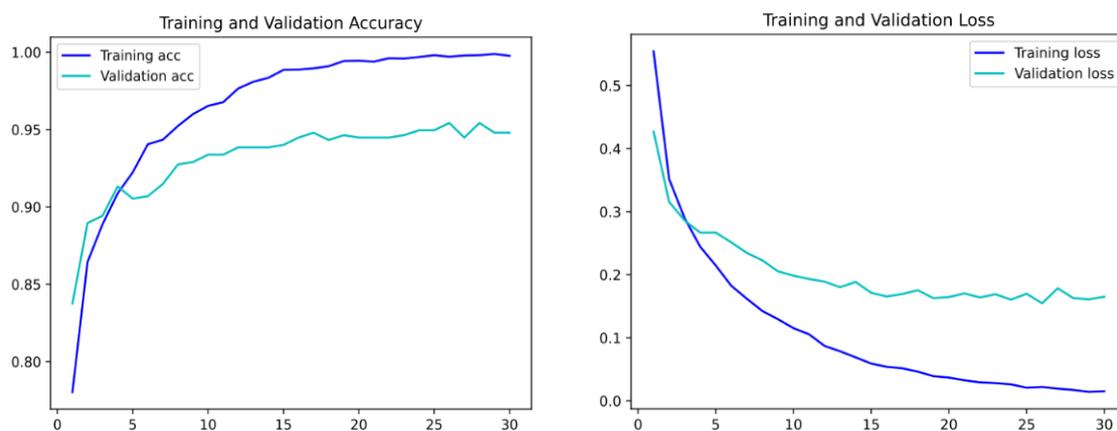


Figura 9-2: Gráficas de aprendizaje y error de DeiT.

### 9.1.3. Resultados obtenidos por la arquitectura Swin

SwinT224 fue el modelo de la arquitectura Swin que mejores resultados obtuvo con un nivel de precisión del 99.55 % durante el entrenamiento, un 98.11 % en la fase de validación y un 96.21 % en la etapa de testeo.

En la Figura 9-3, se puede apreciar como los valores de precisión mostrados durante la fase de validación son superiores con respecto a los de entrenamiento en la gran mayoría de las épocas. Dicha diferencia se mantiene relativamente estable a lo largo de todo el periodo de aprendizaje. Mientras que las tasas de precisión de validación se encuentran ampliamente por encima del 95 % prácticamente en todas las épocas, los niveles de esta métrica durante el entrenamiento no alcanzan este umbral hasta superar las 25 épocas. Como ya se comentó en el análisis de ViTB32, lo más probable es que esta diferencia de resultados se deba por el *Dropout*. Sin embargo, las cifras de pérdidas obtenidas tanto en el entrenamiento como en la validación son mayormente idénticas.

Al igual que en los anteriores modelos analizados, las cifras de precisión y de pérdida observadas a lo largo del entrenamiento aumentan de manera paulatina y controlada. Sin embargo, las funciones de pérdida y precisión de validación representadas tienen un comportamiento más inestable. Sobre todo, este último presenta una gran cantidad de picos ascendentes y descendentes.

Las tasas de precisión en ninguna de las fases logran estabilizarse, por lo que el modelo podría estar sufriendo de un pequeño problema de subajuste. Para solucionar este inconveniente, podría ser conveniente emplear un mayor número de épocas a la hora de realizar el proceso de aprendizaje. A pesar de esto, cabe destacar el magnífico rendimiento presentado por este modelo.

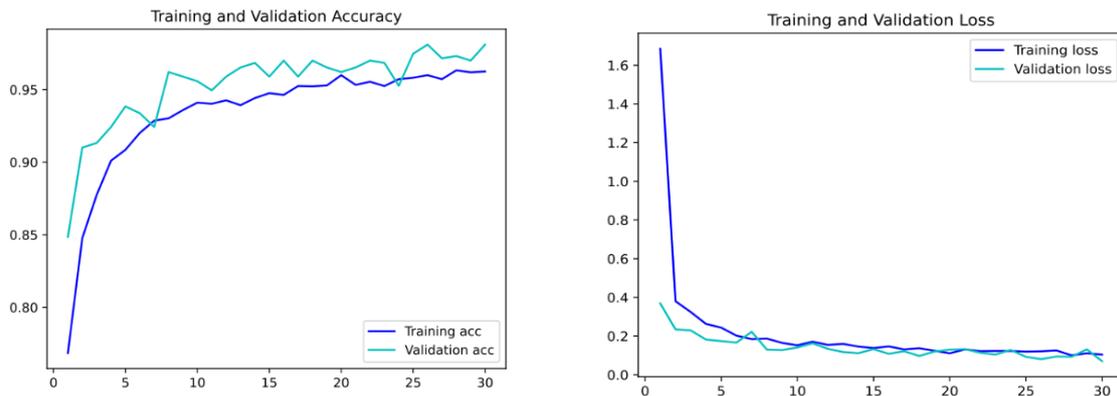


Figura 9-3: Gráficas de aprendizaje y error de Swin.

#### 9.1.4. Conclusiones basadas en la exactitud y el error

Como ya se ha comentado con anterioridad, los resultados obtenidos por parte de los modelos que fueron entrenados con aumento de datos y transferencia de aprendizaje fueron totalmente sobrepasados por aquellas redes que fueron tan solo entrenadas mediante esta última técnica. Por lo que, ese tipo de redes han sido descartadas de ser analizadas en posteriores apartados.

Entre los *Transformers* entrenados sin aumento de datos, destacan a simple vista los niveles de desempeño obtenidos por Swins y los DeiTs. Este tipo de arquitecturas demostraron unos elevadísimos porcentajes de precisión. En concreto, SwinT224 y DeiTB224 fueron los que mejores resultados alcanzaron a nivel general en las tres etapas del proceso de aprendizaje (entrenamiento, validación y testeo), por lo que fueron analizados más profundamente para poder detectar posibles problemas de sobreajuste o subajuste. Asimismo, también se evaluó el comportamiento del modelo ViTB32, ya que a pesar de que obtuviera peores resultados que las otras dos arquitecturas comentadas, se quería conocer si el máximo potencial de dicha red había sido alcanzado o no.

Los análisis realizados de las gráficas de aprendizaje y pérdida de los *Transformers* ViTB32, DeiT224 y SwinT224, arrojaron información de suma importancia relacionada con el comportamiento de estos durante las fases de entrenamiento y validación. ViTB32 y SwinT224 presentaron pequeños problemas de subajuste, por lo que se podría decir que estos todavía esconden cierto potencial rendimiento que podría ser explotado con ciertas modificaciones en algunos hiperparámetros. En lo que concierne a DeiT224, se pudo observar algunos problemas de sobreajuste.

En conclusión, en base a la información recopilada, el autor ha seleccionado como propuestas las redes SwinT224 y DeiT224. Este primero debido a sus excelentes niveles de precisión y la posible mejora que podría suponer la modificación de algunos hiperparámetros. Mientras que, DeiT224 ha sido escogida, ya que fue el modelo más exacto durante el testeo.

## 9.2. Evaluación del rendimiento por clases

Para medir el nivel de rendimiento de los modelos durante la clasificación de cada una de las clases del conjunto de datos, se emplearon como métricas la precisión (*accuracy*) y el número de aciertos obtenidos por estos en la fase de testeo.

Como se puede visualizar en la Tabla 9-3, los modelos apenas demostraron escasas dificultades a la hora de clasificar los Gliomas, siendo SwinB224 el que peor nivel de precisión obtuvo con un 94 %. El resto de redes demostraron un gran desempeño rondando entre un rango del 96-98 % de precisión, llegando incluso DeiTB224 a superar este mismo con un 99 %. Sin embargo, el rendimiento de los modelos varió en mayor medida durante el etiquetado de los meningiomas. Ciertos *Transformers*, como DeiTT224 o SwinT224, llegaron a alcanzar unas considerables tasas de precisión con un 96 % y un 91 %, respectivamente. Aunque, en la mayoría de casos, las redes no pudieron sobrepasar la barrera del 90 % de precisión e incluso algunos modelos como ViTL16 se quedaron lejos de esta marca con un 84 %. En cuanto a la detección de aquellas imágenes MRI que no contienen ninguna clase de tumor cerebral, una gran porción de los *Transformers* obtuvo magníficos resultados. ViTL16, SwinT224 y SwinS224 demostraron un excelente desempeño con un 100 % de precisión a la hora de etiquetar este tipo de muestras, mientras que otras redes como SwinB224 o ViTB16 registraron resultados lejanos a esta cifra con un 90 % y un 87 %. Durante la identificación de tumores pituitarios, se pudo visualizar más claramente la diferencia en términos de rendimiento entre las diferentes arquitecturas, mostrando los DeiTs y Swins bastante mejores tasas de exactitud que los ViTs. Sin embargo, a pesar de esta clara desigualdad, se puede decir que los niveles de desempeño alcanzados son bastante notables en general.

Porcentaje de Exactitud por Clase				
Modelos	Glioma	Meningioma	No tumor	Tumor Pituitario
ViTB16	96.00 %	85.00 %	87.00 %	90.00 %
ViTB32	98.00 %	89.00 %	96.00 %	93.00 %
ViTL16	98.00 %	84.00 %	<b>100.00 %</b>	93.00 %
DeiTT224	97.00 %	<b>96.00 %</b>	98.00 %	98.00 %
DeiTS224	98.00 %	91.00 %	98.00 %	97.00 %
DeiTB224	<b>99.00 %</b>	89.00 %	98.00 %	98.00 %
SwinT224	97.00 %	91.00 %	<b>100.00 %</b>	<b>99.00 %</b>
SwinS224	98.00 %	88.00 %	<b>100.00 %</b>	98.00 %
SwinB224	94.00 %	88.00 %	90.00 %	<b>99.00 %</b>

Tabla 9-3: Tasas de exactitud por clase.

Respecto al número de aciertos, se puede observar como DeiT224 y SwinT224 fueron los que mayores cifras lograron en la detección de gliomas, con un total de 228 y 225. ViTL16 fue la red

que peor rendimiento alcanzó con 204 identificaciones correctas. En lo que concierne al etiquetado de meningiomas, las arquitecturas DeiT y Swin fueron las que mejores desempeños alcanzaron. DeiTS224 fue la líder de este grupo con 160 aciertos, seguidos por SwinT224 y SwinS224 con una cifra de 158. El modelo ViTB16 fue la colista, registrando un total de 139 detecciones correctas. Las redes consiguieron rendimientos muy similares a la hora de clasificar cerebros sanos. Sorprendentemente, en este caso, un modelo de la arquitectura ViT fue la que logró el mayor número de aciertos, etiquetando hasta 48 muestras correctamente. Por último, con respecto a la clasificación de tumores pituitarios, las cifras de aciertos obtenidas por los diferentes modelos también son bastante parecidas. DeiT224 logró alcanzar un máximo de 187 identificaciones correctas.

Número de Aciertos por Clase				
Modelos	Glioma	Meningioma	No tumor	Tumor Pituitario
ViTB16	205	139	<b>48</b>	182
ViTB32	216	148	45	186
ViTL16	204	152	44	185
DeiTT224	<b>228</b>	155	46	<b>187</b>
DeiTS224	221	<b>160</b>	46	181
DeiTB224	223	156	44	184
SwinT224	225	158	41	186
SwinS224	220	158	43	184
SwinB224	219	147	45	182

Tabla 9-4: Números de aciertos por clase.

A continuación, se procede a realizar un análisis de las matrices de confusión de los modelos DeiTT224 y SwinT224, ya que fueron los que mejores tasas de precisión obtuvieron por clases. Las matrices de confusión son un tipo de representación de gran utilidad, debido a que nos permiten visualizar cuáles son las clases que confunden en mayor medida un modelo en específico. A través de esta clase de gráficas, se pueden representar métricas tales como la precisión o el número de aciertos. Las matrices de confusión se interpretan de la siguiente forma:

- Las filas hacen referencia a las clases verdaderas de las muestras clasificadas.
- Las columnas simbolizan las clases predichas por el modelo.

En la Figura 9-4, se pueden observar las matrices de confusión generadas, a través de los resultados obtenidos, por el modelo DeiTT224 durante el periodo de testeo. En estas gráficas, se puede apreciar el excelente desempeño demostrado por la red, obteniendo magníficas tasas de precisión en todas las clases. Sorprendentemente, es el único *Transformer* que no presentó una bajada estrepitosa en su nivel de desempeño a la hora de clasificar muestras de

meningiomas. Aun así, esta fue la clase en la cual obtuvo su peor tasa de precisión con un 96 %. El modelo demostró un gran nivel de precisión durante la clasificación de cerebros sanos y tumores pituitarios, logrando un 98 % en ambas clases. La mayoría de fallos cometidos por DeiT224 ocurrieron a la hora de identificar gliomas y meningiomas. La red falló un total de 7 veces en el etiquetado de gliomas, confundiendo este tipo de muestras en su mayoría con meningiomas. Mientras que en la clasificación de meningiomas, el *Transformer* erró hasta 8 veces, etiquetando dichas imágenes incorrectamente en repetidas ocasiones como Gliomas. Por lo tanto, se puede apreciar fácilmente como DeiT224 suele confundir en determinadas situaciones ambas clases.

A través de la Figura 9-5, se pueden visualizar las matrices de confusión generadas por SwinS224 durante la fase de testeo. A diferencia que DeiT224, la tasa de precisión de este modelo durante el etiquetado de las imágenes de meningiomas cayó drásticamente con respecto al resto de clases. Sin embargo, este modelo alcanzó unos elevadísimos niveles de precisión a la hora de detectar cerebros sanos y tumores pituitarios, con un 100 % y un 99 %, respectivamente. En lo referente a las clases que más confundió la red, fueron las imágenes de gliomas y cerebros sanos que la gran mayoría de las ocasiones fueron incorrectamente etiquetadas como meningiomas.

Mediante el análisis de las tasas de exactitud y los números de aciertos por clases, se pudo observar claramente como los *Transformers* tienden a confundir los meningiomas con el resto de clases. El modelo que demostró una mayor calidad a lo largo del proceso de testeo fue DeiT224.

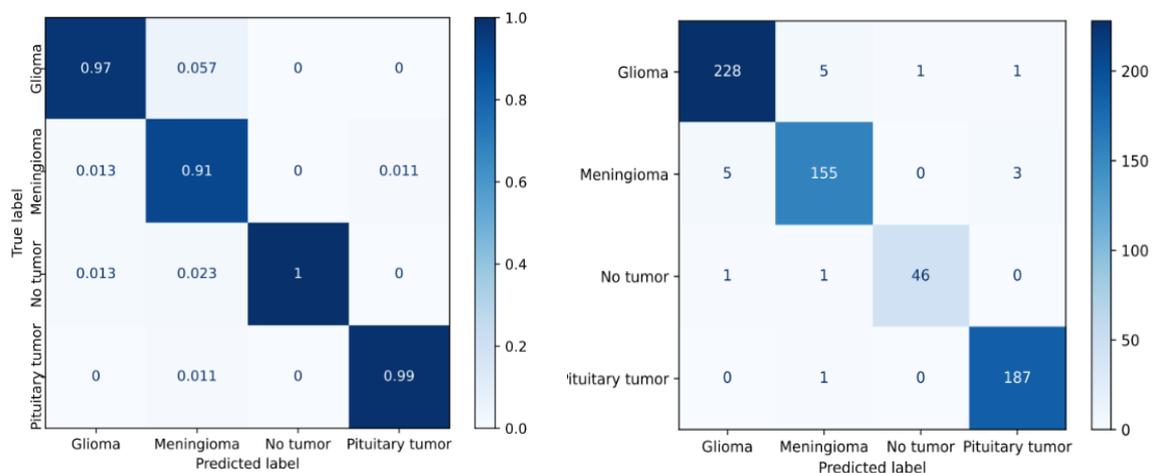


Figura 9-4: Matrices de confusión de DeiT224.

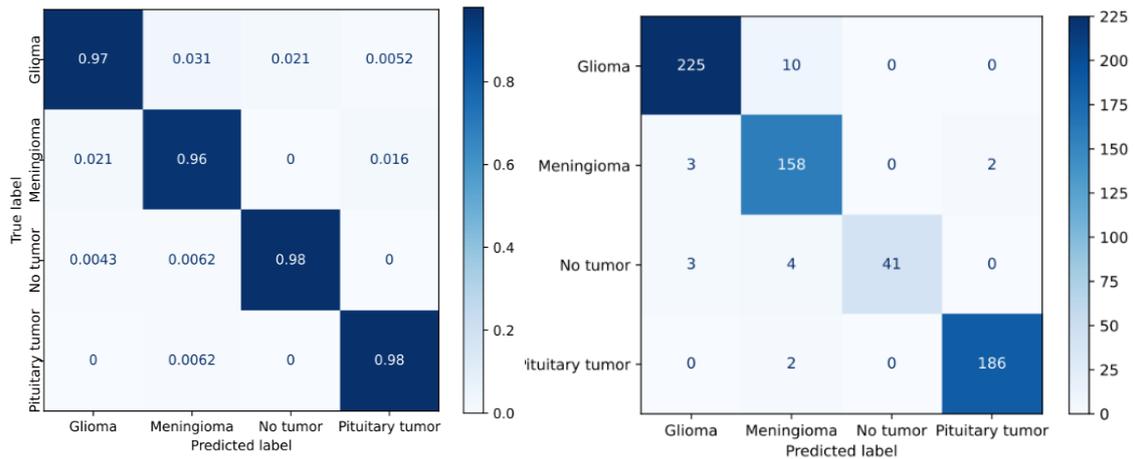


Figura 9-5: Matrices de confusión de SwinS224.

### 9.3. Precision, Recall y F1 Score de los modelos

A continuación, en la Tabla 9-5, se pueden visualizar los resultados obtenidos por los modelos durante la fase de testeo en términos de precisión (*precision*), recuperación (*recall*) y *F1 score*. A nivel de precisión, se puede apreciar como las arquitecturas DeiT y Swin fueron las que mejores cifras lograron. DeiT224 y SwinT224 fueron los modelos más precisos con un 97.22 % y un 96.79 %. En cuanto respecta al *recall*, las redes DeITs fueron las líderes en términos de desempeño. El *Transformer* DeiT224 consiguió una recuperación del 96.85 %. Por últimos, si se tiene en cuenta el *F1 Score*, nuevamente los modelos DeITs fueron los que mejores resultados alcanzaron. Concretamente, DeiT224 fue el que registró la mayor magnitud de *F1 Score* con un 97.03 %. En base a estas tres métricas, podemos concluir que el modelo que cuenta con mayor grado de calidad se trata de DeiT224.

<b>Rendimiento de los Modelos</b>			
<b>Modelos</b>	<b>Precision</b>	<b>Recall</b>	<b>F1 Score</b>
<b>VitB16</b>	89.59 %	92.33 %	90.77 %
<b>VitB32</b>	93.89 %	93.85 %	93.81 %
<b>VitL16</b>	93.52 %	92.53 %	92.82 %
<b>DeiT224</b>	<b>97.22 %</b>	<b>96.85 %</b>	<b>97.03 %</b>
<b>DeiT224</b>	96.08 %	96.08 %	96.03 %
<b>DeiT224</b>	95.87 %	95.03 %	95.39 %
<b>Swin224</b>	96.79 %	94.26 %	95.35 %
<b>Swin224</b>	96.10 %	94.50 %	95.16 %
<b>Swin224</b>	92.73 %	93.48 %	93.09 %

Tabla 9-5: Porcentajes medios de precisión, recuperación y F1 Score.

#### 9.4. Comparación con el estado del arte

Se procede a realizar una comparativa del conjunto de datos empleado con respecto a los utilizados por otros autores. Si se observa la Tabla 9-6, se puede vislumbrar información de gran relevancia en referencia a los distintos conjuntos de datos empleados en otras investigaciones, como las fuentes de las que obtuvieron los datos, los autores, el número de imágenes, el tamaño de las muestras, el tipo de resonancias y las clases.

La mayoría de los autores han empleado para el entrenamiento imágenes MRI provenientes del conjunto de datos de Figshare, ya explicado anteriormente y empleado en este mismo proyecto. Marco Antonio Gómez Guzmán et al. [16] usaron el conjunto de imágenes de mayor tamaño, con un total de 7023 muestras. Las colecciones de datos utilizadas en este proyecto y en el de J. Sánchez et al. [24] fueron las segundas más voluminosas con 6323 secuencias MRI, cada una. Aunque [24] empleó el mismo número de imágenes que este proyecto, los entrenamientos fueron llevados a cabo con los conjuntos de Kaggle y Figshare por separado. Lo que conlleva que el número de muestras usadas en cada entrenamiento sea menor que las utilizadas en el presente proyecto. El modelo propuesto por [24] fue entrenado mediante la colección de Kaggle.

El tamaño de imágenes empleado por cada autor varía bastante, existiendo algunos que solo usan una dimensión y otros que experimentan con varias. Gran parte de los proyectos utilizan muestras de 224x224 o de 256x256. Ali Mohammad Alqudah et al. [18] y Sudhakar Tummala et al. [21] demostraron que el tamaño de las secuencias influye en gran medida en los resultados alcanzados por los modelos. [18] emplearon hasta tres dimensiones diferentes y su modelo propuesto logró las mejores cifras siendo entrenado con muestras de 32x32. En cambio, [21] obtuvo sus mejores resultados usando imágenes de 384x384.

En lo referente al tipo de resonancias, muchos de los autores apostaron por incluir imágenes MRI generados desde distintos planos anatómicos (axial, sagital y coronal) y a partir de diferentes valores de TE y TR (*T1-weighted*, *T2-weighted* y Flair).

Por último, cabe destacar que prácticamente la totalidad de los autores usaron conjuntos de datos con 3 o 4 clases. Aquellos que empleaban un conjunto de datos de 3 clases solo incluían distintos tipos de tumores cerebrales primarios, mientras que otros incluían una cuarta categoría que representaba aquellas imágenes que no contenían ninguna clase de tumor.

Conjuntos de Datos Empleados en cada Proyecto					
Autor	Fuentes	Número de imágenes	Tamaño de imagen	Tipos de resonancia	Clases
Kavin Kumar et al. [13]	Brainweb y Jansons MRI scan center Erobe	134	X	Axial	Cerebros con tumor y sin tumor
Özlem Polat et al. [15]	Figshare	3064	X	<i>T1-weighted</i> , axial, sagital y coronal	Glioma, meningioma y tumor pituitario
Marco Antonio Gómez Guzmán et al. [16]	Figshare, SARTAJ y Br35H	7023	256x256	<i>T1-weighted</i> , axial, sagital y coronal	Glioma, meningioma, tumor pituitario y cerebros sin tumor
Pallavi Tiwari et al. [17]	Kaggle	3264	224x224	<i>T1-weighted</i> , <i>T2-weighted</i> , Flair, axial, sagital y coronal	Glioma, meningioma, tumor pituitario y cerebros sin tumor
Ali Mohammad Alqudah et al. [18]	Figshare	3064	32x32, 64x64 y 128x128	<i>T1-weighted</i> , axial, sagital y coronal	Glioma, meningioma y tumor pituitario
M.Sharma et al. [19]	BRATS2015	330	32x32	<i>T1-weighted</i> , <i>T2-weighted</i> , Flair, sagital	Tumores cerebrales benignos y malignos
Sudhakar Tummala et al. [21]	Figshare	3064	224x224 y 384x384	<i>T1-weighted</i> , axial, sagital y coronal	Glioma, meningioma y tumor pituitario
Md. Azad Hossain et al. [22]	Figshare y BraTS-21	4315	32x32	<i>T1-weighted</i> , <i>T2-weighted</i> , Flair, axial, sagital y coronal	Glioma, meningioma y tumor pituitario
J. Sánchez et al. [24]	Kaggle y Figsare	6342	256x256	<i>T1-weighted</i> , <i>T2-weighted</i> , Flair, axial, sagital y coronal	Glioma, meningioma, tumor pituitario y

					cerebros sin tumor
<b>El presente autor</b>	Kaggle y Figshare	6342	224x224	<i>T1-weighted, T2-weighted, Flair, axial, sagital y coronal</i>	Glioma, meningioma, tumor pituitario y cerebros sin tumor

Tabla 9-6: Comparativa de los conjuntos de datos.

Antes de comenzar con la comparativa a nivel de resultados entre los obtenidos por este trabajo y el estado del arte, cabe destacar que algunas de las investigaciones no utilizaron ciertas métricas empleadas por el presente autor para la evaluación de los modelos, como es el caso de la precisión (*precision*), la recuperación y *F1-score*. Este hecho, junto a la diferencia de fuentes en cuanto a lo que respecta a los datos usados por cada uno de estos para el entrenamiento de las redes, ha dificultado en cierta medida la realización del estudio comparativo. A pesar de estos obstáculos, como se podrá observar a continuación, el análisis ha arrojado una gran cantidad de información de enorme valor.

Como se puede visualizar en la Tabla 9-7, el nivel de precisión (*accuracy*) alcanzado en investigaciones anteriores fue elevadísimo. Pallavi Tiwari et al. [17] y Ali Mohammad Alqudah et al. [18] desarrollaron los modelos más precisos, obteniendo un 99.00 % y un 99.19 %, cada uno. Sin embargo, cabe destacar también el hecho de que estos hayan utilizado conjuntos de datos de menor tamaño que los usados por el presente autor y por Marco Antonio Gómez Guzmán et al. [16]. En comparación con este último autor, la tasa de precisión del 97.16 % del modelo DeiTT224 propuesto en este proyecto, logra superar ligeramente al de la red InceptionV3.

Con respecto a la precisión (*precision*), no todos los proyectos emplearon esta métrica para la evaluación de sus modelos entrenados. Entre las investigaciones que, si utilizaron esta medida, destacó [18] logrando alcanzar un 99.16 % de precisión. Aunque las redes propuestas del actual proyecto hayan mostrado unos ligeros peores niveles de desempeño, obtuvieron unos muy buenos resultados con un 97.22 % y un 96.79 % de precisión.

Las métricas menos empleadas dentro del estado del arte para la evaluación de los modelos fueron la recuperación y el *F1-Score*. A pesar de esto, algunos autores sí registraron las cifras, en términos de ambas métricas, alcanzadas por sus modelos. Pallavi Tiwari et al. [17] obtuvieron los mejores resultados con un 99 % de recuperación y de *F1-score*. El modelo EfficientNetB0 entrenado por J. Sánchez et al. [24] alcanzó también un muy buen nivel de desempeño obteniendo un 98 % de *recall* y un 98.05 % de *F1 Score*. DeiT224 logró superar a la red propuesta por Marco Antonio Gómez Guzmán et al. [16] en términos de recuperación con un 96.85 %. Sin embargo, este último superó el valor de *F1-score* logrado por DeiT224 con un 97.28 %.

Resultados de Proyectos Anteriores					
Autor	Modelo propuesto	Accuracy	Precision	Recall	F1 Score
Kavin Kumar et al. [13]	Combinación de MMTH y MTMD + SVM	95.00 %	X	X	X
Özlem Polat et al. [15]	Pre-trained ResNet50 + Optimizador Adam	99.02 %	X	X	X
Marco Antonio Gómez Guzmán et al. [16]	Pre-trained InceptionV3	97.12 %	97.97 %	96.59 %	97.28 %
Pallavi Tiwari et al. [17]	Custom CNN	99.00 %	99.00 %	<b>99.00 %</b>	<b>99.00 %</b>
Ali Mohammad Alqudah et al. [18]	Custom CNN	<b>99.19 %</b>	<b>99.16 %</b>	X	X
M.Sharma et al. [19]	CNN + SVM	98.50 %	X	X	X
Sudhakar Tummala et al. [21]	Ensemble of ViTs	98.70 %	X	X	X
Md. Azad Hossain et al. [22]	LCDeiT	98.73 %	97.87 %	97.94 %	97.86 %
J. Sánchez et al. [24]	EfficientNetB0	98.40 %	98.10 %	98.00 %	98.05 %
El presente autor	Pre-trained DeiT224	97.16 %	97.22 %	96.85 %	97.03 %
El presente autor	Pre-trained SwinT224	96.21 %	96.79 %	94.26 %	95.35 %

Tabla 9-7: Comparación de resultados con respecto a otros autores.

## 9.5. Aportaciones científicas

Una vez finalizado el proceso de estudio y evaluación de los resultados obtenidos, así como la comparativa llevada a cabo con respecto a trabajos anteriores, se procede a explicar las aportaciones científicas generadas por este proyecto.

En primer lugar, el autor quiere remarcar que los modelos propuestos tienen un interés meramente de carácter investigativo, por lo que estos no deben ser utilizados por neurólogos u otros especialistas como ayuda para la realización de diagnósticos de tumores cerebrales primarios. El desarrollo de una aplicación basada en este tipo de técnicas de *Deep Learning*, debería conllevar un estudio mucho más profundo e intenso para, de esta manera, poder sacar unas conclusiones más precisas sobre el grado de viabilidad que tiene su implementación y su efectividad a la hora de detectar tumores cerebrales. Asimismo, haría falta elaborar un conjunto de muestras mucho más amplio y diverso, a partir de imágenes MRI de miles de pacientes, para

evitar descompensaciones en lo que respecta al número de imágenes por clase y, de esta manera, garantizar la generalización del problema abarcado por parte de los modelos

Este proyecto aporta información de gran relevancia para futuros trabajos científicos de la misma índole, ya que ha demostrado que los *Transformers* en términos de rendimiento pueden equiparar e incluso igualar algunas redes CNN, las cuales han acaparado el paradigma de la clasificación de imágenes a lo largo de esta última década. Del mismo modo, se ha podido apreciar que a pesar de que estos hayan sido entrenados con un conjunto de datos bastante voluminoso, no se han quedado atrás en lo que se refiere a sus niveles de rendimiento con respecto a las propuestas de otros proyectos.

Por último, el autor quiere destacar el margen de mejora observado en el desempeño de ciertos modelos, los cuales con una selección de hiperparámetros algo más precisa, podrían superar incluso las cifras registradas por aquellas redes propuestas de investigaciones anteriores que han logrado los mayores niveles de calidad.

## Capítulo 10. Conclusiones

Los tumores cerebrales causan cientos de miles de muertos cada año. Esta enfermedad no es solo temible por su enorme porcentaje de mortalidad, sino que también por las severas secuelas que pueden afectar a los pacientes que sufren esta afección. Por lo tanto, es de suma importancia el hecho de detectar este tipo de tumores en etapas tempranas para, de esta manera, evitar su evolución mediante la aplicación de una serie de tratamientos específicos seleccionados por el equipo médico. El desarrollo de herramientas rápidas, efectivas y eficaces que permitan automatizar el proceso de diagnóstico puede ser de gran ayuda para los especialistas, ya que reducirían la carga de trabajo y la cantidad de errores cometidos por estos.

Este proyecto ha logrado desarrollar una serie de modelos, basados en la topología *Transformer*, que han demostrado un excelente desempeño durante la clasificación de distintos tipos de tumores cerebrales primarios a partir de imágenes generadas por resonancias magnéticas. Asimismo, estos son capaces de discernir cuando un paciente no padece la enfermedad. La obtención de estas redes fue llevada a cabo mediante un proceso largo y exhaustivo en el que se buscó maximizar sus niveles de desempeño a lo largo de los entrenamientos. Finalmente, se efectuó una evaluación profunda de los resultados en la que se determinó que los modelos DeiT224 y SwinT224 fueron los que presentaban un mayor grado de calidad en base a ciertas métricas, como la precisión (*accuracy*). En lo que respecta a esta medida, los *Transformers* fueron capaces de obtener porcentajes de un 97.16 % y un 96.21 % durante el periodo de testeo.

A pesar de que los modelos propuestos no puedan ser empleados como ayuda en la detección de tumores cerebrales de manera profesional, este proyecto aporta información de enorme relevancia para el desarrollo de proyectos de la misma temática. Por lo que, en base a este, futuras investigaciones que indaguen en mayor medida sobre la viabilidad del uso de estas técnicas de aprendizaje profundo podrían desarrollar aplicaciones con la capacidad de precisión, velocidad y eficacia necesarias para ser implementadas en un gran número de centros médicos y que puedan empleadas por los equipos médicos para facilitar las tareas de diagnóstico. La creación de este tipo de herramientas tendría un gran impacto en la sociedad, ya que contarían con la capacidad de poder salvarle la vida a millones de personas en todo el mundo.

# Anexo: competencias específicas

En esta sección, se procede a comentar las competencias específicas que han sido cubiertas durante la elaboración de este proyecto, las cuales son abordadas con un gran nivel de profundidad en [108] y son las siguientes:

TFG: *“Ejercicio original a realizar individualmente y presentar y defender ante un tribunal universitario, consistente en un proyecto en el ámbito de las tecnologías específicas de la Ingeniería en Informática de naturaleza profesional en el que se sinteticen e integren las competencias adquiridas en las enseñanzas”*. El proyecto de aprendizaje profundo llevado a cabo, cuya idea es original, se ha efectuado individualmente. El objetivo principal del trabajo es la realización de una exposición y defensa delante de un tribunal, en el que se demuestre los conocimientos adquiridos e integrados a lo largo de los estudios por parte del autor.

CI7: *“Conocimiento, diseño y utilización de forma eficiente de los tipos y estructuras de datos más adecuados a la resolución de un problema”*. Para la elaboración del proyecto, se ha tenido que realizar primero un proceso de documentación y aprendizaje previo a la creación del conjunto de datos utilizado, para conocer y descubrir el tipo de muestras que se deben usar, el tamaño de imagen más adecuado y hallar la estructura más adecuada para la colección de muestras elaborada. El autor ha tenido que llevar a cabo una exploración exhaustiva del estado del arte, con el fin de conocer los conjuntos de datos más empleados dentro del paradigma de los clasificadores de tumores cerebrales.

CI15: *“Conocimiento y aplicación de los principios fundamentales y técnicas básicas de los sistemas inteligentes y su aplicación práctica”*. El autor ha efectuado una búsqueda, análisis, y comprensión profunda de numerosos conocimientos relacionados con el sector del aprendizaje profundo. Asimismo, se ha ejecutado una investigación intensiva para encontrar las mejores herramientas para la implementación y entrenamiento de los diferentes *Transformers*, así como las métricas más idóneas para la evaluación de los resultados obtenidos por los modelos.

# Bibliografía

- [1] «¿Qué es la visión artificial?», *IBM*. Disponible en: <https://www.ibm.com/es-es/topics/computer-vision>. [Accedido: 23 de febrero de 2023]
- [2] «Tumor cerebral: Estadísticas», *American Society of Clinical Oncology*, febrero de 2022. Disponible en: <https://www.cancer.net/es/tipos-de-cancer/tumor-cerebral/estadisticas>. [Accedido: 23 de febrero de 2023]
- [3] «The Evolution of AI in Healthcare», *Xsolis*, 2 de febrero de 2021. Disponible en: <https://www.xsolis.com/blog/the-evolution-of-ai-in-healthcare>. [Accedido: 27 de febrero de 2023]
- [4] Sergio Sancho Azcoitia, «MYCIN, El comienzo de la Inteligencia Artificial en el mundo de la medicina», *Telefónica Tech*, 15 de noviembre de 2018. Disponible en: <https://empresas.blogthinkbig.com/mycin-el-comienzo-de-la-inteligencia/>. [Accedido: 27 de febrero de 2023]
- [5] V. Kaul, S. Enslin, y S. A. Gross, «History of artificial intelligence in medicine», *Gastrointest Endosc*, vol. 92, n.º 4, pp. 807-812, oct. 2020, doi: 10.1016/j.gie.2020.06.040. Disponible en: [https://www.giejournal.org/article/S0016-5107\(20\)34466-7/fulltext](https://www.giejournal.org/article/S0016-5107(20)34466-7/fulltext).
- [6] «DXplain», *Wikipedia*. Disponible en: <https://en.wikipedia.org/wiki/DXplain>. [Accedido: 27 de febrero de 2023]
- [7] «IBM Begins Development of Watson, the First Cognitive Computer», *Jeremy Norman's HistoryofInformation*, 9 de septiembre de 2013. Disponible en: <https://www.historyofinformation.com/detail.php?id=2830>. [Accedido: 28 de febrero de 2023]
- [8] Ed Burns, «IBM Watson supercomputer», *TechTarget*, enero de 2023. Disponible en: <https://www.techtarget.com/searchenterpriseai/definition/IBM-Watson-supercomputer>. [Accedido: 28 de febrero de 2023]
- [9] Benilda Eleonor V. Comendador, Bien Michael B. Francisco, Jefferson S. Medinilla, Sharleen Mae T. Nacion, y Timothy Bryle E. Serac, «Pharmabot: A Pediatric Generic Medicine Consultant Chatbot», *Journal of Automation and Control Engineering*, vol. 3, n.º 2, pp. 137-140, 2015, doi: 10.12720/joace.3.2.137-140
- [10] Bernard Marr, «First FDA Approval For Clinical Cloud-Based Deep Learning In Healthcare», *Forbes*, 20 de enero de 2017. Disponible en: <https://www.forbes.com/sites/bernardmarr/2017/01/20/first-fda-approval-for-clinical-cloud-based-deep-learning-in-healthcare/>. [Accedido: 28 de febrero de 2023]
- [11] Adam Conner-Simons y Rachel Gordon, «Using AI to predict breast cancer and personalize care», *MIT News*, 7 de mayo de 2019. Disponible en: <https://news.mit.edu/2019/using-ai-predict-breast-cancer-and-personalize-care-0507>. [Accedido: 28 de febrero de 2023]

- [12] Jordi Laguarda, Ferran Hueto, y Brian Subirana, «COVID-19 Artificial Intelligence Diagnosis using only Cough Recordings», *IEEE EMBS*, 30 de septiembre de 2020. Disponible en: <https://www.embs.org/ojemb/articles/covid-19-artificial-intelligence-diagnosis-using-only-cough-recordings/>. [Accedido: 28 de febrero de 2023]
- [13] Kavin Kumar K., Meera Devi T., y Maheswaran S., «An Efficient Method for Brain Tumor Detection Using Texture Features and SVM Classifier in MR Images.», *Asian Pac J Cancer Prev*, vol. 19, n.º 10, pp. 2789-2794, oct. 2018, doi: 10.22034/APJCP.2018.19.10.2789
- [14] Albers Uzila, «5 Popular CNN Architectures Clearly Explained and Visualized», *Medium*, 22 de agosto de 2022. Disponible en: <https://towardsdatascience.com/5-most-well-known-cnn-architectures-visualized-af76f1f0065e>. [Accedido: 4 de marzo de 2023]
- [15] Ö. Polat y C. Güngen, «Classification of brain tumors from MR images using deep transfer learning», *J Supercomput*, vol. 77, n.º 7, pp. 7236-7252, jul. 2021, doi: 10.1007/s11227-020-03572-9
- [16] M. A. Gómez-Guzmán *et al.*, «Classifying Brain Tumors on Magnetic Resonance Imaging by Using Convolutional Neural Networks», *Electronics (Basel)*, vol. 12, n.º 4, p. 955, feb. 2023, doi: 10.3390/electronics12040955
- [17] P. Tiwari *et al.*, «CNN Based Multiclass Brain Tumor Detection Using Medical Imaging», *Comput Intell Neurosci*, vol. 2022, pp. 1-8, jun. 2022, doi: 10.1155/2022/1830010
- [18] A. M. Alqudah, «Brain Tumor Classification Using Deep Learning Technique - A Comparison between Cropped, Uncropped, and Segmented Lesion Images with Different Sizes», *International Journal of Advanced Trends in Computer Science and Engineering*, vol. 8, n.º 6, pp. 3684-3691, dic. 2019, doi: 10.30534/ijatcse/2019/155862019
- [19] M. O. Khairandish, M. Sharma, V. Jain, J. M. Chatterjee, y N. Z. Jhanjhi, «A Hybrid CNN-SVM Threshold Segmentation Approach for Tumor Detection and Classification of MRI Brain Images», *IRBM*, vol. 43, n.º 4, pp. 290-299, ago. 2022, doi: 10.1016/j.irbm.2021.06.003
- [20] A. Dosovitskiy *et al.*, «An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale», oct. 2020.
- [21] S. Tummala, S. Kadry, S. A. C. Bukhari, y H. T. Rauf, «Classification of Brain Tumor from Magnetic Resonance Imaging Using Vision Transformers Ensembling.», *Curr Oncol*, vol. 29, n.º 10, pp. 7498-7511, oct. 2022, doi: 10.3390/curroncol29100590
- [22] G. J. Ferdous, K. A. Sathi, Md. A. Hossain, M. M. Hoque, y M. A. A. Dewan, «LCDEiT: A Linear Complexity Data-Efficient Image Transformer for MRI Brain Tumor Classification», *IEEE Access*, vol. 11, pp. 20337-20350, 2023, doi: 10.1109/ACCESS.2023.3244228
- [23] D. Reyes y J. Sánchez, «Performance of Convolutional Neural Networks for the Classification of Brain Tumors using Magnetic Resonance Images», *Preprint*, 2023.
- [24] J. Sánchez y J. M. Medina, «High Accuracy Brain Tumor Classification with EfficientNet and Magnetic Resonance Images», *In Proc. 5th International Conference on Advances in Signal Processing and Artificial Intelligence (ASPAI)*, pp. 151-156, 2023.

- [25] «Qué es SCRUM», *proyectos ágiles*. Disponible en: <https://proyectosagiles.org/que-es-scrum/>. [Accedido: 12 de marzo de 2023]
- [26] Julio Roche, «Scrum: roles y responsabilidades», *Deloitte*. Disponible en: <https://www2.deloitte.com/es/es/pages/technology/articles/roles-y-responsabilidades-scrum.html>. [Accedido: 12 de marzo de 2023]
- [27] María M. López y Andrés F. Cardona, «HISTORIA DEL CÁNCER Y EL CÁNCER EN LA HISTORIA», *revistamedica*, dic. 2020, Disponible en: <https://revistamedicina.net/index.php/Medicina/article/download/1559/1982?inline=1>. [Accedido: 16 de marzo de 2023]
- [28] «Las cifras del cáncer en España», 2022. Disponible en: [https://seom.org/images/LAS\\_CIFRAS\\_DEL\\_CANCER\\_EN\\_ESPANA\\_2022.pdf](https://seom.org/images/LAS_CIFRAS_DEL_CANCER_EN_ESPANA_2022.pdf). [Accedido: 16 de marzo de 2023]
- [29] Javier Puente y Guillermo de Velasco, «¿Qué es el cáncer y cómo se desarrolla?», *SEOM*, 16 de diciembre de 2019. Disponible en: <https://seom.org/informacion-sobre-el-cancer/que-es-el-cancer-y-como-se-desarrolla>. [Accedido: 16 de marzo de 2023]
- [30] «Tumor», *MedlinePlus*. Disponible en: <https://medlineplus.gov/spanish/ency/article/001310.htm>. [Accedido: 16 de marzo de 2023]
- [31] «¿Qué es un Tumor Cerebral?», *Clinic Barcelona*, 20 de febrero de 2018. Disponible en: <https://www.clinicbarcelona.org/asistencia/enfermedades/tumor-cerebral>. [Accedido: 17 de marzo de 2023]
- [32] L. E. Contreras, «EPIDEMIOLOGÍA DE TUMORES CEREBRALES», *Revista Médica Clínica Las Condes*, vol. 28, n.º 3, pp. 332-338, may 2017, doi: 10.1016/j.rmclc.2017.05.001
- [33] «Tumor cerebral: Introducción», *American Society of Clinical Oncology*, septiembre de 2021. Disponible en: <https://www.cancer.net/es/tipos-de-cancer/tumor-cerebral/introducci%C3%B3n>. [Accedido: 17 de marzo de 2023]
- [34] Marta Guerri, «Las células gliales (neuroglia) del sistema nervioso: Qué son, tipos y funciones», *PsicoActiva*. Disponible en: <https://www.psicoactiva.com/blog/las-celulas-gliales-tipos-funciones/>. [Accedido: 17 de marzo de 2023]
- [35] «Meninges», *Mayo Clinic*, 5 de octubre de 2018. Disponible en: <https://www.mayoclinic.org/es-es/diseases-conditions/meningioma/multimedia/meninges/img-20008665#:~:text=Existen%20tres%20capas%20de%20membranas,capa%20externa%20es%20denominada%20duramadre>. [Accedido: 17 de marzo de 2023]
- [36] «Tumores pituitarios», *Mayo Clinic*, 8 de febrero de 2023. Disponible en: <https://www.mayoclinic.org/es-es/diseases-conditions/pituitary-tumors/symptoms-causes/syc-20350548>. [Accedido: 17 de marzo de 2023]

- [37] «Glioma difuso de línea media», *NIH*, 26 de marzo de 2020. Disponible en: <https://www.cancer.gov/rare-brain-spine-tumor/espanol/tumores/gliomas-difuso-linea-media>. [Accedido: 17 de marzo de 2023]
- [38] «Tumores de Hipófisis», *DEI*, 21 de enero de 2015. Disponible en: <https://www.deidiagnostico.com/tumores-de-hipofisis/>. [Accedido: 17 de marzo de 2023]
- [39] «Tumor cerebral: Diagnóstico», septiembre de 2021. Disponible en: <https://www.cancer.net/es/tipos-de-c%C3%A1ncer/tumor-cerebral/diagn%C3%B3stico>. [Accedido: 18 de marzo de 2023]
- [40] «¿Cómo se diagnostica un tumor cerebral?», *Dacer*. Disponible en: <https://www.dacer.org/como-se-diagnostica-un-tumor-cerebral/>. [Accedido: 18 de marzo de 2023]
- [41] «Tumor cerebral», *Mayo Clinic*. Disponible en: <https://www.mayoclinic.org/es-es/diseases-conditions/brain-tumor/diagnosis-treatment/drc-20350088>. [Accedido: 18 de marzo de 2023]
- [42] «Diagnóstico», *GEINO*. Disponible en: <https://geino.es/diagnostico/>. [Accedido: 18 de marzo de 2023]
- [43] «Resonancia magnética», *Mayo Clinic*. Disponible en: <https://www.mayoclinic.org/es-es/tests-procedures/mri/about/pac-20384768>. [Accedido: 18 de marzo de 2023]
- [44] Eva Experts, «Resonancia magnética T1 y T2, imágenes potenciadas TR y TE», *Evacenter*, 17 de agosto de 2021. Disponible en: <https://evacenter.com/blog/conceptos-basicos-de-radiologia-en-resonancia-magnetica/>. [Accedido: 19 de marzo de 2023]
- [45] Úrsula Alcañas, «Resonancia Magnética, los protocolos y su práctica.», *Protocolos resonancia magnetica*, 25 de enero de 2015. Disponible en: <https://protocolosresonanciamagnetica.org/2015/01/25/planos-anatomicos/>. [Accedido: 19 de marzo de 2023]
- [46] «Corte Sagital De Cabeza», *Anatomía Humana 3D*. Disponible en: <https://anatomiahumana3d.com/muestras3d/cabeza-y-cuello,sistema-nervioso-central/corte-sagital-de-cabeza/>. [Accedido: 19 de marzo de 2023]
- [47] «Tiempo de eco», *Clínica Universitaria de Navarra*. Disponible en: <https://www.cun.es/diccionario-medico/terminos/tiempo-eco#:~:text=El%20tiempo%20de%20eco%2C%20tambi%C3%A9n,y%20tejidos%20internos%20del%20cuerpo>. [Accedido: 19 de marzo de 2023]
- [48] Sonia Claret Loaiza y Víctor Federico Cáceres Filippon, «PRINCIPIOS BÁSICOS DE RM: LO QUE TODO RADIOLOGO DEBE CONOCER PARA SU PRÁCTICA DIARIA». Disponible en: [https://www.piper.espacioseram.com/index.php/seram/article/download/2659/1299/#:~:text=%2D%20Tiempo%20de%20repetici%C3%B3n%20\(TR\)%3A,%20secuencias%20mixtas%20\(DP\)](https://www.piper.espacioseram.com/index.php/seram/article/download/2659/1299/#:~:text=%2D%20Tiempo%20de%20repetici%C3%B3n%20(TR)%3A,%20secuencias%20mixtas%20(DP)). [Accedido: 19 de marzo de 2023]

- [49] David C. Preston, «Magnetic Resonance Imaging (MRI) of the Brain and Spine: Basics», *Case*, 7 de abril de 2016. Disponible en: <https://case.edu/med/neurology/NR/MRI%20Basics.htm>. [Accedido: 20 de marzo de 2023]
- [50] Yahya Baba, «T1 weighted image», *Radiopaedia*, 27 de febrero de 2023. Disponible en: <https://radiopaedia.org/articles/t1-weighted-image>. [Accedido: 20 de marzo de 2023]
- [51] «Imágenes por resonancia magnética de recuperación de la inversión atenuada de fluido para detectar lesiones corticales y yuxtacorticales de esclerosis múltiple», *IntraMed*. Disponible en: <https://www.intramed.net/contenidover.asp?contenidoid=15106>. [Accedido: 20 de marzo de 2023]
- [52] «El tutorial de Python», *Python Software Foundation*. Disponible en: <https://docs.python.org/es/3/tutorial/>. [Accedido: 16 de abril de 2023]
- [53] «¿Qué es Python?», *Amazon Web Services*. Disponible en: <https://aws.amazon.com/es/what-is/python/>. [Accedido: 16 de abril de 2023]
- [54] «¿Qué es Machine Learning? Cómo, dónde y cuándo», *KeepCoding*, 17 de abril de 2023. Disponible en: <https://keepcoding.io/blog/que-es-machine-learning/>. [Accedido: 17 de abril de 2023]
- [55] «Preguntas frecuentes», *Google*. Disponible en: <https://research.google.com/colaboratory/intl/es/faq.html>. [Accedido: 17 de abril de 2023]
- [56] «Acceso sencillo y seguro a tu contenido», *Google Drive*. Disponible en: [https://www.google.com/intl/es\\_es/drive/](https://www.google.com/intl/es_es/drive/). [Accedido: 17 de abril de 2023]
- [57] «¿Para qué sirve TensorFlow?», *KeepCoding*, 20 de marzo de 2023. Disponible en: <https://keepcoding.io/blog/para-que-sirve-tensorflow/>. [Accedido: 17 de abril de 2023]
- [58] «Keras», *TensorFlow*, 24 de enero de 2022. Disponible en: <https://www.tensorflow.org/guide/keras?hl=es-419>. [Accedido: 17 de abril de 2023]
- [59] «La librería Numpy», *Aprende con Alf*, 12 de mayo de 2022. Disponible en: <https://aprendeconalf.es/docencia/python/manual/numpy/>. [Accedido: 19 de abril de 2023]
- [60] Neema Muganga, «Una introducción a Matplotlib en Python», *GeekFlare*, 4 de octubre de 2022. Disponible en: <https://geekflare.com/es/matplotlib-in-python/>. [Accedido: 19 de abril de 2023]
- [61] «Scikit-learn Machine Learning in Python», *Scikit Learn*. Disponible en: <https://scikit-learn.org/stable/>. [Accedido: 19 de abril de 2023]
- [62] «ViT-Keras», *Github*, 20 de abril de 2023. Disponible en: <https://github.com/faustomorales/vit-keras>. [Accedido: 19 de abril de 2023]

- [63] «TensorFlow Hub es un repositorio de modelos de aprendizaje automático entrenados.», *Tensor Flow*. Disponible en: <https://www.tensorflow.org/hub?hl=es-419>. [Accedido: 20 de abril de 2023]
- [64] Sayak Paul, «DeiT», *TensorFlow Hub*. Disponible en: <https://tfhub.dev/sayakpaul/collections/deit/1>. [Accedido: 20 de abril de 2023]
- [65] «Tfswin», *Github*. Disponible en: <https://github.com/shkarupa-alex/tfswin>. [Accedido: 20 de abril de 2023]
- [66] «Os — Interfaces misceláneas del sistema operativo», *Python Software Foundation*. Disponible en: <https://docs.python.org/es/3.10/library/os.html>. [Accedido: 20 de abril de 2023]
- [67] Riley F. Edmunds y Lev Maximov, «Import-ipynb 0.1.4», *PyPi*, 22 de abril de 2022. Disponible en: <https://pypi.org/project/import-ipynb/#description>. [Accedido: 20 de abril de 2023]
- [68] «Figshare», *Wikipedia*. Disponible en: <https://es.wikipedia.org/wiki/Figshare>. [Accedido: 30 de abril de 2023]
- [69] «The all in one data and paper institutional repository», *Digital Science*. Disponible en: <https://www.digital-science.com/product/figshare/>. [Accedido: 30 de abril de 2023]
- [70] Jun Cheng, «Brain tumor dataset», *Figshare*, 2 de abril de 2017. Disponible en: [https://figshare.com/articles/dataset/brain\\_tumor\\_dataset/1512427](https://figshare.com/articles/dataset/brain_tumor_dataset/1512427). [Accedido: 30 de abril de 2023]
- [71] «¿Qué es Kaggle?», *KeepCoding*, 14 de septiembre de 2022. Disponible en: <https://keepcoding.io/blog/que-es-kaggle/>. [Accedido: 1 de mayo de 2023]
- [72] Ashwini Swain, «Novice to Grandmaster», *Kaggle*, 2017. Disponible en: <https://www.kaggle.com/code/ash316/novice-to-grandmaster>. [Accedido: 1 de mayo de 2023]
- [73] Maximiliano Pona, «¿Qué son las competencias de Kaggle?», *Masel Data*, 26 de octubre de 2022. Disponible en: <https://www.maseldata.com/post/que-es-kaggle>. [Accedido: 1 de mayo de 2023]
- [74] Sartaj Bhuvaji, Ankita Kadam, Prajakta Bhumkar, Sameer Dedge, Navoneel Chakrabarty, y Swati Kanchan, «Brain Tumor Classification (MRI)», *Kaggle*. Disponible en: <https://www.kaggle.com/datasets/sartajbhuvaji/brain-tumor-classification-mri>. [Accedido: 1 de mayo de 2023]
- [75] «DeiT: Data-efficient Image Transformers», *Github*. Disponible en: [https://github.com/facebookresearch/deit/blob/main/README\\_deit.md](https://github.com/facebookresearch/deit/blob/main/README_deit.md). [Accedido: 6 de mayo de 2023]
- [76] «Vision Transformer and MLP-Mixer Architectures», *Vision Transformer and MLP-Mixer Architectures*. Disponible en: [https://github.com/google-research/vision\\_transformer](https://github.com/google-research/vision_transformer). [Accedido: 6 de mayo de 2023]

- [77] «Swin Transformer», *Github*. Disponible en: <https://github.com/microsoft/Swin-Transformer>. [Accedido: 6 de mayo de 2023]
- [78] A. Vaswani *et al.*, «Attention Is All You Need», jun. 2017.
- [79] Rick Merrit, «¿Qué es un Modelo Transformer?», *Nvidia*, 19 de abril de 2022. Disponible en: <https://la.blogs.nvidia.com/2022/04/19/que-es-un-modelo-transformer/>. [Accedido: 8 de mayo de 2023]
- [80] Cristian Santander, «Transformers: Redes Neuronales», *LinkedIn*, 14 de abril de 2021. Disponible en: <https://www.linkedin.com/pulse/transformers-redes-neuronales-cristian-santander/?originalSubdomain=es>. [Accedido: 8 de mayo de 2023]
- [81] «Introducing ChatGPT», *Introducing ChatGPT*, 30 de noviembre de 2022. Disponible en: <https://openai.com/blog/chatgpt#OpenAI>. [Accedido: 8 de mayo de 2023]
- [82] «Transformer's Encoder-Decoder: Let's Understand The Model Architecture», *KiKaBeN*, 13 de diciembre de 2021. Disponible en: <https://kikaben.com/transformers-encoder-decoder/>. [Accedido: 8 de mayo de 2023]
- [83] M. Raghu, T. Unterthiner, S. Kornblith, C. Zhang, y A. Dosovitskiy, «Do Vision Transformers See Like Convolutional Neural Networks?», ago. 2021.
- [84] Nikolas Adaloglou, «How the Vision Transformer (ViT) works in 10 minutes: an image is worth 16x16 words», *AI Summer*, 28 de enero de 2021. Disponible en: <https://theaisummer.com/vision-transformer/>. [Accedido: 9 de mayo de 2023]
- [85] Deval Shah, «Vision Transformer: What It Is & How It Works [2023 Guide]», *V7*, 15 de diciembre de 2022. Disponible en: <https://www.v7labs.com/blog/vision-transformer-guide>. [Accedido: 9 de mayo de 2023]
- [86] H. Touvron, M. Cord, M. Douze, F. Massa, A. Sablayrolles, y H. Jégou, «Training data-efficient image transformers & distillation through attention», dic. 2020.
- [87] Hugo Touvron, Matthijs Douze, Francisco Massa, Alex Sablayrolles, y Hervé Jégou, «Data-efficient image Transformers: A promising new technique for image classification», *Meta AI*, 23 de diciembre de 2020. Disponible en: <https://ai.facebook.com/blog/data-efficient-image-transformers-a-promising-new-technique-for-image-classification/>. [Accedido: 10 de mayo de 2023]
- [88] Sik-Ho Tsang, «Review: Data Efficient Image Transformer (DeiT)», *Medium*, 11 de febrero de 2022. Disponible en: <https://sh-tsang.medium.com/review-deit-data-efficient-image-transformer-b5b6ee5357d0>. [Accedido: 10 de mayo de 2023]
- [89] Ze Liu *et al.*, «Swin Transformer: Hierarchical Vision Transformer using Shifted Windows», *Microsoft Research Asia*, ago. 2021.
- [90] James Loy, «A Comprehensive Guide to Microsoft's Swin Transformer», *Medium*, 20 de mayo de 2022. Disponible en: <https://towardsdatascience.com/a-comprehensive-guide-to-swin-transformer-64965f89d14c>. [Accedido: 11 de mayo de 2023]

- [91] Koushik S., «Swin Transformer is all you need for Computer Vision», *Medium*, 23 de octubre de 2021. Disponible en: <https://koushik0901.medium.com/swin-transformer-is-all-you-need-for-computer-vision-f2763a4f3fed>. [Accedido: 11 de mayo de 2023]
- [92] «¿Qué es el ajuste de hiperparámetros?», *Amazon Web Services*. Disponible en: <https://aws.amazon.com/es/what-is/hyperparameter-tuning/>. [Accedido: 14 de mayo de 2023]
- [93] Jose Manuel Casas, «Tuneando los hiperparámetros de una red neuronal LSTM para obtener un aprendizaje más eficiente», *LinkedIn*, 17 de junio de 2020. Disponible en: <https://www.linkedin.com/pulse/tuneando-los-hiperpar%C3%A1metros-de-una-red-neuronal-lstm-casas-gonzalez/?originalSubdomain=es>. [Accedido: 14 de mayo de 2023]
- [94] Luis Velasco, «Optimizadores en redes neuronales profundas: un enfoque práctico», *Optimizadores en redes neuronales profundas: un enfoque práctico*, 26 de abril de 2020. Disponible en: <https://velascoluis.medium.com/optimizadores-en-redes-neuronales-profundas-un-enfoque-pr%C3%A1ctico-819b39a3eb5>. [Accedido: 14 de mayo de 2023]
- [95] Jason Brownlee, «Gentle Introduction to the Adam Optimization Algorithm for Deep Learning», *Machine Learning Mastery*, 3 de julio de 2017. Disponible en: <https://machinelearningmastery.com/adam-optimization-algorithm-for-deep-learning/>. [Accedido: 14 de mayo de 2023]
- [96] Akash Ajagekar, «Adam», *Cornell University*, 16 de diciembre de 2021.
- [97] Sukanya Bag, «Activation Functions — All You Need To Know!», *Medium*, 13 de febrero de 2021. Disponible en: <https://medium.com/analytics-vidhya/activation-functions-all-you-need-to-know-355a850d025e>. [Accedido: 14 de mayo de 2023]
- [98] «ReLU», *DeepAI*. Disponible en: <https://deepai.org/machine-learning-glossary-and-terms/relu>. [Accedido: 14 de mayo de 2023]
- [99] Thomas Wood, «Softmax Function», *DeepAI*. Disponible en: <https://deepai.org/machine-learning-glossary-and-terms/softmax-layer>. [Accedido: 14 de mayo de 2023]
- [100] «Understanding Categorical Cross-Entropy Loss, Binary Cross-Entropy Loss, Softmax Loss, Logistic Loss, Focal Loss and all those confusing names», *Github*, 23 de mayo de 2018. Disponible en: [https://gombru.github.io/2018/05/23/cross\\_entropy\\_loss/](https://gombru.github.io/2018/05/23/cross_entropy_loss/). [Accedido: 14 de mayo de 2023]
- [101] «Categorical cross-entropy loss — The most important loss function», *Medium*, 2 de diciembre de 2022. Disponible en: <https://neuralthreads.medium.com/categorical-cross-entropy-loss-the-most-important-loss-function-d3792151d05b>. [Accedido: 14 de mayo de 2023]
- [102] «Confusion matrix», *Wikipedia*. Disponible en: [https://en.wikipedia.org/wiki/Confusion\\_matrix](https://en.wikipedia.org/wiki/Confusion_matrix). [Accedido: 15 de mayo de 2023]
- [103] Ahmed Fawzy Gad, «Evaluating Deep Learning Models: The Confusion Matrix, Accuracy, Precision, and Recall», *Paperspace*, 2019. Disponible en: <https://blog.paperspace.com/deep-learning-metrics-precision-recall->

