ULPGC
Universidad de
Las Palmas de
Gran Canaria

ESCUELA DE INGENIERÍA
INFORMÁTICA

Trabajo Fin de Grado

Grado en Ingeniería Informática

Detección y clasificación de enfermedades a partir de imágenes médicas por medio de redes neuronales

Óscar Alexander Martín Tacoronte

Tutor

Javier Sánchez Pérez

Las Palmas de Gran Canaria
Junio de 2024

Agradecimientos

En este apartado me gustaría expresar mi agradecimiento a todas las personas que de alguna manera han aportado su granito de arena a que pueda seguir progresando, y que, de una manera u otra, han contribuido a que este proyecto se lleve a cabo. Uno de los valores que me han enseñado es a ser agradecido, y no puedo cerrar esta etapa en mi vida sin antes reconocer a quienes me han ayudado a seguir adelante y a superar cada obstáculo.

En primer lugar, agradezco a Javier Sánchez Pérez por tutorizar este proyecto, aportar conocimientos útiles y de gran valor, y distintos puntos de vistas interesante que me han ayudado a mejorar en distintos ámbitos. He ganado una experiencia de gran valor con este trabajo.

En segundo lugar, me gustaría dar las gracias a mis compañeros y profesores de la Universidad que han sido un continuo apoyo y un pilar fundamental para que yo pueda estar completando este Trabajo de Fin de Título. También me gustaría mencionar a mis compañeros con los que compartí muchos momentos agradables y experiencias inolvidables durante el programa de Movilidad de Erasmus en República Checa.

En tercer lugar, me gustaría expresar mi gratitud a mis padres, familia y amigos. Han sido la piedra angular para mi crecimiento personal y profesional, por todo lo que hemos vivido codo con codo: hemos disfrutado en las rachas buenas, y en las rachas malas hemos hecho de todo para poder seguir adelante sin rendirse y con determinación. Las experiencias personales no se olvidan, y soy muy afortunado de poder contar con ustedes. He aprendido mucho durante este año, y les admiro probablemente más de lo que se imaginan por todo lo que han demostrado.

Por último, quería comentar en especial a algunas personas que de alguna manera han contribuido muchísimo para mí con sus acciones. Aunque puede que no tenga un contacto continuo con algunos, cada gesto de apoyo estará grabado en mí: a mi hermano Javier Martín Tacoronte y a mi cuñada Isabel Bañolas Díaz, a mis padres Jerónimo Martín Benítez y María del Carmen Tacoronte Gil, Gustavo Tacoronte Díaz, Raúl Aday Benítez Tacoronte, Juan Antonio Tacoronte Díaz, Marta García Cueto, Tania Mateo Gutiérrez, el *inigualable* Jorge Marrero Sarmiento, Cristina Crespo Rodríguez y Gabriel León Lantigua.

Me gustaría seguir nombrando a personas, pero no puedo extenderme más.

Muchas gracias por formar parte de este camino, y por ayudarme a seguir alcanzado cada objetivo.

Óscar Alexander Martín Tacoronte.

Índice

Índice de Figuras

Índice de Tablas

Resumen

El trabajo consiste en poner a prueba y analizar distintas arquitecturas de redes neuronales de vanguardia que sean capaces de detectar, segmentar y clasificar distintas enfermedades como pueden ser los tumores cerebrales, o el cáncer de riñón. Una vez analizadas y comparadas, la idea es obtener qué arquitectura neuronal ofrece mayores prestaciones computacionales, mientras mantiene una precisión confidente en la identificación y clasificación de tumores.

Se plantea este proyecto para proporcionar una solución a la detección de tumores de manera óptima y rápida, ya que es primordial la detección temprana para poder incrementar la esperanza de vida del paciente.

Sin embargo, el enfoque fundamental será centralizar la toma de decisión de múltiples enfermedades de distintas regiones que no estén correlacionadas, como podría ser el de riñón y el de pulmón, y capacitar al modelo la toma de decisiones inteligentes.

En definitiva, el objetivo es aportar un sistema robusto y eficiente en la detección y clasificación de enfermedades médicas, contribuyendo a la investigación en la lucha contra el cáncer.

Abstract

The aim of this work is to put into practice and analyse different architectures of neural networks at leading edge of technology which they can detect, segment, and classify different kinds of diseases such as brain or kidney tumour. Once they are analysed and compared, the aim is obtaining which architecture gets the best performance while keeps a trustful accuracy.

This project is proposed to provide a solution at the optimal and rapid tumour detection, since the early tumour detection is essential to be able to increase the life expectancy.

However, the fundamental approach will be to centralize decision making for multiple diseases from different regions that are not correlated, such as kidney and lung, and to enable the model to make intelligent decisions.

In short, the aim is to bring a robust and efficient system at the detection and classification of medical diseases, contributing to the research against the cancer.

Capítulo 1. Introducción

La tecnología en la medicina ha ido evolucionado a pasos agigantados en esta última década, siendo las redes neuronales una pieza fundamental en este avance y se ha ido aplicando eventualmente en la investigación de enfermedades y diagnósticos médicos. La combinación de estos dos campos, por un lado, el de la informática y, por otro lado, el de la medicina, presenta una colaboración significativa de la cual la sociedad en su conjunto se puede beneficiar proporcionando una calidad de vida prolongada y de mayor calidad. Según [1], se puede ver un ejemplo significativo de estos avances donde se puede concluir en esta propia cita que el uso de la inteligencia artificial reduce la labor de los profesionales en la medicina pudiendo aumentar la eficacia de los diagnósticos mientras al mismo tiempo ayuda a filtrar el trabajo de los expertos.

Existe diversas maneras de aplicar métodos de inteligencia artificial en apoyo a la medicina [2]. Algunos de estos casos pueden ser el diseño de software de monitorización en dispositivos ponibles (*wearables*) recogiendo datos constantemente sobre el estado de salud para que puedan recibir y detectar anomalías, también existen algoritmos de aprendizaje profundo para identificar anomalías en la interpretación del ADN. Por tanto, se entiende que la inteligencia artificial es un campo muy amplio, y puede ser integrado en múltiples casos de uso en la salud. En este trabajo, la fuente de aprendizaje de los modelos será por el diagnóstico por imágenes, de tal manera que se podrá determinar qué tipo de tumor se aprecia en la imagen, y su localización. Dentro de este ámbito, se puede identificar varios formatos de imágenes, las cuales son vitales tener en cuenta pues cada formato tiene su propósito en diversas situaciones. En este trabajo se tratará con imágenes de resonancia magnética e imágenes de tomografía computarizada, que permitirán utilizarse para detectar zonas blandas del cuerpo y masas cancerígenas [3].

En relación con el desarrollo del proyecto, la visión artificial incluye el análisis y el procesamiento de imágenes, el cual proporcionará diferentes algoritmos de aprendizaje profundo con el que se podrá poner a prueba el uso de diversas técnicas para extraer las cualidades y patrones, e interpretar estas imágenes para localizar la masa cancerígena anómala. Estas herramientas requieren un entrenamiento con los que se pueda encontrar y detectar estos patrones, y para ello se llevará a cabo un entrenamiento denominado como aprendizaje automático. De esta manera, el entrenamiento consiste en proporcionar al sistema un conjunto de imágenes, comparar entre diferentes clases continuamente y una supervisión que le pueda enseñar a base de prueba y error si ha realizado correctamente la tarea de detección de enfermedades de manera correcta.

En este proyecto se emplearán redes neuronales basadas en la arquitectura denominada Transformador de Visión (*Vision Transformer,* o también *ViT*) y alguna de sus variantes. Este tipo de arquitectura se ha utilizado en las investigaciones recientes tanto en procesamiento de lenguaje natural como en reconocimiento de imágenes dados sus buenos resultados en forma de rendimiento y efectividad sustituyendo a las redes neuronales convolucionales. En concreto, la arquitectura base de Transformador de Visión, Transformador Deslizante (*Shifted Window Transformer*, o también *Swin Transformer*) y MaxVit son los modelos que han sido estudiados.

Las diferencias entre las redes convolucionales y los transformadores son notables desde el punto de vista arquitectónico, y la elección entre ellas dependerá de la tarea en cuestión. Teniendo en cuenta la

complejidad de detectar patrones anómalos en imágenes, por ejemplo, presentar permutaciones en las imágenes, es decir, cambios en el orden de localización o región de la masa cancerígena u otra clase de distorsiones, se deberá elegir un enfoque distinto. Para esta tarea, los transformadores están diseñados para enfrentarse a esta clase de inconvenientes de manera más eficaz que las redes convolucionales, por lo que se llevará a cabo a partir de este enfoque arquitectónico.

Para llevar a cabo la investigación, se deberá aplicar un marco de trabajo o metodología adecuada para un proyecto que requiera una organización reiterativa en las fases de modelado, evaluación y despliegue de los modelos entrenados, así como de búsqueda y tratamiento de datos que se utilizará en el propio entrenamiento [4]. La metodología CRISP-DM ha sido empleada para la orientación de cada una de las fases desarrolladas en el proyecto, y se entrenará más en detalle en el **capítulo 3**. Además, será necesario realizar un estudio con el objetivo de comprender la naturaleza del problema en el contexto de las enfermedades, así como un análisis del estado actual de la tecnología y de la medicina, ambas lecturas se pueden localizar en este documento en los **capítulos 2 y 5**, respectivamente. También es importante mencionar que cuando se emprende un proyecto, se deberá listar las herramientas empleadas, así como de recursos disponibles y los limites computacionales, y de ser consciente qué tipo de solución al problema se puede plantear con los recursos que se tienen. En el **capítulo 4** se pueden consultar acerca de estas consideraciones.

Una vez establecido un marco de trabajo adecuado como CRISP-DM, haber completado la fase de entendimiento del problema a nivel tecnológico y el contexto actual de la medicina, y haber analizado las herramientas disponibles y sus límites inherentes, entonces se puede comenzar con la búsqueda de los conjuntos de datos utilizados. Se emplearon imágenes de resonancia magnética y tomografía computarizada de tumores originados en el cerebro y se puede identificar una tarea principal que consistirá en la clasificación de clases como gliomas, meningiomas o el tumor de la hipófisis. Otro tipo de cáncer trabajado en este proyecto son los tumores de pulmón, identificando clases de cualquier tumor cancerígeno, cualquier tumor benigno, y de pulmón sano; siendo el objetivo de este caso la detección de cáncer genérico de pulmón frente al tumor benigno y pulmón sano. En cuanto al último caso, se encuentra el cáncer del riñón, cuya tarea destaca entre clasificar el cáncer y otras clases como el quiste, piedras o riñón sano que puedan dificultar la detección de un cáncer genérico de riñón. Como se puede apreciar, la naturaleza de los conjuntos de datos es distintas, por ejemplo, en el caso del cerebro se requerirá un modelo que capture detalles finos para poder clasificar los tipos de cáncer, mientras que en el caso del riñón y pulmón consistirá en la detección y clasificación de casos más genéricos. En el **capítulo 7** se podrá acceder a la información de los conjuntos de datos.

Una vez haber determinado qué modelos serán objetos de estudio, y haber comprobado que los conjuntos de datos encajan en el método de solución al problema en el **capítulo 6**, se puede enfocar en la fase de evaluación de modelos en el **capítulo 8**, donde además se deberá ajustar los distintos hiperparámetros que convergerán hacia la mejor efectividad posible de cada modelo. Posteriormente, se hará comparaciones de las arquitecturas y se sacarán las conclusiones en base al contexto del problema.

1.1 Motivación

Eventualmente, es inevitable no tomar conciencia del cáncer. El cáncer es un término que abarca un amplio grupo de enfermedades y si se presta atención a las estadísticas, se puede ver que el cáncer es la principal causa de muerte que afecta a la sociedad causando millones de muerte por año, por ejemplo, según [5] hubo hasta 18 millones de casos nuevos de cáncer y hasta 9 millones de muertes por cáncer.

Muchas de estas enfermedades son potencialmente peligrosas para la salud, la detección y por tanto, la aplicación tardía de un diagnóstico puede causar efectos irreversibles o incluso la muerte. Teniendo en cuenta la complejidad del problema, se necesita complementar la aplicación de tecnologías que puedan dotar a los profesionales en la salud herramientas que sean capaz de detectar con eficacia, rapidez y precisión aquellas enfermedades nocivas para la salud. Además, se necesita buscar que estas herramientas sean automatizadas, tratando de que detecte aquellos detalles que el error humano natural pueda pasar desapercibido y aliviando la carga de trabajo a los profesionales de la salud.

Se pondrá entonces a prueba diversas arquitecturas de redes neuronales que puedan proporcionar un rendimiento optimo garantizando la optimización computacional y la eficacia y precisión de decisión. También se pretenderá centralizar la toma de decisiones en un mismo modelo, manteniendo su eficacia en el momento de la clasificación y detección de varias enfermedades no correlacionadas, y de imágenes de formatos distintos.

1.2 Objetivos

El objetivo primordial es poner en práctica diversas arquitecturas del modelo Transformer y medir el rendimiento a la hora de detectar y clasificar diferentes tipos de tumores a partir de imágenes de resonancia magnética.

Se deberá entrenar un conjunto de datos balanceado y completo. Para cada entrenamiento, se buscará y ajustará los hiperparámetros de los modelos empleados con el objetivo de maximizar su precisión. Y finalmente, se tendrá en cuenta los costes computacionales, y la eficacia de cada arquitectura.

1.3 Organización del documento

En esta sección se presenta de forma resumida cómo se dividen los distintos capítulos que compone este trabajo:

Capítulo 2. Estado del arte: se investigará y estudiará el origen y avance de la inteligencia artificial en los sistemas de información. Asimismo, se introducirá el contexto de la industria tecnológica aplicada al campo de la medicina con el propósito de entender el estado de la materia que va a ser estudiada en este proyecto.

Capítulo 3. Planificación y metodología: se presentará la estructura de las tareas que se llevará a cabo en el proyecto, así como el enfoque sistemático que se empleará para llevar a cabo la investigación exploratoria.

Capítulo 4. Herramientas tecnológicas: se describirán las herramientas más relevantes y los recursos utilizados durante el proyecto.

Capítulo 5. Estudio de enfermedades médicas: se explicarán cuestiones básicas de las enfermedades presentes en la exploración, con el propósito de entender los métodos de diagnósticos que serán utilizados en la investigación. Además, se buscará la correlación existente entre las características que se manifiesta en los diagnósticos y la detección de las enfermedades en la inteligencia artificial.

Capítulo 6. Conjuntos de datos: se hará referencia a las fuentes de los conjuntos de datos que se dispondrá en el entrenamiento de la inteligencia artificial.

Capítulo 7. Modelos de redes neuronales: se desarrollarán los distintos diseños arquitectónicos, identificando las diferencias entre los mismos.

Capítulo 8. Evaluación de los modelos: se analizarán los resultados obtenidos, y se desarrollará una explicación rigurosa sobre el desempeño de cada arquitectura, evaluando el enfoque óptimo de cada modelo relacionado con las características de la fuente de datos, así como de las enfermedades.

Capítulo 9. Conclusiones y Trabajo Futuro: se comentará la información relevante extraída de la investigación científica que pueda ser identificada de esta exploración tecnológica.

Capítulo 2. Estado del arte

La inteligencia artificial en la informática tiene sus orígenes en el siglo pasado a partir del concepto del perceptrón, la unidad con la que se formaría la base de las redes neuronales entre las décadas de 1950 y 1960, y con la primera red neuronal multicapa publicada en el primer artículo científico en 1975 [6]. Sin embargo, esta última década ha experimentado un crecimiento significativo con las nuevas investigaciones en el área del aprendizaje profundo.

Estos últimos avances proporcionan a la comunidad científica unas herramientas de alto valor con las que se pueden desarrollar algoritmos capaces de aportar un apoyo al equipo médico y seguir ofreciendo soluciones a uno de los grandes problemas de la sociedad como es la causa de muerte por cáncer.

2.1 Avance tecnológico del aprendizaje profundo

La primera implementación de un software de aprendizaje automático nace a partir del concepto del perceptrón en la década de 1950 por el Doctor Frank Rosenblatt. En una de sus investigaciones presenta el perceptrón como un clasificador binario o discriminador lineal, el cual a partir de una entrada (*input*) y una ponderación para cada entrada y una función de activación, el algoritmo clasifica la salida (*output*) en formato binario. Se llegó a implementar en un software por primera vez en un IBM704 (el primer ordenador comercial), llegando a ejecutar hasta 40.000 instrucciones por segundo comparados con 100.000 MIPS que podría realizar un procesador Intel Core i7 en la actualidad según [7].

Es en el periodo de 1980 y 1990 cuando surge el concepto de *backpropagation*, algoritmo fundamental en aprendizaje profundo con el que se puede aplicar para poder entrenar de forma masiva las redes neuronales, tal y como se explica en los orígenes del *Deep Learning* [8]. Una vez se obtiene un resultado en una red neuronal, con este algoritmo se logra minimizar los errores que, propagando el ajuste de pesos de las neuronas desde el final del modelo hacia el comienzo, es lo que se conoce por su terminología en inglés como *backward pass*. El resultado de la función de error será la medida de cuán lejos está las neuronas de acertar la solución correcta.

Este avance dará pie al aprendizaje profundo, permitiendo el aprendizaje de las capas ocultas de una red neuronal y ofrecerá a la comunidad la aparición de las **redes convolucionales** (*CNN – Convolutional Neuronal Network*) capaces de distinguir y reconocer objetos durante la década de 1990, aproximadamente. Sin embargo, la consolidación y adopción a mayor escala será durante el periodo entre 2000 y 2012, siendo el avance del hardware un gran apoyo a la comunidad tecnológica permitiendo extraordinarios acontecimientos en la historia del aprendizaje profundo [9] [10]. El auge de las redes convolucionales ha marcado una nueva era marcando avances como *DeepFace* capaces de reconocer y verificar personas en imágenes con una precisión muy alta.

No obstante, en 2017 aparece por primera vez la arquitectura denominada como Transformador (*Transformer*) [11] y tras el dominio de las redes convolucionales, esta nueva arquitectura impulsará el aprendizaje profundo a un alcance y escala más potente que permiten avances actuales como es el ejemplo de un chatbot muy conocido, ChatGPT [12] [13].

El modelo Transformer es capaz de realizar tareas que ayudan a la sociedad de diferentes formas. Por ejemplo, logran traducir lenguaje natural tanto escrito como en hablado casi en tiempo real, permitiendo capacidades tecnológicas como acceso a reuniones para asistentes con discapacidades auditivas. Otro ejemplo, son capaces de detectar y comprender el espacio en el que se rodea una cámara, adaptándose al ambiente a partir de imágenes en tiempo real ofreciendo productos como la conducción autónoma [14].

Un proyecto que ha presentado el Cabildo de Gran Canaria junto a la Universidad de Las Palmas de Gran Canaria es el desarrollo de una guagua autónoma que circula en piloto automático (sin conductor) haciendo un recorrido preconfigurado por el Campus de Tafira y adaptándose a la circulación normal donde circulan el resto de los usuarios viales. Dentro de la guagua hay un controlador para situaciones de emergencia o no previstas [15].

2.2 Aplicaciones del aprendizaje profundo en la medicina

Los primeros trabajos de inteligencia artificial en medicina se originan sobre la década de 1970 donde se destaca algunos proyectos como Present illness Program (1976) un sistema de emulación de evaluación de pacientes con edema o CASNET (1978) un método de decisiones médicas basada en computación asistida basada en una red casual-asociativa, publicado en [16]. Posteriormente, en la siguiente década se desarrollan herramientas con el objetivo de resolver tareas más automatizadas de las cuales naturalmente están encargadas el cuerpo médico como son los diagnósticos, o planificación de tareas. De esta manera, nace la Asociación de Inteligencia artificial avanzada (*AAAI - Association for the Advancement of Artificial Intelligence*) en Estados Unidos. La fuente [17] ha sido de gran utilidad en cuanto a la historia de la Inteligencia Artificial en medicina.

Posteriormente, el interés de la inteligencia artificial llega a una escala más globalizada y durante este tiempo, según [18], tiene lugar el avance de métodos probabilísticos como la teoría de conjuntos borrosos o las redes Bayesianas, métodos computacionales como las propias redes neuronales, así como las redes de comunicación y expansión de los microordenadores. Estos avances permitirán a los investigadores y científicos aplicar sistemas inteligentes en el campo de la salud a nivel global. De esta manera, aproximadamente por la mitad de este siglo, la evolución es considerablemente significativa: avances en el rendimiento computacional en forma de tiempo de respuesta y en almacenamiento, aumento del volumen de datos con el crecimiento del Big Data, avances en investigación tecnológica en herramientas informáticas y médicas [18].

En la actualidad, la aplicación de la inteligencia artificial en la medina está en continua evolución, se estima que en 2025 habrá inversiones en la aplicación de la inteligencia artificial en medicina que alcanzará hasta los 36 mil millones de dólares a nivel global, lo que implica un aumento de un 50% con respecto a 2018 según [19]. Algunos de estos beneficios con los que se puede encontrar en la detección y localización temprana de enfermedades, y donde destacan los modelos de aprendizaje profundo como los que se tratarán en este proyecto capacitados para extraer patrones anómalos y reportar dichos datos al equipo médico para que puedan realizar un diagnóstico próspero. Otros beneficios con los que se puede encontrar en la inteligencia artificial es el tratamiento de fármacos, facilitando el análisis de las secuencias genéticas para poder investigar el avance de vacunas o la monitorización de datos en dispositivos ponibles según [20].

Enfocándose la inteligencia artificial en la detección de tumores, se aprecian experimentos donde aplicando algoritmos de aprendizaje profundo se puede lograr resultados excelentes. Por ejemplo, en una investigación reciente [21] sobre la detección y clasificación de tumores cerebrales, se utiliza unas variantes de modelos de redes convolucionales como el modelo de *EfficentNet*. En este estudio, se logró hasta un 98,4 % de precisión utilizando varios conjuntos de datos de imágenes de resonancia magnética, poniendo a prueba el impacto de la inteligencia artificial para mejorar la precisión y eficacia en la detección temprana de tumores cerebrales. Otro proyecto similar presentado en [22], también apoyándose en el empleo de redes convolucionales, pero en esta ocasión se resalta entre otros, modelos como *VGG* y *MobileNet*, logrando un resultado extraordinario. En este caso, entre otros resultados, destaca precisiones que alcanzan hasta un 98,7 %, que demuestra la importancia y avance del aprendizaje profundo en el área de la medicina.

En el artículo reciente [23], se destaca la importancia en la segmentación de imágenes médicas a partir de un modelo de Transformador de Visión (*ViT - Vision Transformer*) en lugar de las redes convolucionales. Aunque se menciona la efectividad de estas redes convolucionales para capturar las relaciones locales, también se destaca la limitación de capturar las relaciones globales en el análisis de las imágenes. Como respuesta a dicha limitación, se propone el uso de modelos de Transformer, cuya característica crucial se presenta en el mecanismo de atención con el objetivo de cubrir este problema. Se pone a prueba, entre otros modelos, algunos como el Transformador de Visión en la versión base, o el Transformador de Ventanas Deslizantes (*Shifted Window Transformer,* o *Swin Transformer*), que será empleado durante esta investigación. Incluso es posible el uso de modelos híbridos de redes convolucionales y el modelo en la versión base del Transformador de Visión, como también se pone a prueba en [23].

Si bien, el actual estado del arte presenta resultados bastante favorables, en este trabajo se pretende unificar la toma de decisiones en un solo modelo entre distintos tipos de enfermedades cancerígenas, benignas y tejido sano; mezclando así distintos conjuntos de datos. Cabe destacar también, que se pretende mezclar también distintos formatos de imágenes, como son las imágenes de resonancia magnéticas y las imágenes de tomografía computarizada. Adicionalmente, se busca usar otro enfoque arquitectónico diferente, como son el modelo Transformer en lugar de las redes convolucionales. Este enfoque se desarrolló originalmente a partir de tareas de procesamiento de lenguaje natural, aunque se ha adaptado y aplicado a tareas de visión por computadora.

Capítulo 3. Planificación y metodología

En proyectos de ingeniería y gestión del software es necesario detallar las planificaciones de las tareas y su seguimiento con el fin de entregar los hitos de manera exitosa cumpliendo con los requisitos dentro del tiempo estimado. Se procede de esta manera a definir la metodología, así como las tareas que se plantean para cumplir los objetivos del proyecto.

3.1 Metodología

Se entiende como metodología al procedimiento a lo largo de una investigación o proyecto que ayuda a establecer las tareas que se ejecutarán de forma sistemática para poder lograr una solución al problema o lograr uno o varios objetivos [24]. Los científicos de datos a menudo construyen modelos con los que puedan extraer patrones a partir de los resultados. Después de los resultados, se tienen en cuenta para poder seguir mejorando los siguientes resultados de forma iterativa. Es un enfoque que en las investigaciones se requiere de una automatización de pasos para crear el modelo, aplicar predicciones, retroalimentar el modelo con los resultados recientes y volver a aplicar predicciones. Ante la necesidad de automatizar los pasos previos independientemente de la tecnología y volúmenes de datos, se presenta una metodología denominada Metodología Fundamental para la ciencia de datos, tal y como se explica en [24].

Para este proyecto se necesita también un enfoque analítico para poder automatizar o facilitar el ciclo de vida del software, partiendo de la creación de los modelos, la evaluación de los modelos, extracción de los patrones y volver a retroalimentar cada modelo de tal manera que pueda aprender y lograr mejores resultados en función de su evaluación previa. Por tanto, se empleará una metodología estándar para la ciencia de datos.

Para este caso, se escogerá una metodología CRISP-DM descripta oficialmente en [25], aunque se guía y se comprende los conceptos en función de [26] al ofrecer un marco de trabajo simplificado del trabajo original. A continuación, en la Figura 3-1 se aprecia los ciclos y fases descriptas en este marco de trabajo. Esta metodología está enfocada para un entorno empresarial, donde el modelo se ejecuta en la etapa final en un entorno de producción operativo, pero en este proyecto se adaptará para que esta fase sea en su lugar las conclusiones del proyecto completo.

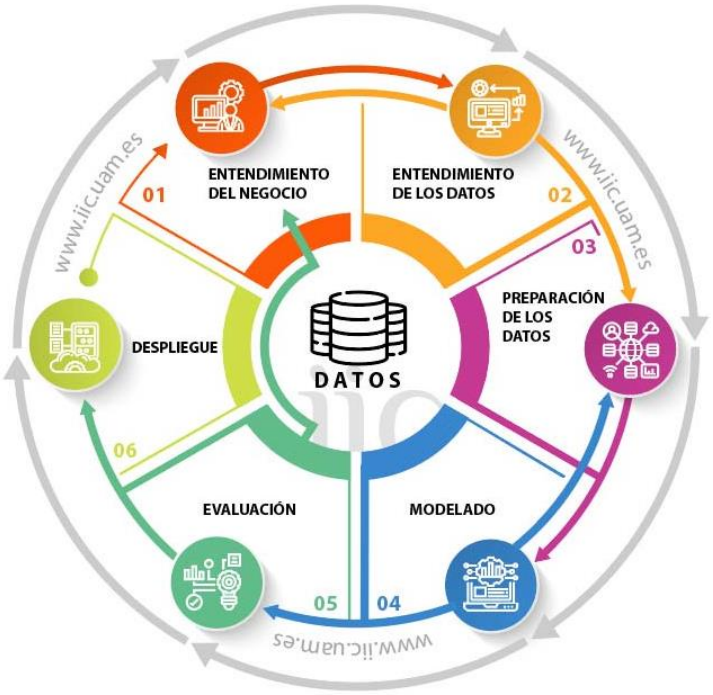


Figura 3-1: Esquema de las fases de la metodología CRISP-DM. Fuente: P.Haya [26].

A continuación, se describe una explicación de cada fase:

- Etapa 1: Entendimiento del negocio: se define el problema, los objetivos y los requisitos que se tenga que cumplir para alcanzar los objetivos. Por consiguiente, se evalúa en qué contexto se envuelve el problema, y se identifica las técnicas más adecuadas para la solución. En este punto se debe identificar en general qué tipo de problema se necesita resolver, por ejemplo, si se trata de un problema de clasificación, de regresión lineal, de agrupamiento o separación u otro tipo. En este caso será un problema de clasificación de distintas clases de enfermedades.

- Etapa 2: Entendimiento de los datos: se explora y se analiza los datos que estén disponibles, con el objetivo de conocer sus características y limitaciones. También se debe verificar si se cumple los requisitos de datos, o si se necesita un volumen mayor en la recopilación de datos. Por lo general, el objetivo es obtener un panorama amplio de los datos disponibles. Se puede regresar a la etapa 1 si el análisis o exploración de datos no ha sido próspero.

- Etapa 3: Preparación de los datos: se adquiere los datos definidos en la etapa anterior según los requisitos identificados. Posteriormente, se limpia y se preprocesa los datos. Si se identifica datos corruptos, o no cumplen las características esperadas, se puede necesitar revisar y explorar nuevas colecciones de datos regresando en la etapa 2. Otras actividades que se pueden encontrar en esta etapa son la eliminación de datos concretos atípicos o de formato incorrecto, reducción del volumen de entrenamiento, la aplicación de restricciones que cumplan los requisitos de datos, se combina conjunto de datos, entre otras. La automatización de estas tareas se vuelve como pieza fundamental para acelerar la velocidad del proyecto, y la elección, así como procesado de datos propiamente correcta o incorrecta puede dirigir al éxito o fracaso de la investigación. Por ejemplo, en este proyecto se ha tenido que automatizar tareas de reducción del volumen de datos para algunas evaluaciones para evitar resultados de sobre aprendizaje.

- Etapa 4: Modelado: se estudia los diferentes modelos candidatos para la solución del problema. Aquí se identifica dos tareas de alta importancia:
    - Se selecciona, y se construye el modelo predictivo que se propone como solución al problema.
    - Se ajusta el modelo con sus respectivos hiperparámetros.

- Etapa 5: Evaluación: se pone a prueba el rendimiento del modelo con el conjunto de datos seleccionado y preparado en función de los hiperparámetros. Posteriormente, se determina si cumple las expectativas. A partir de aquí se detectan diferentes bifurcaciones:
    - No cumple con las expectativas:
        - Se debe seleccionar otro modelo predictivo o ajustar los hiperparámetros, o ambas acciones en la etapa 4.
    - Sí cumple con las expectativas:
        - Se pone a prueba otro modelo predictivo en la etapa 4.
        - Se pone a prueba otro conjunto de datos, por lo que se regresa a la etapa 2.
        - Se progresa a la siguiente etapa final número 6.

- Etapa 6: Despliegue: cuando el modelo ha sido satisfactoriamente evaluado, se integra en una aplicación o entorno de producción, o en este trabajo será revisar la tarea de identificar las conclusiones del trabajo.

La metodología CRISP-DM se caracteriza por su enfoque iterativo, y tal y como se muestra en la Figura 3-1, los ciclos de trabajo serán continuamente entre las etapas 1 y 2 para poder redefinir si fuese necesario los objetivos del proyecto (etapa 1) tras la fase exploratoria si se detecta alguna información relevante (fase 2). Otro flujo que podrá repetirse con frecuencia podría ser entre la etapa 3 y 4 cuando se tenga que preprocesar los datos con nuevas mejoras debido al ajuste del modelo, por ejemplo, al tratar con una arquitectura base que requiera un tamaño fijo de las imágenes, quizás se necesite preprocesar nuevamente los datos. Y finalmente, la evaluación final puede requerir una revisión de los objetivos del negocio en caso de ser necesario.

3.2 Planificación de los capítulos según la metodología

En este capítulo se desglosará los capítulos desarrollados en la investigación resumidos en el apartado 1.3 Organización del documento y se le asociará la etapa correspondiente según la metodología explicada:

Etapa 1: Entendimiento del negocio: corresponde al **Capítulo 2** y **Capítulo 5** del documento, donde se investiga el contexto de las distintas enfermedades y se identifica el paradigma de resolución del problema.

Etapa 2: Entendimiento de los datos: corresponde al **Capítulo 6** del documento, donde se explora distintas fuentes de conjuntos de datos y se selecciona la más apropiada para lograr los objetivos.

Etapa 3: Preparación de los datos: corresponde al **Capítulo 8** del documento, donde además de exponer las evaluaciones, también se explica las configuraciones del modelo y preparación de datos.

Etapa 4: Modelado: corresponde al **Capítulo 7** del documento, donde se evalúa las arquitecturas que se pondrán a prueba.

Etapa 5: Evaluación: corresponde al **Capítulo 8** del documento, donde se reservará varios apartados para analizar, explicar y comparar resultados de las evaluaciones del modelo.

Etapa 6: corresponde al **Capítulo 9**, conclusiones y trabajo futuro del documento, donde se reflexionará y compartirá las conclusiones de la investigación.

3.3 Planificación inicial

Se presenta en la Tabla 3-1 las tareas planteadas en el formulario TFT-01 con su estimación esperada. Las tareas aquí explicadas son mostradas de forma orientativa ya que muestra cómo será el marco de trabajo. Sin embargo, en el apartado 3.4 se expondrá las tareas de forma más detallada.

Fases	***Duración Estimada*** ***(horas)***	***Tareas*** ***(nombre y descripción, obligatorio al menos una por fase)***
Estudio previo / Análisis	60	Tarea 1.1: Investigación y aprendizaje de las enfermedades sobre el campo de la medicina a explorar
		Tarea 1.2: Estudio sobre conceptos de Machine Learning y Vision Transformers
		Tarea 1.3: Exploración tecnológica: selección de modelos variantes de Vision Transformers
Diseño / Desarrollo / Implementación	110	Tarea 2.1: Preparación de los datos de entrenamiento de los distintos modelos de redes neuronales.
		Tarea 2.2: Implementación de los distintos modelos de redes neuronales.
		Tarea 2.3 Validación de los distintos modelos de redes neuronales.
Evaluación / Validación / Prueba	90	Tarea 3.1: Análisis de los resultados de las distintas variantes puestas en práctica
		Tarea 3.2: Comparativa de los resultados obtenidos
Documentación / Presentación	40	Tarea 4.1: Desarrollo de la memoria
		Tarea 4.2: Preparación de la presentación de la exposición

Tabla 3-1: Planificación inicial del proyecto

3.4 Planificación final

En este apartado, se podrá ver en la Tabla 3-2 las tareas planteadas que finalmente se han realizado de manera más detallada.

Fases	***Tareas*** ***(nombre y descripción, obligatorio al menos una por fase)***
Estudio previo / Análisis	Tarea 1.1: Investigación y aprendizaje de las enfermedades sobre el campo de la medicina a explorar.
	Tarea 1.2: Estudio sobre conceptos de Machine Learning y Vision Transformers.
	Tarea 1.3: Exploración tecnológica: selección de modelos variantes de Vision Transformers.
	Tarea 1.4: Estudio de uso, instalación, y pruebas básicas de entorno de herramientas tecnológicas necesarias como Pytorch y Jupyter.

	Tarea 1.5: Investigación del Estado del arte tanto del campo de la medicina como de la Inteligencia Artificial.
Diseño / Desarrollo / Implementación	Tarea 2.1: Búsqueda de datos de entrenamiento, extracción y validación de requisitos de datos, y visualización de un conjunto de casos de pruebas.
	Tarea 2.2: Programación del modelo, así como de herramientas de visualización como Matplotlib para generar gráficos comparativos, y otras funcionalidades que complemente la automatización del código y legibilidad de los resultados.
	Tarea 2.3: Preparación y procesamiento de los datos de entrenamiento para los distintos modelos de redes neuronales.
	Tarea 2.4: Implementación de los distintos modelos de redes neuronales.
	Tarea 2.5 Validación de los distintos modelos de redes neuronales.
	Tarea 2.6: Análisis y reflexión de los hiperparámetros de entrenamiento.
	Tarea 2.7: Elección de los modelos con mayor rendimiento.
Evaluación / Validación / Prueba	Tarea 3.1: Análisis de los resultados de las distintas variantes puestas en práctica durante la fase de Diseño, desarrollo e implementación de los modelos neuronales.
	Tarea 3.2: Comparativa de los resultados obtenidos.
	Tarea 3.3: Conclusiones del trabajo. Reflexión sobre futuros trabajos y mejoras.
Documentación / Presentación	Tarea 4.1: Desarrollo de la memoria.
	Tarea 4.2: Preparación de la presentación de la exposición.

Tabla 3-2: Planificación final del proyecto

Capítulo 4. Herramientas tecnológicas

En esta sección se comentarán las tecnologías más relevantes utilizadas durante el proyecto explicando los motivos de emplear estas herramientas. También se comentará brevemente los recursos disponibles para realizar la investigación y del valor aportado de cada elemento al proyecto.

4.1 Entorno de trabajo

Se formará un entorno de desarrollo en Visual Studio Code, un editor de código que permite una alta personalización que facilitará el desarrollo del software. También se usará **Jupyter**, que proporciona el entorno de desarrollo para aplicaciones en ciencia de datos, con esta herramienta se facilitará la investigación y análisis del software desarrollado. Un *notebook* es un documento donde se podrá desarrollar una aplicación, en este proyecto en Python, y que se compone de celdas donde se puede incluir texto en formato *markdown*, o código ejecutable en Python. De esta manera permitirá realizar las investigaciones de forma apropiada, algunos ejemplos podrían ser volver a ejecutar celdas anteriores o posteriores, o interrumpir la ejecución de celdas para reanudarlas en otro momento, entre otras opciones [27].

Con el propósito de facilitar gestión de paquetes, se usará **Anaconda** en la versión de miniconda 4.14.0. Se trata de una herramienta para gestionar el uso de librerías y dependencias del proyecto. Anaconda permitirá la administración de paquetes, por ejemplo, se podría tener un entorno con Python 2.7 y TensorFlow 2.2 y preparar otro entorno con Python 3.9 y Pytorch 2.12, y en función del proyecto en cuestión se podría elegir cómodamente el que se necesite sin necesidad de mezclar las librerías y dependencias de los proyectos.

4.2 Lenguaje de Programación y módulos.

Se empleará Anaconda para preparar un entorno con las siguientes herramientas:

- Python en la versión 3.9.5, es el lenguaje de programación de alto nivel de alta popularidad en el aprendizaje automático. Por tanto, se desarrollará los modelos de inteligencia artificial con este lenguaje y se apoyará entre las múltiples librerías que ofrece. Se hará uso de algunas funcionalidades como el módulo *time* para contabilizar el tiempo de entrenamiento de los modelos.

- PyTorch en la versión 2.12, es una biblioteca de aprendizaje profundo basado en Torch de Python, utilizada para el procesamiento de lenguaje natural y para la visión artificial [28] .

- Matplotlib en la versión 3.4.3, es una biblioteca de Python que permitirá generar gráficos en dos dimensiones de los resultados del aprendizaje de los modelos [29].

4.3 Entorno de ejecución y tecnologías adicionales

El entrenamiento de los modelos será ejecutado localmente en un sistema operativo **Microsoft Windows 10 Pro**. La elección de este sistema operativo se debe por la compatibilidad y usabilidad de otras herramientas empleadas en esta investigación, siendo importante tener en cuenta que tradicionalmente **Linux** ha sido el entorno de ejecución más eficiente de modelos de aprendizaje automático.

4.4 Recursos de Hardware

El entrenamiento se llevará a cabo en una Unidad de Procesamiento Gráfico (*GPU – Graphics Processing Unit)*, se trata de un coprocesador dedicado a realizar operaciones aritméticas con el propósito de aligerar la carga de trabajo de la Unidad de Procesamiento Central (*CPU – Central Processing Unit*).

La diferencia entre la CPU y GPU se puede identificar en su arquitectura, mientras la CPU es de una arquitectura de von Neumann con un número reducido de núcleos, la GPU se especializa en el procesamiento en paralelo con un repertorio de instrucciones más simple pero mayor número de núcleos, siendo altamente optimizadas para operaciones aritméticas en paralelo [30] [31] .

Los modelos de aprendizaje profundo han sido entrenados en una computadora localmente en una GPU **NVIDIA GeForce RTX 2060 SUPER** con las siguientes características:

- Núcleos CUDA (*Compute Unified Device Architecture*): 2176
- Frecuencia de reloj: 7001 MHz
- Memoria VRAM (*Video Random Access Memory*): 16 GB GDDR6, de las cuales 8 GB es memoria dedicada y los 8 GB restantes son memoria compartida entre CPU y GPU.

El almacenamiento de los conjuntos de datos ha sido almacenado localmente en un disco local conectado a la misma computadora en la que se realiza el entrenamiento.

4.5 Tecnologías de computación acelerada

Para poder utilizar una GPU y aprovechar el paralelismo, así como el ancho de banda de la memoria gráfica se necesita incluir un conjunto de tecnologías que permitan ofrecer la implementación de estas tareas e internamente gestione la comunicación entre la CPU, así como el rendimiento de la GPU, entre otras tareas. Por tanto, se instala y se utiliza las siguientes tecnologías:

CUDA o Arquitectura Unificada de Dispositivos de Cómputo en la versión 11.8, es una plataforma de computación paralela para la programación de modelos en unidades de procesamiento grafico (GPU) desarrollado por NVIDIA. CUDA incluye un compilador, librerías y herramientas de desarrollo para ejecutar aplicaciones que requieran el uso intensivo de operaciones aritméticas y serán optimizadas cargando el procesamiento en los núcleos *cuda* [32].

cuDNN (*NVIDIA CUDA Deep Neural Network*) en la versión 7.7.0, es una librería de Aprendizaje profundo, específicamente diseñada para optimizar y acelerar las operaciones en el entrenamiento y la inferencia de redes neuronales profundas sobre el hardware de NVIDIA [33].

Capítulo 5. Estudio de enfermedades médicas

En este capítulo se abordará una explicación sobre los conceptos médicos relacionados con el trabajo, así como el origen y característica de la enfermedad en cuestión como son los tumores, las enfermedades concretas y sus tipos que han sido puesta en práctica en los modelos de entrenamiento de red neuronal.

Es primordial entender con qué enfermedad se trabajará, el formato de los datos que se a utilizar y el motivo de por qué se usa dicho formato para poder comprender el contexto y lograr los objetivos expuestos.

5.1 Tumores y diagnóstico: Imágenes médicas

Se considera como tumor al crecimiento anómalo en la masa de un tejido en el cuerpo. En función de este crecimiento puede ser un tumor benigno o maligno. Un tumor benigno está caracterizado por un crecimiento lento y que además no se expande a otros tejidos cercanos u otras partes del cuerpo. Este tipo de tumor no es nocivo para la salud, pero su comportamiento podría cambiar sin previo aviso por lo que se recomienda una revisión periódica. En cambio, un tumor maligno, también llamado como tumor cancerígeno, se caracteriza por un crecimiento más rápido invadiendo tejidos cercanos incluso alcanzando otras partes del cuerpo. Este tipo de tumor sí presenta un riesgo perjudicial para las personas, provocando un deterioro en la calidad de vida o incluso el fallecimiento de la persona.

Para el diagnóstico, el cuerpo médico realiza una serie de pruebas que incluye una historia clínica en la que se recopila la información respecto a la enfermedad y tratamiento, pruebas de laboratorio en algunos casos, o el estudio de imágenes para la detección temprana o para examinar detalladamente enfermedades. El diagnóstico por imagen permite observar el interior del cuerpo del paciente con el que podrá buscar alguna anomalía. Para ello existe una amplia variedad de tecnologías y técnicas que puede usar el profesional en la salud, y que podrá seleccionar en función de los síntomas del paciente, la parte del cuerpo que se requiere estudiar, o incluso en revisiones médicas periódicas de prevención [34].

5.1.1 Tomografía computarizada

La tomografía computarizada es un procedimiento médico para poder obtener imágenes detalladas de alguna región del cuerpo que usa un equipo especial de rayos X, tal y como se explica en [35]. Las maquinas más modernas de tomografía computarizada incluyen una característica que permite tomar imágenes en forma de espiral, de tal manera que resulta en imágenes de mejor calidad incluyendo la tercera dimensión (3D), y puede detectar más fácilmente las anomalías pequeñas.

Su utilización normalmente se aplica en oncología y para enfermedades del sistema circulatorio, lesiones en la cabeza, traumatismo óseo o lesiones de órganos internos.

En oncología, se integra de múltiples formas: detección precoz del cáncer, obtener información acerca del estadio de un cáncer, en el diagnóstico de un cáncer, en el seguimiento de un tratamiento, en la recurrencia de un tumor, entre otros [35].

5.1.2 Imagen por resonancia magnética

Las imágenes por resonancia magnética es otro procedimiento con el que se puede producir imágenes tridimensionales con detalles. Por lo general, su aplicación se centra en la detección de enfermedades, el diagnóstico o la monitorización del estado de la enfermedad [36].

Las imágenes por resonancia magnética se basan en una serie de imanes que producen un campo magnético, con el que logra que los protones en el cuerpo se alineen con el campo magnético. A partir de aquí, se aplica un pulso de ondas electromagnética de alta frecuencia con el que se perturba al protón de tal manera que lo fuerza a girarse de 90 a 180 grados con el campo magnético. Una vez el pulso electromagnético deja de ejercer su influencia sobre los protones del cuerpo humano, el protón de realinea con el campo magnético de la maquina liberando la energía electromagnética durante el trayecto de realineación. La máquina detecta esta energía diferenciando los tejidos en función de la rapidez que libera la energía después de eliminar el pulso de ondas electromagnética, según [36] y su explicación gráfica bastante intuitiva [37].

La motivación a utilizar este tipo de imágenes se debe a su principal característica de diferenciar tejidos blandos del cuerpo, así como la masa cancerígena. Puede diferencia entre materia blanca y gris, y es particularmente útil para diagnósticos de aneurisma y tumores según [36]. Aunque es la modalidad preferida para imágenes en el cerebro, también tiene un mayor coste que el escáner de tomografía computarizada.

5.2 Tumor del riñón

Según [38] se estima que hasta aproximadamente 430.000 personas fueron diagnosticadas de cáncer de riñón en el año 2020, y el número de casos en los Estados Unidos ha ido creciendo con el paso de las décadas siendo en 2023 superada la cifra de 80.000 nuevos casos de cáncer de riñón en adultos, y posicionándose en el sexto lugar de tipo de cáncer más frecuente en los Estados Unidos en los hombres y noveno, en las mujeres.

La tasa de supervivencia relativa según [38] es del 77% a 5 años, en función de varios factores como el tratamiento, la expansión del cáncer, la edad del paciente, o el estado de bienestar entre otros factores. Esta estadística se refiere a la esperanza de vida del paciente durante el tiempo estipulado tras el diagnóstico de la enfermedad o el inicio del tratamiento.

La importancia de la detección temprana es apreciable en las estadísticas que ofrece esta misma fuente, y según su expansión se tiene las siguientes características:

- Aproximadamente dos de cada tres personas reciben el diagnóstico cuando el cáncer se encuentra **únicamente** en el riñón, entonces la tasa de supervivencia a 5 años asciende hasta el **93%**
- Si se tiene una diseminación hacia los **tejidos u órganos próximos**, o a los ganglios linfáticos regionales, la tasa de supervivencia desciende al **72%**.
- Si finalmente se disemina a una **parte distante del cuerpo**, dicha tasa de supervivencia desciende drásticamente hasta un preocupante **15%**.

Para entrenar el modelo de inteligencia artificial, se pondrá a prueba un conjunto de datos de riñón clasificados en 4 clases:

- No tumor: representará la clase que pertenezcan las imágenes de riñones sanos.
- Piedra: representará la clase cuyas imágenes en el riñón se localicen un material sólido anormal en el riñón. Según [39], este tipo de material se presenta en los riñones cuando los niveles de ciertos minerales son altos. Normalmente se presenta con un dolor agudo en la espalda baja o en la ingle o incluso sangre en la orina. Su diagnóstico depende de la salud física, las pruebas del laboratorio y el tamaño de la piedra visible en las imágenes. Por lo general, el tratamiento suele ser extraer la piedra o romper la piedra en pedazos.
- Quiste: representará la clase donde se identifique pequeñas estructuras acuosas o líquidas que normalmente no son nocivas para la salud. Es necesario identificar el quiste simple que no tiene riesgo de transformarse en cáncer de riñón, y el quiste complejo, que existe riesgo de transformarse en cáncer de riñón, aunque este riesgo es bajo y requeriría de la supervisión de un urólogo cirujano que diagnostique el caso con detenimiento. El riesgo de este quiste complejo depende de su estructura interna y se puede clasificar según el sistema de clasificación *Bosniak* accesible en [40].
- Cáncer: representará la clase donde se detecte la presencia de masa cancerígena. Según [41] el carcinoma de células renales es el tipo de tumor más común que representa hasta el 90% de cáncer de riñón. Gran parte de los casos se detectan en etapas tempranas del tumor, y durante las revisiones médicas periódicas. Por lo general, no suele presentar síntomas, y la cirugía es el tratamiento estándar para extraer el tumor, alcanzando una tasa de curación superior al 70%.

5.3 Tumor de cerebro

El tumor cerebral se origina en el crecimiento anormal de células cancerígenas de la estructural cerebral. Puede ser un tumor benigno o maligno, y puede afectar a diversas áreas del cerebro y consecuentemente provocar distintas variedades de síntomas dependiendo de su ubicación, tamaño y grado de gravedad [42]. Se puede tratar de un tumor primario, cuando se origina en el cerebro a partir de células cerebrales, membrana alrededor del cerebro, las glándulas, o cualquier tejido del cerebro. También se puede tratar de un tumor metastásico, que son aquellos que se han expandido al cerebro procedente de otra localización externa al cerebro.

Las investigaciones determinan que la esperanza de vida del paciente depende, en gran parte, de la extirpación total del tumor, por lo que una detección temprana del mismo dará mayor posibilidad de tratamiento adecuado antes del crecimiento del tumor. Además, un tumor detectado tempranamente sería un tumor más pequeño, lo que facilitaría una extirpación completa y minimizaría estas complicaciones. De esta forma, cabe resaltar la importancia de que aumentar las

posibilidades de un pronóstico más próspero del paciente es crucial una detección temprana del tumor [42].

Sin embargo, el pronóstico también depende en gran medida de la agresividad del tumor. Para este proyecto se pone a prueba la clasificación de diferentes tipos de tumores cerebrales:

- **Glioma**: se trata de un tipo de tumor con diversas variantes, y es un tipo de tumor considerado como común. Según [43], hasta un 33 % de tumores cerebrales son de tipo gliomas. En cuanto a sus características, algunos no se consideran peligrosos, pero otros se consideran bastante agresivos expandiéndose hacia los tejidos sanos del cerebro y provocando presión sobre el cerebro o la médula espinal [44]. En el caso de los gliomas, se trata de células cancerígenas cuyo aspecto es similar a las células gliales que rodean a las células nerviosas del cerebro y la medula espinal. Crecen dentro de la sustancia blanca del cerebro, y peligrosamente se diseminan con los tejidos cerebrales sanos.
- **Meningioma**: como se menciona en [45], es el tumor primario cerebral más común, sin embargo, los meningiomas se clasifican en tres grados según sus características y siendo el grado alto muy poco común.
  1. **Grado I**. El más común, y el tumor de grado bajo, siendo por tanto un tumor de crecimiento lento
  2. **Gado II**. Grado intermedio, denominado meningioma atípico. Se caracteriza por una mayor probabilidad de aparición después de haber sido extirpado.
  3. **Grado III**. El de mayor grado, pero menos común, caracterizado por ser maligno y denominado meningioma anaplásico. Su apariencia discierne de las células normales, y se expanden con gran rapidez.

  Las estadísticas que proporciona [45] no son muy favorables. Según dicha fuente, se calcula que cada año 371 personas se diagnostican de este tumor mientras que 2692 personas conviven con este tumor sin ser detectado siendo la esperanza de vida de 5 años para el 63,8% de personas. Además, los meningiomas atípicos y los meningiomas anaplásicos tienen la capacidad de diseminarse tanto por el cerebro como por otros órganos del cuerpo. Este tipo de meningiomas, se podrán distinguir por una masa sobre la capa externa del tejido del cerebro. Con estos datos en conocimiento, se concluye la importancia de añadir este tipo de cáncer al conjunto de datos que será entrenado con el objetivo de que pueda detectarse con la mayor tempraneidad posible.
- **Tumor de la hipófisis**: tipo de tumor cerebral que por lo general no es invasivo. Según [46], presenta una altísima probabilidad de ser en cualquier caso un tumor benigno con un grado de crecimiento bajo. Sin embargo, existe el riesgo de crecimiento y diseminación hacia otras partes como el nervio óptico o las arterias carótidas, presentando características más agresivas. A continuación, se procede a compartir la cita del neurocirujano Joaquim Enseñat:

  "*Los tumores de hipófisis se tienen que operar en aquellas situaciones en que el tumor crece y produce una compresión de los nervios ópticos. El posoperatorio de la cirugía de los tumores de la hipófisis suele ser muy bien tolerado.*" Fuente: J. Enseñat, accesible en [46].

  Según [46], a menudo los tumores hipofisiarios pasan desapercibidos y son detectados al realizarse una prueba por motivos ajenos al cáncer, y teniendo en cuenta la cita de Joaquim Enseñat, se concluye que el tumor de la hipófisis debería tenerse mayor en cuenta en las investigaciones y será incluido en este proyecto.

5.4 Tumores de pulmón

Los tumores de pulmón pueden originarse en varias células del pulmón, incluyendo los bronquios, bronquiolos y alvéolos pulmonares. Se pueden identificar dos tipos:

- Cáncer de células pequeñas, el menos común y de crecimiento rápido.
- Cáncer de células no pequeñas, el más común y de crecimiento lento.
- Cáncer mixto de células grandes y pequeñas.

También pueden desarrollar cáncer de pulmón metastásico, que se origina cuando el cáncer comienza en otro lugar del cuerpo y se expande hacia los pulmones.

Se trata de un cáncer nocivo que atienta contra la salud del paciente, llegando a ser un cáncer letal en muchos casos. Según [47], el cáncer pulmonar provoca mayor número de defunciones que el cáncer de mama, de colon y de próstata en su conjunto. Sin embargo, es importante tener en cuenta que el pronóstico dependerá del tipo de cáncer, su grado de expansión y su tamaño. En ocasiones, los tumores de pulmón no causan síntomas para las personas, y suelen ser detectados durante pruebas de imágenes realizadas por otras causas o motivos ajenos al cáncer de pulmón. Teniendo en cuenta que a menudo pasan desapercibidos en el historial clínico, puede ser útil desarrollar un modelo predictivo que pueda proporcionar información sobre la identificación de un posible tumor de pulmón. En base a [48], un diagnóstico precoz del cáncer de pulmón, cuando el tumor no ha tenido el desarrollo expansivo hacia otros tejidos, se logra un pronóstico favorable hasta en un 90% de los pacientes. De esta manera, se trata de un cáncer que a menudo puede ser peligroso para la salud de las personas, y se pretende mejorar el diagnóstico y contribuir al desarrollo de un tratamiento más efectivo.

El diagnóstico asistido por inteligencia artificial puede tener un impacto significativo en el ámbito social y en la atención médica de este tipo de tumores, por lo que también será incluido en el aprendizaje en esta investigación.

Capítulo 6. Conjuntos de datos

En esta sección se explicarán los conjuntos de datos utilizados en este proyecto. La elección de los conjuntos de datos es de alta relevancia teniendo en cuenta que determinará: primero como se preprocesará los conjuntos de datos, y segundo, porque deben ser válidos para entrenar las redes neuronales según los requisitos de datos. Además, otro aspecto para tener en cuenta es que debe tener un volumen de datos necesario para que el modelo pueda aprender. Se entrenará cada modelo con cada conjunto de datos, y finalmente, se mezclará todos los conjuntos de datos y se entrenará también cada modelo. Este conjunto de datos mezclados se denominará durante este proyecto como conjunto de datos mezclados, o por su equivalencia semántica en el inglés "*All dataset together*".

Se ha supervisado varias fuentes, y finalmente se ha decidido escoger *Kaggle* debido a la alta usabilidad ofrecida con el que se podrá alcanzar los objetivos dentro del tiempo estimado. Kaggle es una comunidad que reúne a los científicos de datos y así como programadores estudiantes o profesionales interesados en el aprendizaje automático. Entre sus características destacables, Kaggle permite a sus usuarios publicar, buscar y colaborar en conjuntos de datos; y también permite el desarrollo de ejecución de programas en la nube en un entorno comparable a herramientas como Jupyter. Como servicio destacable, también se permite la competición individual o en grupo por construir el algoritmo con mayor puntuación que resuelve un determinado problema [49].

Sin embargo, en este trabajo se utilizará el servicio de plataforma de conjuntos de datos públicos de Kaggle donde los usuarios comparten los conjuntos de datos.

6.1 Datos de entrenamiento para el tumor de riñón

Se ha escogido este conjunto de datos concreto [50] donde se declara dos colaboradores: el usuario MD NAZMUL ISLAM y el usuario Md Humaion Kabir Mehedi.

El conjunto de datos de riñón, durante este proyecto también denominado por su traducción al inglés como "*Kidney Tumor*", se recopiló de diferentes hospitales de Dhaka, Bangladesh, donde se puede identificar cuatro clases: tumor de riñón, quiste, piedras de riñón y un grupo final para representar imágenes de riñón sano.

En cuanto al procedimiento de recopilación de datos, se seleccionó imágenes desde un plano anatómico axial y coronal de estudios con y sin contraste en imágenes para todo el abdominal. Posteriormente, a parir de un estudio DICOM, se selecciona un diagnóstico uno por uno, y se crean los lotes de imágenes DICOM de la región previamente mencionada. Dado este lote con imágenes DICOM de las regiones de interés, se excluyen los metadatos del paciente y se convierte las imágenes DICOM a formato JPG manteniendo la calidad de las imágenes. Para validar los diagnósticos y comprobar que

la conversión se ha hecho correctamente, cada imagen fue verificada por un médico y un radiólogo. Todo este procedimiento se explica en su fuente [50].

Finalmente, el conjunto de datos contiene un total de 12.446 imágenes, de las cuales son 3.709 imágenes para la clase del quiste, 1.377 para la clase de piedras en el riñón, 2.283 para la clase de tumores y 5.077 para la clase de riñón sano.

DICOM es un estándar internacional para almacenar, transferir o procesar imágenes médicas, que además se incluye información o metadatos relacionados sobre el estado clínico. Es útil para transferencia de datos y procesamiento entre distintos dispositivos y bases de datos para propósitos médicos [51]. En la Figura 6-1 se puede observar distintas muestras del dataset, mostrando un extracto de cada clase: arriba a la izquierda se encuentra la clase de riñón con quiste, arriba a la derecha se puede ver la clase de riñón sano, abajo a la izquierda se observa la clase de riñón con piedra y abajo a la derecha se puede apreciar la clase de riñón con tumor.

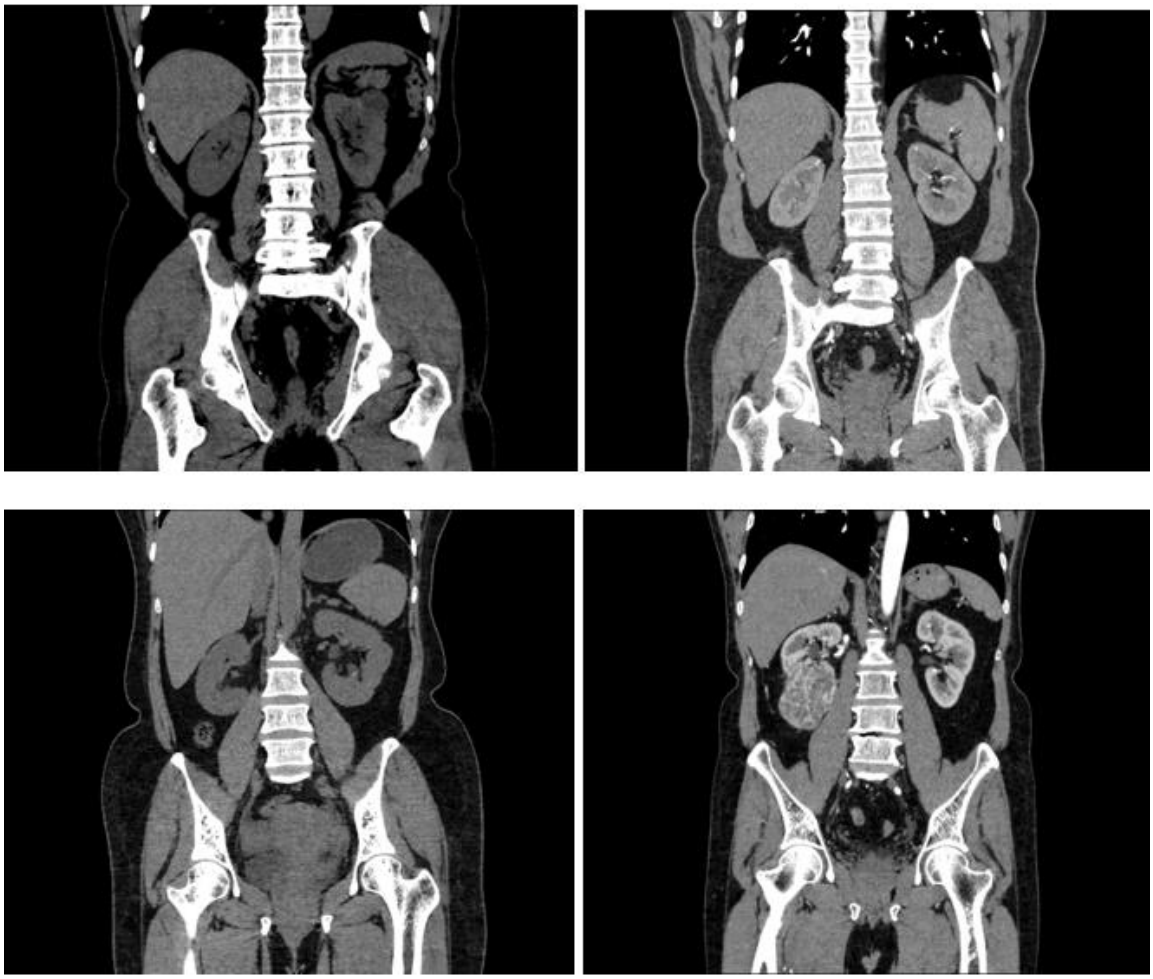


Figura 6-1: Muestra del conjunto de datos del riñón.

6.2 Datos de entrenamiento para el cáncer de cerebro

El segundo conjunto de datos escogido ha sido como parte de un conjunto de datos. La fuente original se encuentra en [52] y se puede identificar 8 tipos de cáncer distintos con un total de 5000 imágenes por subclases: leucemia linfoblástica aguda, cáncer cerebral, cáncer de mama, cáncer cervical, cáncer de riñón, cáncer de pulmón y colon, linfoma y cáncer oral. El formato de las imágenes es JPEG de dimensiones 512x512 píxeles.

Para este proyecto, se selecciona el cáncer localizado en el cerebro, durante este proyecto también denominado por su traducción al inglés como "*Brain*", donde se enfocará en la clasificación de las tres subclases de cáncer cerebral:

- Glioma: existen tipos de gliomas de crecimiento lento, y otros de crecimiento rápido que se les considera como maligno, detectarlo cuanto antes para intentar lograr un pronóstico más favorable [53].
- Meningioma: la mayoría no suelen ser cancerígenos y suelen tender a crecer lentamente. Su tasa de supervivencia es alta, y detectarlo de forma temprana puede ayudar su control de crecimiento [54].
- Tumor pituitario, o tumor de hipófisis: se considera como tumor benigno en la mayoría de los diagnósticos según [55].

Cada subclase contiene exactamente 5000 imágenes, siendo el formato de resonancia magnética en la clase de cáncer cerebral. En la Figura 6-2 se puede observar distintas muestras del dataset, mostrando un extracto de cada clase: arriba a la izquierda se encuentra la clase de cerebro con tumor de glioma, arriba a la derecha se encuentra la clase de cerebro con tumor de meninglioma, y, por último, por debajo de estas dos, se localiza la clase de cerebro con tumor de hipófisis.

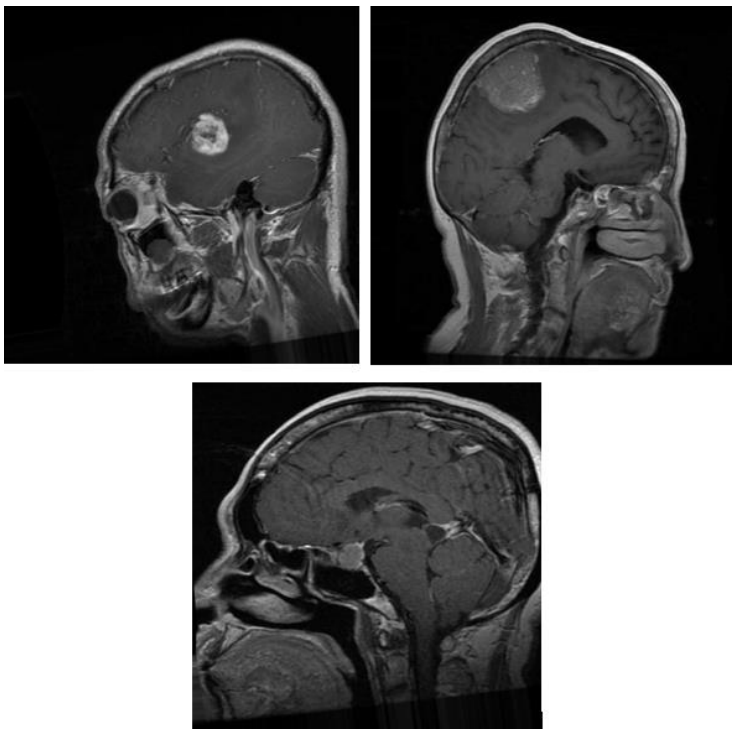


Figura 6-2: Muestra del conjunto de datos del cerebro.

6.3 Datos de entrenamiento para el cáncer de pulmón

Publicado por el autor Hamdalla F. Al-Yasriy, disponible en [56], se escoge este conjunto de datos de pulmón, durante este proyecto también denominado por su traducción al inglés como "*Lung*", que proporciona imágenes de Tomografía Computarizada (*CT – Computed Tomography*), recogidas por el Hospital de Enseñanza de Oncología de Irak (IQ-OTH) y el Centro Nacional para enfermedades del Cáncer (NCCD). Cabe destacar que las imágenes fueron escogidas por los especialistas de dichos centros durante los últimos meses de 2019 [57] [58].

Los escáneres fueron originalmente recogidos en formato DICOM, fueron preprocesados para que no se pudieran identificar a los pacientes, y contiene 110 casos que varían según la edad, el sexo, el área de residencia, el nivel de vida, entre otras variables.

Los 110 casos se pueden agrupar en tres clases con un total de 1190 imágenes que se compone en:

- Pulmón sano: 55 casos, 416 imágenes.
- Pulmón como tumor cancerígeno: 40 casos, 561 imágenes.
- Pulmón con tumor benigno: 15 casos, 120 imagen

En la Figura 6-3 se puede observar distintas muestras del dataset, mostrando un extracto de cada clase: arriba a la izquierda se encuentra la clase de pulmón con tumor benigno, arriba a la derecha se encuentra la clase de pulmón con tumor maligno, y, por último, por debajo de estas dos, se observa la clase de pulmón sano.

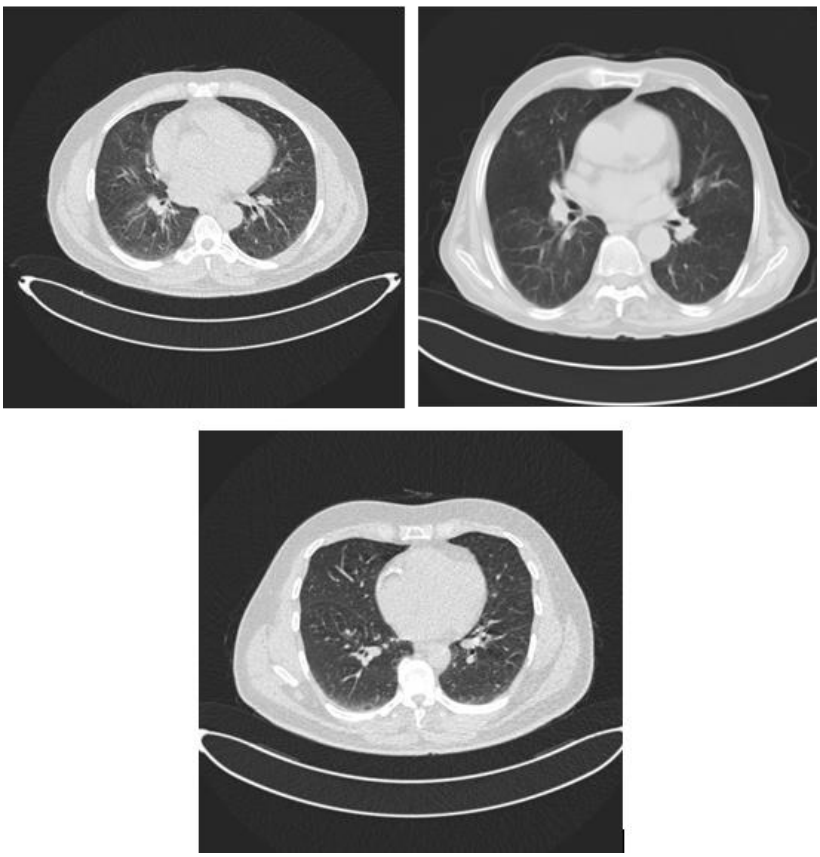


Figura 6-3: Muestra del conjunto de datos del pulmón.

Capítulo 7. Modelos de redes neuronales

La elección de arquitecturas o modelos de redes neuronales no supone una tarea sencilla para los investigadores o desarrolladores de software. En esta misma elección se tiene en cuenta distintas variables, desde aquellas que depende del contexto del problema hasta las limitaciones computacionales del equipo de investigación. En este estudio, se pondrá a prueba varios modelos y variantes que se basan en el modelo de Transformador de Visión, siendo una de sus aplicaciones principales el procesamiento de imágenes.

7.1 Origen del mecanismo de atención en Vision Transformer

Los modelos basados en Transformer se basan en un concepto denominado *Vision Attention*, el cual es una técnica cuyo algoritmo pondera el nivel de importancia en aquellas partes a partir de los identificadores (*tokens*) de entrada para que sean más relevantes en la decisión de la predicción [59] [60]. Aparece por primera vez en el artículo científico de Vaswani et al. (2017) *Attention Is All You Need* publicado en Advances in Neural Information Processing Systems [11], con el que aplica esta técnica en un traductor de idiomas para un modelo neuronal de procesamiento de lenguaje natural. También se buscaba una optimización del hardware en el procesamiento paralelo de los recursos de las GPU, y es por ello por lo que propusieron esta arquitectura con la que revolucionaron la inteligencia Artificial. Los resultados de los experimentos demostraron claramente la efectividad de este nuevo paradigma, logrando la precisión establecida como objetivo y manteniendo el coste computacional dentro de los límites esperados.

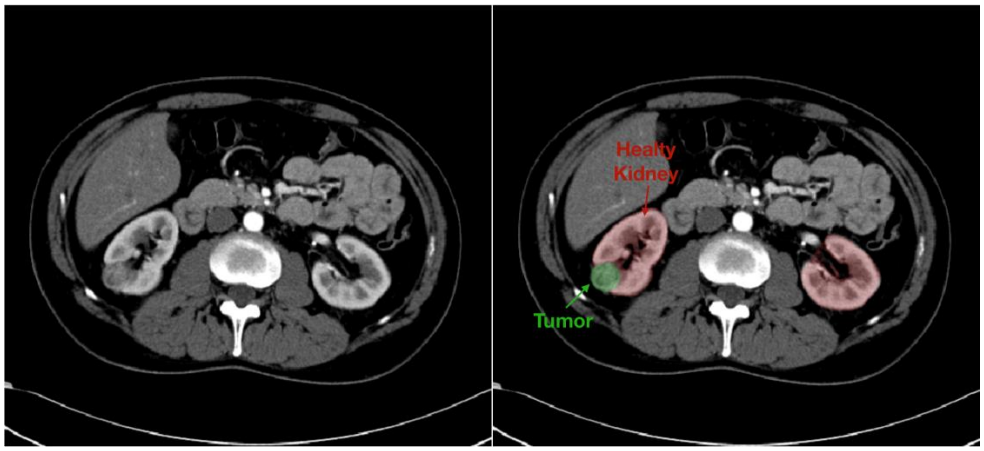


Figura 7-1: Segmentación de tumor de riñón de una imagen por TC. Fuente original: G. Santini [61].

En la Figura 7-1 se puede ver un ejemplo de lo que aplicaría la técnica de *Vision Attention*, perteneciente al conjunto de datos KITS. En la izquierda se observa la imagen original, y en la derecha, la imagen señalando las partes relevantes de los riñones en rojo, y el tumor en verde. En un caso práctico, el efecto en el algoritmo aplicaría un valor ponderado a nivel matemático dedicando más atención a las partes coloreadas en color rojo, y color verde, mientras que al resto de píxeles disminuiría su atención ponderada debido a que no tiene importancia en la decisión. El aprendizaje

supervisado permitiría el entrenamiento mediante el descenso del gradiente, esto es, mientras que al principio la red neuronal no sabría qué partes son las más relevantes en la atención, a partir de calcular el error y medir su distancia a la solución real, iterativamente se acercaría a la solución aplicando la atención a las partes más relevantes de la imagen y permitiendo una clasificación más precisa.

En aprendizaje profundo, el modelo de Redes Convolucionales es una de las principales herramientas de los investigadores en el procesamiento visual para resolver problemas de clasificación de clases. Históricamente han formado parte de este campo de estudio desde 1980, sin embargo, fue en el año 2012 cuando se logra un avance significativo con la publicación del modelo de AlexNet diseñado por Alex Krizhevsky en colaboración con Geoffrey Hinton y Ilya Sutskeve. Cabe destacar que este modelo alcanzó un error del 15,3% en el reto de ImageNet. Posteriormente, la industria comienza a adoptar estos desarrollos y continúa la mejora significativa como los logros obtenidos con GoogleNet con tasas de error de 0,06 en MNIST en 2015, o la publicación de las ResNet según [62], y también publicado en otras fuentes como [9].

Inspirado y escalando la técnica empleada en procesamiento de lenguaje natural, se combinas las redes convolucionales con *Vision Attention* (Wang et al., 2018; Carion et al., 2020) parcialmente, o incluso remplazando totalmente las redes convolucionales (Ramachandran et al., 2019; Wang et al., 2020). Según la teoría, eran modelos que podrían escalar la solución de forma efectiva pero no lograron los objetivos debido a los patrones especializado de atención según [63].

En 2021, se publica el primer Transformador de Visión (en el artículo "*An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale*" [63]), de tal manera que se introducía directamente las imágenes como token de entrada en parches o bloques de 16x16 píxeles. De esta manera, reduce el coste computacional, mientras se obtiene unos resultados excelentes en clasificación y segmentación de imágenes en conjuntos de datos como ImageNet o CICAR-100. La idea es descomponer las imágenes en bloques secuenciales, se transforman en vectores y procesarlos como palabras en lo que fue principalmente en el artículo propuesto [11]. De forma análoga, mientras en el procesamiento de lenguaje se atiende a las palabras relevantes y sus relaciones, en el procesamiento por visión se enfoca en las diferentes porciones o conjunto de píxeles más importantes.

7.2 Vision Transformer

En esta sección se comentará con detalle sobre el modelo denominado como Vision Transformer y **la contribución del mecanismo de control** al transformador de visión. Tal y como se explicó en la introducción del capítulo 7.1, un Transformador de Visión se basa en la idea de modificar lo mínimo posible de la arquitectura original empleada en el procesamiento de lenguaje natural, como bien expone [64] en su explicación sobre el origen, y funcionamiento básico del modelo ViT. Como entrada de dato al modelo (*input*) se proporciona imágenes de dimensión 2D, dichas imágenes se dividen en bloques o parches de píxeles en dimensiones de tamaño de un cuadrado (es decir, lado y base tienen los mismos números de píxeles: 14x14, 16x16, 32x32, etc.). En la Figura 7-2 se puede observar las operaciones que realiza el modelo antes de ser introducido al núcleo del modelo, donde cada parche se tratará como un token de entrada al modelo.

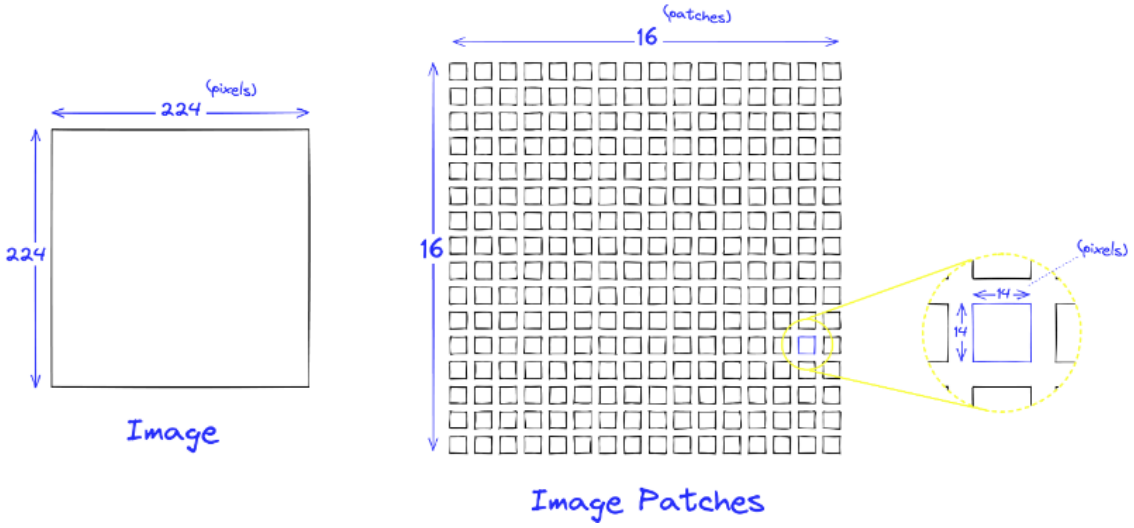


Figura 7-2: Conversión de una imagen en Vision Transformer de dimensiones de 224x224 píxeles divididos en parches de 14x14 píxeles, resultaría en total en 256 parches. Fuente: [65].

Dichos parches necesitan ser tratados como vectores de 1D para que puedan ser procesados como los tokens (*words*) de una arquitectura basada en procesamiento de lenguaje natural, por lo que se les aplica una operación de aplanamiento (*flatten*).

El Transformer utiliza un vector de tamaño fijo en todas sus capas, por lo que se le añade una capa predecesora que consiste en mapear una proyección lineal que toma los vectores aplanados y ajustar el vector aplanado a un tamaño fijo para que el modelo pueda trabajar de manera consistente a una dimensión constante. Con esta operación, se puede ajustar a una dimensión mayor o menor de los vectores previamente aplanados.

Posteriormente se aplica una técnica conocida como *embeddings* de posición, lo que permite poder retener la posición original de los parches en la imagen original. El token de la posición "0" de la Figura 7-4 se le denomina *CLS embeddings*, y consiste en un token que representa conceptualmente la imagen completa durante el entrenamiento y determina en el algoritmo la clase a la que pertenece.

Esta operación representa vectorialmente cada parche en un espacio multidimensional, y permite ser asociado con otros vectores. Se trata de una característica que se aplica al procesamiento de lenguaje natural con el objetivo de relacionar la semántica de las palabras, mientras que en los Transformadores de Visión se limita a preservar la información posicional de los parches. Sin embargo**, el mecanismo de atención global** es lo que permite al modelo relacionar cada parche y que el modelo pueda capturar las relaciones entre los parches. Se trata del concepto más relevante del modelo y necesario para comprender la base del algoritmo del Transformador de Visión.

Desde el tratamiento de los parches en dimensiones de un cuadrado, hasta el mecanismo de atención, fuentes como [66], [65] han resultado de gran utilidad para el entendimiento del modelo.

Entrando en mayor detalle, el siguiente paso del *embeddings de posición* conlleva a explicar el núcleo del Transformer, descripto gráficamente en Figura 7-3. En este bloque se identifica distintas capas y etapas en función de la arquitectura o algoritmo del Transformer:

Capas de normalización que se le aplica a las secuencias de los parches introducidos y permite adaptar y mantener el seguimiento del modelo ante la variación de los parámetros del entrenamiento [67].

Capas de atención multi-cabezas (MHA - *Multi-head Self-Attention Block*) que permiten al modelo relacionar y ponderar la relevancia de los pixeles de la imagen en un contexto global, internamente el concepto se le reconoce como mapas de atención y se pueden encontrar en forma matricial. El modelo captura las relaciones tanto a nivel local en cada parche como a nivel global relacionando la importancia de cada parche con el parche que se relaciona en cuestión [67].

Bloques de perceptrón multicapa (*MLP*), consiste en un modelo secuencial de red neuronal de al menos 3 capas totalmente conectas (capa de entrada, capas ocultas y capa de salida), tal y como se explica en el "Anexo II. Definiciones básicas". Destaca por tener la capacidad de resolver problemas no lineares [68].

- Para las neuronas se le aplica la función de activación de Unidad Lineal Rectificada (*ReLU*) o si se requiere introducir una componente no lineal, otra muy utilizada en los modelos Transformer es la función de activación Lineal de Error Gaussiano (*GeLU*). En el caso del artículo ya comentado "*An Image is Worth 16x16 Words*", se utiliza una capa de entrada, dos capas ocultas con activación de GeLU y la capa de salida [67].
- Al final del perceptrón multicapa se localizará la denominada cabeza del perceptrón multi-cabezas (es decir, la parte específica del modelo que produce la salida), y se le aplica una función *Softmax* para obtener una distribución probabilística de cada clase.

En el artículo "*An Image is Worth 16x16 Words: Transformers for Image Recognition At Scale*", se apreciar cómo sería la estructura de un Transformer estándar en la Figura 7-4.

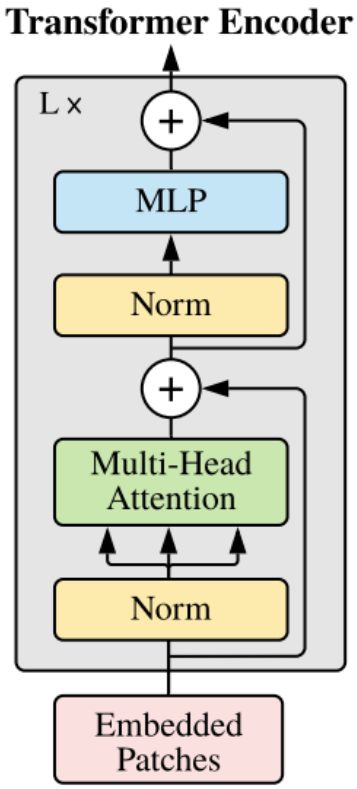


Figura 7-3: Estructura estándar del núcleo de un Transformer. Fuente: J. Manuel Cuesta Ramírez [69].

Finalmente, en la Figura 7-4 se plantea cómo quedaría finalmente un Transformer estándar. Existen diferentes variantes que consisten en diferentes capas del perceptrón multicapa, dimensiones de la proyección lineal del vector aplanado, y del número de capas de atención. De esta manera, en Pytorch

se dispone de distintas variantes: el modelo base, grande y gigante; cuyos detalles se pueden consultar en la Tabla 7-1.

Modelo	Capas	Capas ocultas	Tamaño del MLP	Cabezas	Parámetros (en Millones)
Vit-Base	12	768	3072	12	86
Vit-Large	24	1024	4096	16	307
Vit-Huge	32	1280	5120	16	632

Tabla 7-1: Detalles de las distintas variantes del núcleo estándar de un Transformer. Fuente: J. Manuel Cuesta Ramírez [69].

En este caso, debido a las limitaciones computacionales, se pondrá a prueba solo el modelo base "*Vit-Base*". A partir de la variante base, se pueden escoger entre dos variantes disponibles en la documentación oficial de Pytorch [70], que consisten en el "vit_b_16" y "vit_b_32", y la diferencia radica en el tamaño de los parches.

Se utilizará, por tanto, la arquitectura base utilizando parches de 16x16 píxeles y 32x32 píxeles ("vit_b_16" y "vit_b_32"), del que se deberá tener en cuenta que el modelo con parches más pequeño es de mayor coste computacional. Esto se debe a que el tamaño de la secuencia del Transformer es inversamente proporcional al cuadrado del tamaño del parche según [71], por lo que la perspectiva es que vit_b_16 consumirá más memoria de cómputo.

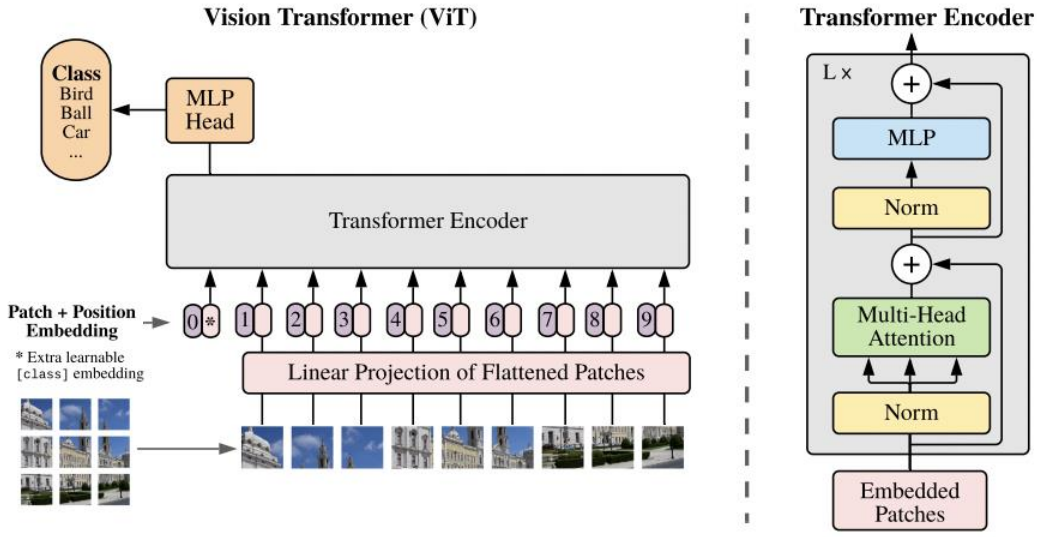


Figura 7-4: Arquitectura de un Transformer estándar. Fuente: J. Manuel Cuesta Ramírez [69].

7.3 Swin Transformer

En este apartado se introducirá a nivel básico sobre el modelo de Swin Transformer. La herramienta de Pytorch, que es la que se empleará en este proyecto, se permite utilizar dos variantes de los Swin Transformer: el modelo *base* que se construye en función de [72], y el modelo *V2* basado en [73]. En este trabajo se utilizará en práctica y se explicará su arquitectura únicamente sobre el modelo base, ya que el modelo *V2* requiere un rendimiento computacional inviable respecto a los recursos disponibles, en el capítulo 5 se puede revisar los recursos disponibles.

Hasta ahora se había propuesto el transformador de visión como un modelo de autoatención global que forma las imágenes de la secuencia de entrada en imágenes de una resolución fija para cada imagen, siendo la misma dimensión para cada parche durante todo el procesamiento de imagen. Como resultado, en los Transformador de Visión en la versión base se tiene un mapa de características que representa menos detalles visuales al no poder computar en detalles más profundos de las imágenes. Esto se debe a que el procesamiento a lo largo de las capas, su mapa de característica se mantiene en un tamaño fijo, no siendo capaz de profundizar en los detalles de los parches proporcionados.

Existe numerosas técnicas de extracción de detalles más profundos en imágenes como la segmentación semántica [74], pero requeriría reconocer y asociar la predicción a nivel de píxel. Estas características se hacen inviable aplicarlo en este tipo de modelos debido a que sus tokens de entrada al modelo son imágenes de alta resolución y su complejidad algorítmica es de nivel cuadrático en las variables del tamaño de imagen en la capa de autoatención, por lo que computacionalmente es inviable.

Como solución a este problema, se presentan los Swin Transformer que construyen una jerarquía de mapas de características. Estos mapas se inicializan en parches de pequeños tamaños y eventualmente se fusionan a lo largo del procesamiento de las capas del Swin Transformer. Esta operación presenta una complejidad lineal en virtud de lograr procesar cada ventana localmente sin llegar a solaparse en comparación a la complejidad cuadrática presente en los transformadores de visión base con la autoatención global. Esto se puede apreciar gráficamente en la Figura 7-5: en el apartado *a)* se puede apreciar la jerarquía de una imagen cómo se procesa en capas desde un inicio y fusiona al final de un Swin Transformer. En el apartado *b)* se puede observar cómo se obtiene una partición de una capa de Swin Transformer en ventanas. Estas ventanas ejecutan de manera local un grupo de los parches logrando autoatención local de complejidad lineal.

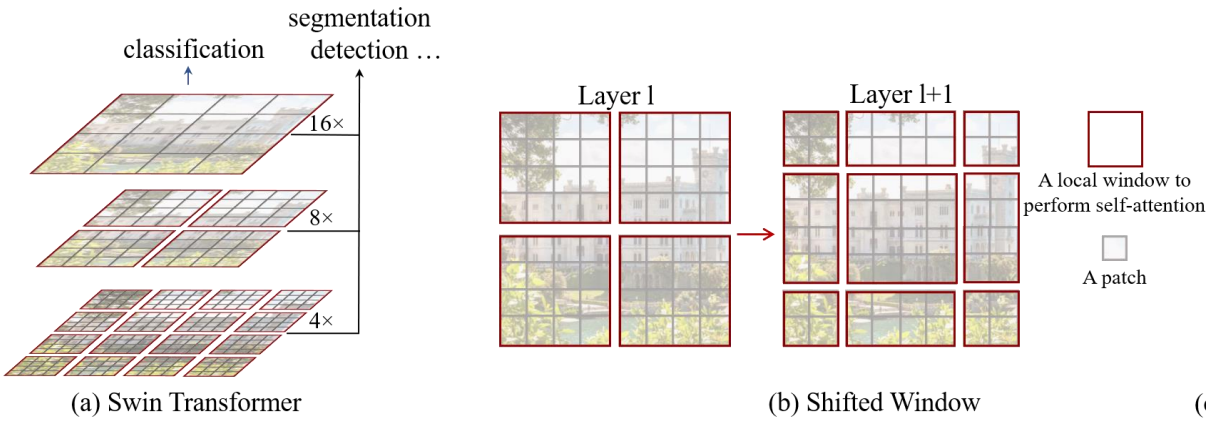


Figura 7-5: Capas de segmentación en un Swin Transformer. Fuente: Usuario "rishigami" [75].

En cuanto a la arquitectura, se proporciona al modelo una imagen RGB y se divide en parches tal y como en los modelos de transformadores de visión en la versión base.

Para explicar la arquitectura, se expondrá el modelo base expuesto en el artículo científico [72]. El tamaño de la entrada es de 4x4 píxeles y el canal es el 3 debido a que se trata de una imagen RGB. En el núcleo del Swin, donde se aplicarán varios bloques:

Fase 1: Se le aplica una capa de embebido lineal y se obtiene una proyección lineal de dimensión arbitraria "C". Posteriormente, los tokens de entradas son introducidos en varios bloques de Swin Transformer. Estos bloques mantienen el número de token original, y pueden ver su funcionalidad en

la Figura 7-6: *LN* es una capa de normalización, *SW-MSA* y *W-MSA* son módulos de autoatención basada en procesamiento local de ventanas.

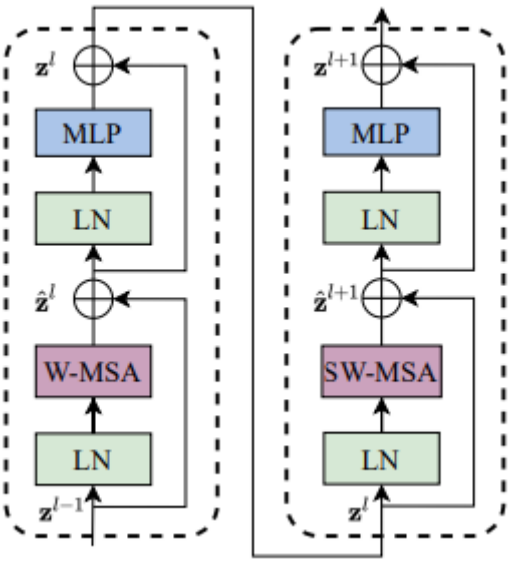


Figura 7-6: Representación de dos bloques sucesivos de Swin Transformer. Fuente: Z. Liu et al. [72].

Fase 2, Fase 3 y Fase 4: Con el objetivo se fusionar los parches y reducir el número de tokens, en estas fases destaca la concatenación de las características de cada grupo de parches con una resolución de agrupación “2x2” y aplicando una concatenación lineal de dimensión “2C”.

En la Figura 7-7 se puede verificar la arquitectura completa, y cómo en la fase 1 no se aplica redimensión, mientras que en la fase 2 se agrupa con una reducción 2x2 (H x 8 x W x 8), e iterativamente en la fase 3 (H x 16 x W x 16) y en la fase 4 (H x 32 x W x 32).

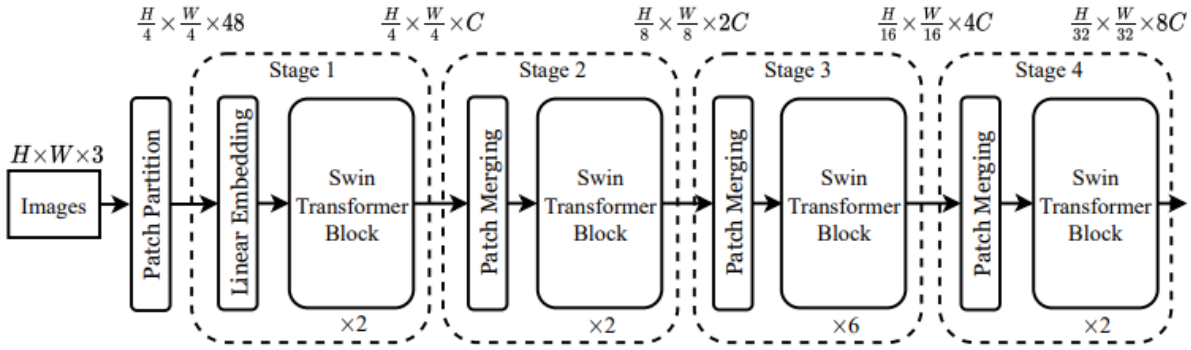


Figura 7-7: Arquitectura del Swin Transformer. Fuente: Z. Liu et al. [72].

7.4 MaxVit Transformer

En este apartado se expondrá la arquitectura Multi-Axis Vision Transformer o MaxVit presentado en [76]. El Transformer estándar siguen teniendo una falta de escalabilidad en el mecanismo de atención con respecto al tamaño de las imágenes, teniendo en cuenta la complejidad cuadrática en el Vision Transformer y complejidad lineal en el modelo Swin Transformer. MaxVit es una arquitectura que se publica en 2022 con el objetivo de introducir un modelo de atención más escalable el cual se le denomina mecanismo de multi eje de atención, y consisten en un bloque de atención local y otro bloque de atención global dilatada.

Este enfoque arquitectónico permite interacciones espaciales locales y globales en resoluciones de entradas arbitrarias y ofreciendo una complejidad lineal. Además, introduce una innovación al mezclar el mecanismo de atención con capas convolucionales pudiendo aprovechar las características y propiedades que tiene cada técnica. La novedad más destacable sería el enfoque estructural simple de repetir un bloque de construcción a lo largo de las etapas, manteniendo un bajo número de parámetros y logrando una mayor eficiencia computacional.

En cuanto a la arquitectura propuesta como base de un MaxVit presentado en [76], se puede dividir en dos fases:

Fase 1: Etapa Inicial S0 (*Stem*): se introduce la imagen en capas convolucionales reduciendo el tamaño de la imagen hasta la mitad. Las capas son de 3 canales, es decir, para imágenes RGB.

Fase 2: Etapa S1 hasta la Etapa S4: en cada etapa se aplica un módulo de MaxVit-SA.

- ***MBConv***: un bloque de inversión residual, diseñado para optimizar la eficiencia y aprendizaje de las redes neuronales, propuesto por primera vez en las arquitecturas convolucionales MobilieBetV2 [77].
- ***Block Attention***: diseñado para obtener la atención de manera local dentro de los bloques de la imagen. Esta capa permite capturar detalles entre pixeles cercanos entre sí.
- ***Grid Attention***: obtiene la atención global de manera eficiente al mismo tiempo que combina la atención local capturada en el bloque de atención local. Corresponde por tanto a una mezcla global espacial dilatada de los tokens de entrada. Esta combinación de los bloques de atención se puede observar en la Figura 7-8: el bloque de atención computa a nivel local en ventanas, mientras que el bloque de cuadrícula computa a nivel global en el espacio, permitiendo una mezcla de atención dilatada. Ambos bloques son de complejidad lineal.

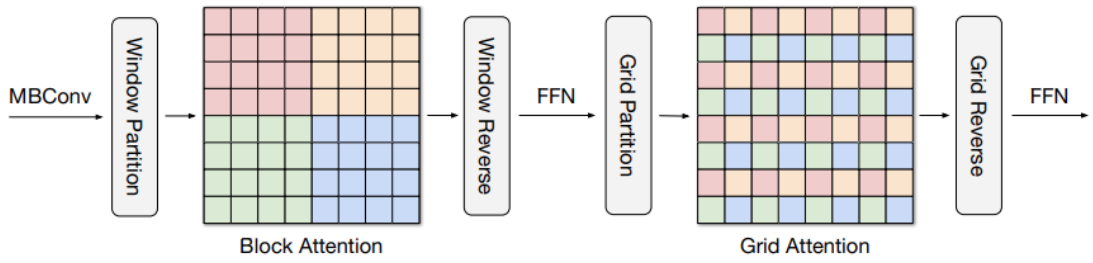


Figura 7-8: Resultado de la autoatención multieje. Fuente: Zhengzhong Tu et al. [76].

Finalmente, en Figura 7-9 se muestra tantos las fases como la arquitectura final de MaxVit.

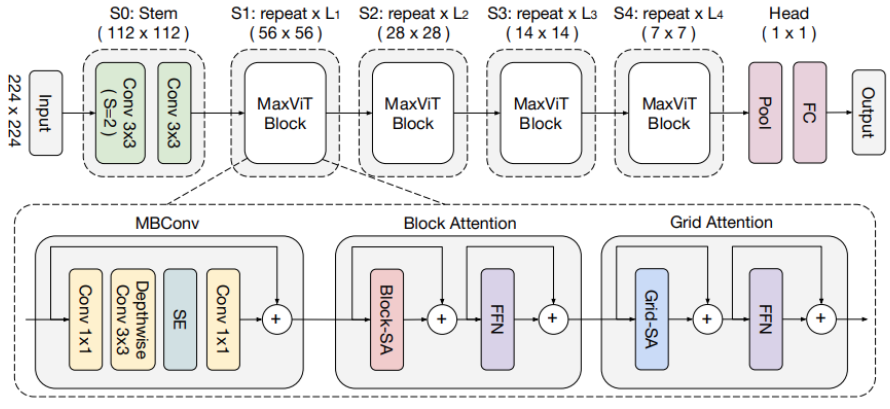


Figura 7-9: Arquitectura estándar de un MaxVit. Fuente: Zhengzhong Tu et al. [76].

7.5 Visión general de las arquitecturas y elección para puesta en práctica

Una vez expuesto y explicado las arquitecturas de vanguardia en la computación de visión, se debe seleccionar en la documentación de Pytorch qué modelo concreto será puesto en práctica.

En cuanto a los Transformadores de Visión en la versión base, se puede encontrar 3 variantes: la arquitectura pequeña, base y grande. Se deberá escoger la arquitectura pequeña debido a las limitaciones computacionales. Además, Pytorch proporciona que en las arquitecturas pequeñas se puedan utilizar parches de imágenes de 16x16 píxeles o 32x32 píxeles. Se hará pequeñas evaluaciones al principio de la investigación y se seleccionará una de estas opciones, ya que en la metodología empleada permite evaluar casos de prueba y verificar cuál ofrece mayores prestaciones, y continuar la investigación con el modelo que más se ajuste al problema. Por lo tanto, en un principio se escoge *vit_b_16* y *vit_b_32*.

En cuanto a los modelos de Swin Transformer, también hay 3 variantes: la arquitectura base enana, pequeña y base. De manera similar que, en los Transformadores de Visión, se pondrá a prueba estas tres opciones y se escogerá la de mayores prestaciones.

En cuanto al MaxVit, solo hay un modelo en la documentación, y es la opción que se escogerá.

Cuando se comenta que se tendrá en cuenta las prestaciones, se refiere a una relación tiempo de entrenamiento y precisión final. Aquellos modelos que requieran de mucho cómputo tanto en términos de tiempo como de carga de memoria en la gráfica (*GPU*) serán descartados.

En la Tabla 7-2, se puede observar que cada modelo será probado con sus características según la documentación de Pytorch.

Modelos	Parámetros del modelo	Operaciones de multiplicación y sumas (en Millones)	Tamaño aproximado total del modelo (en Megabyte)
VitB16	86,415,592	456.96	360.24
VitB32	88,185,064	456.96	344.59
Swin_t	28,288,354	123.43	141.22
Swin_s	49,606,258	208.40	236.33
Swin_b	87,768,224	363.69	397.41
MaxVit_b	814,464	545.68	22.07

Tabla 7-2: Características y detalles de las arquitecturas.

Capítulo 8. Evaluación de los modelos

8.1 Configuración de los modelos

8.1.1 Algoritmo de optimización en aprendizaje automático. Función de error

Una vez construido un modelo de aprendizaje, y aplicado una entrada que produce un resultado, se necesita calcular la diferencia de error que estima la distancia entre el error y el acierto de la predicción. Esta distancia la calcula la función de error o función de perdida, y el objetivo del algoritmo de optimización es minimizar el error.

El algoritmo comúnmente utilizado y que también ha sido empleado en este proyecto es el **optimizador de Adam,** que en la práctica se utiliza comúnmente en reconocimiento de imágenes. El algoritmo se trata de una versión del descenso por el vector gradiente con momento y el método *RMSProp* [78].

En cuanto a la función de error escogida, **la función Entropía Cruzada Categórica o Cross-Entropy** encajará con el planteamiento del problema que trata de clasificar modelos en base a la probabilidad de cada clase. Esta función dará como resultado un valor entre "0" y "1" representando una probabilidad para cada etiqueta o clase que se entrene en el modelo. Cuando el error es muy alto, la entropía cruzada penalizará en mayor medida aquellos valores que se alejan de las predicciones esperadas con el empleo de la aplicación del logaritmo en su fórmula descrita tal y como se discute en los foros de Pytorch [62] y [79] en Ecuación 1:

$$Loss = -\sum_{i=1}^{N} y_i \cdot \log \check{y}_i$$

Ecuación 1: Fórmula de Cross-Entropy utilizada por Pytorch.

8.1.2 Hiperparámetros

El procedimiento de aprendizaje de una red neuronal implica la configuración de varios parámetros externos a los parámetros internos de la red, son los llamados hiperparámetros. Estos hiperparámetros afectan de forma directa al aprendizaje del modelo, así como al rendimiento y efectividad. La elección de los parámetros desde un inicio es incierta, y se hace necesario un proceso iterativo para ajustar la combinación más efectiva. Desde un comienzo, se eligen parámetros estándar que, a medida que se realiza evaluaciones, se hace necesario puntuar un registro sistemático para identificar la configuración optima. Cabe destacar la explicación de los hiperparámetros en el Anexo III. Definiciones básicas.

El ajuste de los hiperparámetros es diferente para cada tipo de problema, arquitectura de red neuronal, conjunto de datos, complejidad y disponibilidad de recursos computacionales por lo que no existirá un número fijo de hiperparámetros, pero se comentará las observaciones que se considere relevantes y un rango de parámetros más frecuente en los entrenamientos en el apartado donde se expongan los resultados, y se puntuará los hiperparámetros exactos para los modelos con mayor efectividad. No obstante, para ofrecer una idea general de cómo se han ido ajustando los hiperparámetros, se expone un rango de valores que han sido utilizados en función del modelo y conjunto de datos:

- Tamaño de mini-lotes: de 4 a 32 imágenes por lote.
- Tamaño de imagen: siempre se escala a una dimensión fija de 224x224 píxeles.
- Canal de la imagen: siempre se ha aplicado tres canales, características de las imágenes de RGB.
- Tasa de aprendizaje: entre 10^{-3} y 10^{-5}, en función de la técnica de aprendizaje y modelo entrenado.
- Dilución (*dropout*): en la última capa se desactiva las conexiones de las neuronas entre 30 % y 40 %. Siempre estuvo activo un valor de dilución en dicho rango.
- Uso de conjunto de datos: varía demasiado según el volumen del *dataset* y Se modificó especialmente en los entrenamientos desde cero sin aplicar Transfer Learning, desde un 10% para conjuntos de datos grandes como el de cerebro o “Brain” hasta un 100% en conjuntos de datos pequeños como el del pulmón o “Lung”. En los entrenamientos de Transfer Learning se utilizó el 100% para que en los entrenamientos se obtuviera mayor rendimiento.
- Aleatoriedad de mini-lotes (*shuffle*): Se aplica para los conjuntos de datos de entrenamiento, y se desactiva para conjuntos de datos de validación y para conjuntos de datos de prueba.
- Épocas: desde las 5 hasta 200 épocas dependiendo del conjunto de datos, y del tamaño de mini-lotes aplicado.

8.1.3 Operaciones sobre el conjunto de datos

Los conjuntos de datos normalmente están preprocesados por la fuente que ofrece su utilización. Sin embargo, a menudo se hace necesario aplicar ciertas operaciones. Llegados a este punto, se puede aplicar distintas acciones según el contexto que se explicará a lo largo de este apartado.

Operaciones para ajustar y normalizar los datos al modelo, que en este proyecto se hará necesario operaciones de ajuste de tamaño. Independientemente del tamaño de origen, se preparará y se ajustará el conjunto de datos sobre un tamaño fijo de 224x224 píxeles para cada imagen. La elección de este tamaño se debe a que todas las arquitecturas escogidas tienen de tamaño de imagen de entrada fijo en 224x224 píxeles.

Durante la etapa de Preparación de datos, se buscó un conjunto de datos que tenga ciertas características como que las imágenes sean cuadradas, es decir, altura y anchura de mismos píxeles; y, además, que el tamaño de las imágenes no fueran demasiados lejos del tamaño fijo pensado en 224x224. El motivo de estas decisiones se debe a que realizar un cambio de tamaño con dimensiones de altura y anchura no parejas distorsionan la proporcionalidad de la imagen, y, además, un escalado de imágenes de tamaños muy grandes también provoca una pérdida de calidad de imagen. Se busca que los conjuntos de datos sean imágenes de calidad para que el modelo pueda aprender los detalles en píxeles.

Otras operaciones necesarias son la codificación de la variable categórica, que se codifica cada etiqueta de cada clase con un identificador numérico; o la división de datos en conjuntos de entrenamiento,

validación y prueba. Dependiendo del entrenamiento, se selecciona un porcentaje para cada conjunto de datos:

- Entrenamiento: comúnmente sobre un 60-80% del tamaño del conjunto de datos, se selecciona con el objetivo de que el modelo realice el aprendizaje.
- Validación: comúnmente sobre un 5-15% del tamaño del conjunto de datos, se selecciona para evaluar durante cada paso hacia adelante (*step)* y final de época (*epoch)*, con el objetivo de monitorizar el aprendizaje, entre otras tareas, por ejemplo, para verificar en tiempo real si existe sobre ajuste (*overfitting*).
- Prueba o test: comúnmente sobre un 5-15% del tamaño del conjunto de datos, se selecciona con el objetivo de evaluar al modelo una vez el entrenamiento ha finalizado, con el objetivo de verificar si el entrenamiento ha sido exitoso. Este conjunto de entrenamiento en ningún momento se introduce en el modelo durante el entrenamiento o aprendizaje, además, es clave para verificar si realmente ha aprendido los detalles del entrenamiento.

Con respecto a las operaciones para aumento de datos (*data augmentation*), se pueden aplicar operaciones de aumento de datos con dos objetivos: o bien, cuando se trata de un conjunto de datos reducido y se necesita mayor volumen de datos, o bien, para generar características en el conjunto de datos para evitar el sobre ajuste (*overfitting*). Algunas de estas operaciones podrían ser la de aplicar transformaciones de rotación sobre un ángulo aleatorio, aplicar transformaciones de volteo vertical u horizontal, o aplicar transformaciones de ampliación (*zoom*), entre otras operaciones. El propósito de esta técnica es para introducir en el aprendizaje tanto el conjunto de datos original como el resto de las transformaciones. Con estas transformaciones se obtiene un mayor volumen de datos, y, además, el modelo debe aprender las características de la imagen para que no aprenda los píxeles de memoria (*overfitting*). Sin embargo, aunque en un principio tiene características atractivas, en esta investigación se decidió tal y como se describe en la metodología aplicar **evaluaciones sin aumento de datos**, y al observar los resultados de las evaluaciones se concluyó durante los ciclos o etapas de trabajo que no era determinante aplicar este tipo de operaciones. En su lugar, se hizo necesario disminuir el volumen del conjunto de datos en algunos entrenamientos para aumentar la eficiencia en tiempo del entrenamiento e incluso para obtener mejores precisiones al evitar resultados de *overfitting*.

En cuanto a las operaciones para balanceo de clases no fueron necesarias durante esta investigación.

8.1.4 Modelos preentrenados. Transfer Learning

Un modelo preentrenado es un modelo que ha sido entrenado previamente para una tarea, y que luego se puede utilizar como punto de partida para ajustarla en el aprendizaje de otras tareas en conjuntos de datos distintos. Normalmente, este ajuste requiere de reconfigurar la cabeza del modelo, es decir, de la última capa del modelo. Posteriormente al ajuste de la capa final, se procede al entrenamiento de dos maneras: o bien, se activa el aprendizaje en todas las neuronas del modelo, o bien, se desactiva el aprendizaje en todos los parámetros excepto en la conexión de las neuronas que conecta la penúltima capa con la cabeza del modelo.

Se ha desarrollado hasta tres configuraciones diferentes para comprobar el rendimiento de los distintos modelos. Una vez decidido los modelos a investigar, todos los modelos han sido puesto a prueba en distintos contextos distintos de entrenamiento:

- Entrenamiento desde cero, sin carga de modelo preentrenado.
- Entrenamiento con carga de modelo preentrenado, y congelando los parámetros de red excepto la cabeza del modelo. A este procedimiento se le denomina Transferencia de Aprendizaje, o por su terminología en inglés *Transfer Learning*.

- Entrenamiento con carga de modelo preentrenado, y desbloqueando el entrenamiento de todos los parámetros.

En cuanto al modelo entrenado, se ha cargado los parámetros del modelo IMAGENET1K_V1 en la documentación de Pytorch en [80], el cual ha sido entrenado para 1000 clases diferentes. Se trata del modelo de red convolucional llamado AlexNet entrenado sobre el conjunto de datos ImageNet, y que ha sido adaptado para reproducir los mismos resultados que la implementación de AlexNet cada arquitectura como son los Transformadores de Vision base, Swin Transformer y MaxVit según [80].

8.2 Resultados de los modelos

En este apartado se describirán los resultados finales para cada arquitectura con la precisión más alta lograda dentro de un tiempo de entrenamiento razonable. En la **Tabla 8-1** se puede observar una primera ronda de pruebas de evaluaciones para verificar la mejor variante o subtipo de las arquitecturas de Vision Transformer y Swin Transformer, tal y como se describió en el apartado 7.5 "Visión general de las arquitecturas y elección para puesta en práctica".

En esta ronda de evaluación, únicamente se puso a prueba con el conjunto de datos del riñón "Kidney Tumor", ya que el objetivo era obtener una retroalimentación ágil (*feedback*) y no se considera necesario poner cada conjunto de datos. En la Tabla 8-1: Comparativa entre las subvariantes de los modelos se describe la mejor precisión alcanzada en el conjunto de datos de validación, y la marca de tiempo en la época que alcanzó un máximo relativo en el conjunto de datos de validación. Se considera que este máximo relativo **puede ser un tiempo orientativo y aproximado** para lograr el resultado obtenido. Se trata de un entrenamiento completo de extremo a extremo sin carga de parámetros preentrenado.

Modelos	Mejor Precisión en validación (en %)	Parámetros del modelo	Operaciones de multiplicación y sumas (en Millones)	Tamaño aproximado total del modelo (en Megabyte)	Tiempo total de entrenamiento
vit_b_16	92,4	86,415,592	456.96	360.24	40m
vit_b_32	96,51	88,185,064	456.96	344.59	16m
Swin_t	94,6	28,288,354	123.43	141.22	46m
Swin_s	92,3	49,606,258	208.40	236.33	1h 21m
Swin_b	91,17	87,768,224	363.69	397.41	9h 28m
MaxVit_b	98,5	814,464	545.68	22.07	1h 2m

Tabla 8-1: Comparativa entre las subvariantes de los modelos

Una vez completada y analizada la Tabla 8-1, se observa con una predominancia clara de las subvariantes tanto en resultados de precisión obtenido como en menor tiempo requerido de entrenamiento para alcanzar dichos resultados. De esta manera, se seleccionan los modelos con mejores prestaciones, y se continúa con el marco de trabajo hasta completar el entrenamiento. Dichos modelos seleccionados son:

Por un lado, se selecciona **"vit_b_32"**, porque obtiene un 4,44 % de mejor precisión en un tiempo 2,5 veces más rápido que "vit_b_16". Además, este "vit_b_32" es un modelo que, a pesar de tener mayor número de parámetros, ocupa menos espacio de memoria lo cual podría explicar el tiempo de entrenamiento menor y resultados más eficiente. Probablemente sea menor por eficiencia en la estructura del modelo, o tipos de datos y precisión diferentes.

También se selecciona **"swin_t"**, porque obtiene un 2,49 % de mejor precisión en un tiempo 1,76 veces más rápido que "swin_s", y un 3,76 % de mejor precisión en un tiempo 12,34 veces más rápido que "swin_b". Los tiempos de entrenamiento mayor es algo esperado debido al tamaño de los modelos. En cuanto a los resultados, el modelo más pequeño y que en menor tiempo de entrenamiento alcanza mejores resultados probablemente se deba a la complejidad de los conjuntos de datos. Puede ser que para este conjunto de datos no sea necesario un modelo con gran número de parámetros, es más, puede que el exceso de parámetros esté dando resultados de sobre ajuste (*overfitting*). Esto se debe a que, a mayor tiempo de entrenamiento, la precisión de validación tiende a un máximo relativo menor que el máximo relativo que en la arquitectura más pequeña, por lo que se forma una asíntota horizontal.

Finalmente, se escoge **"MaxVit_b"**, siendo una opción ajustable al problema. Es más, se observa mayor precisión que en cualquier otra arquitectura del 98,5% en un tiempo razonable de 1 hora y 2 minutos.

A partir de este punto y hasta el final de este documento, una vez en conocimiento los modelos específicos que se estudiará y sus características concretas, se referenciará por legibilidad los siguientes modelos de la siguiente manera: "vit_b_32" será identificado como Vit, "swin_t" como Swin, y "maxvit_b" como MaxVit.

Una vez comentada la ronda de reconocimiento para verificar qué modelos serán evaluados, se omite todas las rondas evaluadas excepto la última ronda cuyos resultados finales serán expuestos a continuación, siendo el resultado final de la investigación. Lo que se buscará es **qué modelo** obtiene mejores resultados, y **qué técnica** de aprendizaje es más efectiva.

Para determinar el modelo más efectivo, se ha decidido recoger los resultados de la Tabla 8-2 y Tabla 8-3 y se obtendrá a partir de ella tres gráficos: Figura 8-1: Resultados del Entrenamiento completo sin modelo preentrenado, Figura 8-2: Resultados del Entrenamiento de solo última capa y Figura 8-3. En cada gráfico consistirá exponer cada conjunto de datos y los modelos entrenados por grupo, y cada gráfico corresponde a cada técnica de aprendizaje empleada. A continuación, se muestran las tablas, y posteriormente se procederá a analizar los resultados de cada conjunto de datos según la técnica empleada.

	Entrenamiento completo sin modelo preentrenado			Transfer Learning: Entrenamiento de solo última capa con modelo preentrenado: IMAGENET1K_V1		
	Vit	Swin	MaxVit	Vit	Swin	MaxVit
Kidney	91,42	**93,02**	92,20	97,85	**99,35**	98,1
Brain	81,51	92,4	**96,4**	94,79	95,2	**96,9**
Lung	98,17	**98,90**	95,98	96,3	**98,78**	97,2
All dataset together	89,90	91,62	**95,69**	96,47	**97,2**	96,63

Tabla 8-2: Comparativa entre las subvariantes de los modelos sin preentrenar y aplicando entrenamiento de solo última capa con modelo preentrenado

	Transfer Learning: Entrenamiento extremo a extremo con modelo preentrenado: IMAGENET1K_V1		
	Vit	Swin	MaxVit

Kidney	95,6	**99,94**	**99,83**
Brain	93,51	**99,68**	98,7
Lung	**97,78**	95,12	**98,78**
All dataset together	97,43	**99,3**	97,5

Tabla 8-3: Resultado final de los modelos preentrenado de extremo a extremo

Llegados a este punto, para "Figura 8-1: Resultados del Entrenamiento completo sin modelo preentrenado" se puede observar que, para cada conjunto de datos, destaca:

- El rendimiento inferior del modelo ViT.
- MaxVit obtiene mayor rendimiento en el "All dataset together" y "Brain".
- Swin obtiene mayor rendimiento en el "Kidney" y "Lung".

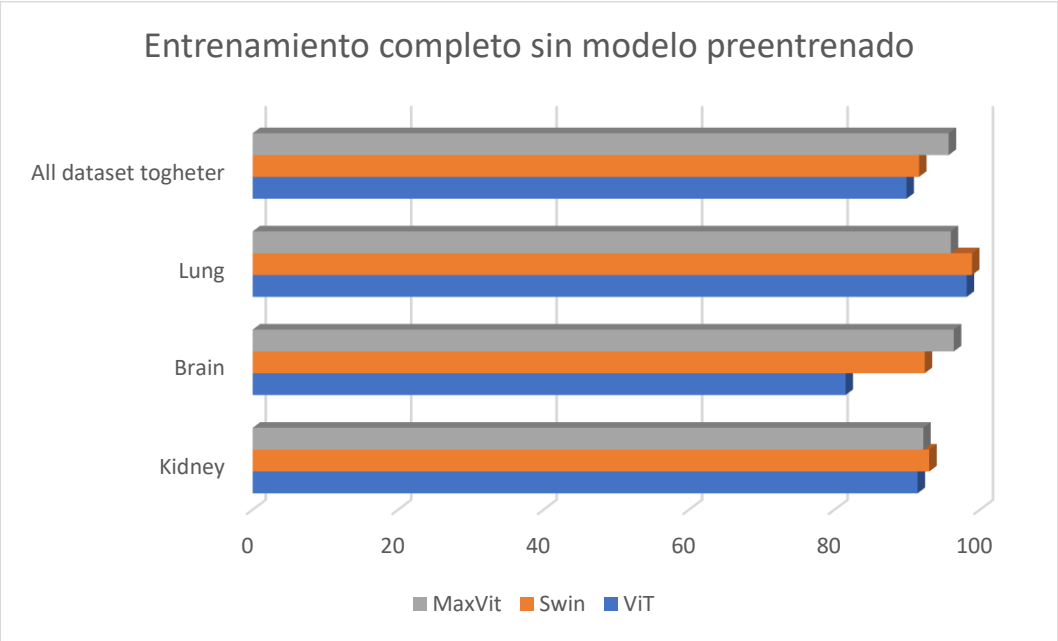


Figura 8-1: Resultados del Entrenamiento completo sin modelo preentrenado

En cuanto a los resultados de la "Figura 8-2: Resultados del Entrenamiento de solo última capa", destaca que:

- El rendimiento inferior del modelo ViT.
- MaxVit obtiene mayor rendimiento en el "Brain".
- Swin obtiene mayor rendimiento en el "All dataset together", "Kidney" y "Lung".

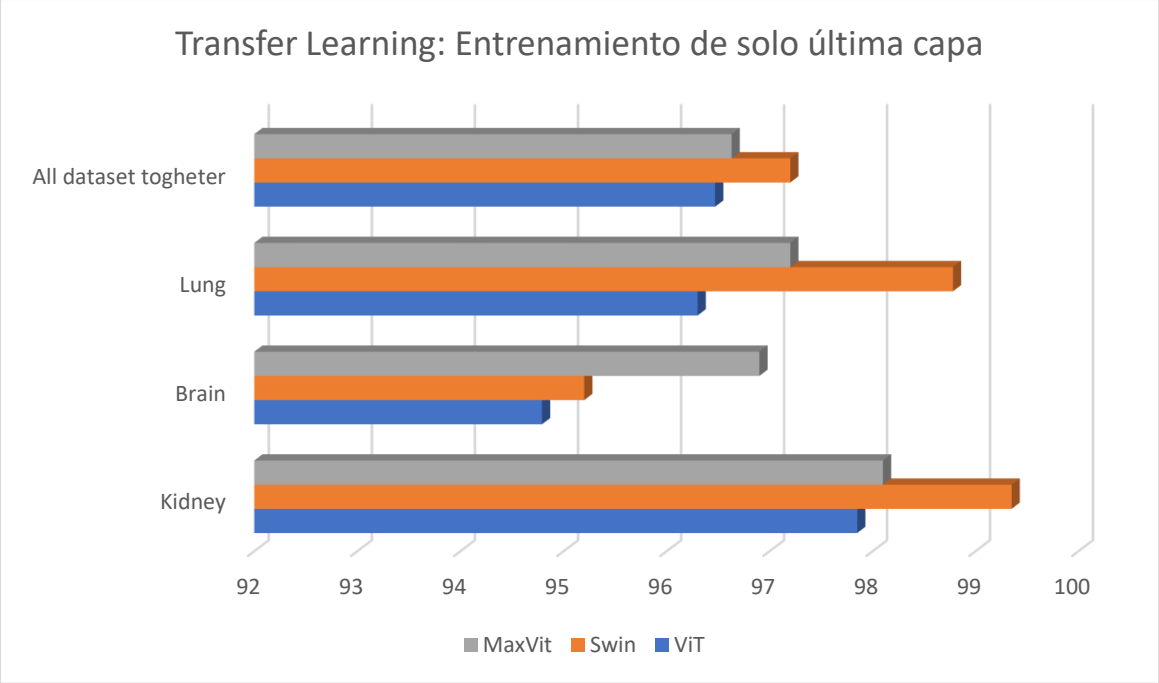


Figura 8-2: Resultados del Entrenamiento de solo última capa

Para finalizar, en cuanto a la "Figura 8-3: Resultados del Entrenamiento de extremo a extremo", destaca:

- El rendimiento inferior del modelo ViT.
- MaxVit obtiene mayor rendimiento en el "Lung".
- Swin obtiene mayor rendimiento en el "All dataset together", "Brain", "Kidney" y "Lung".

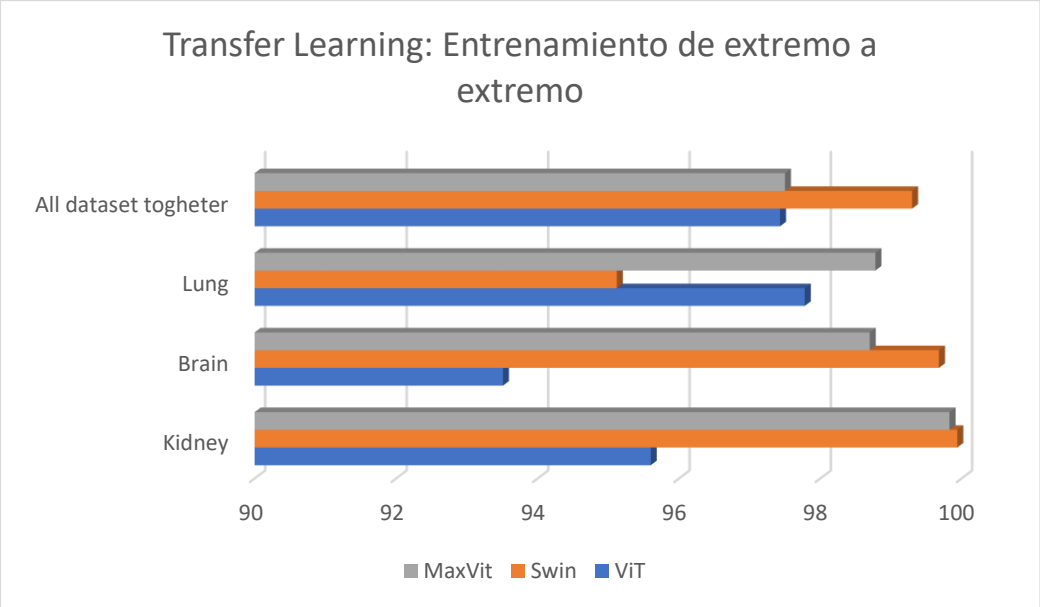


Figura 8-3: Resultados del Entrenamiento de extremo a extremo

En primer lugar, de este análisis, **se pondrá en estudio a los modelos.** Las observaciones generales de estos resultados de estos modelos para el análisis de imágenes médicas sugieren que el modelo base de Vision Transformer manifiesta un menor rendimiento y podría no ser la opción óptima para la detección y clasificación de imágenes médicas. En cuanto al modelo MaxVit y Swin Transformer

muestran mayor efectividad y fortaleza en sus decisiones. Con precisiones de alto rendimiento desde 96 % y 99 % aproximadamente logran beneficios sustanciales en tareas de clasificación de imágenes médicas, e invita al optimismo para trabajos futuros.

No obstante, se pretende diferenciar qué modelo computa de forma más efectiva. A partir de este punto:

- MaxVit destaca por tener una efectividad similar o superior a Swin en los entrenamientos sin modelo preentrenado, siendo capaz de generalizar los detalles de forma más global y es capaz de captar los detalles espacialmente. Esto probablemente se deba a su modelo arquitectónico, en función de la premisa de que el mecanismo de atención de MaxVit aplica distintos bloques de atención logrando una atención más global que el modelo Swin. Aunque ambos utilizan ventanas de autoatención en ventanas pequeñas, se diferencian en que MaxVit permite capturar relaciones a larga distancia que el modelo de Swin Transformer. Por tanto, este modelo es potente y robusto de entrenamientos donde se requiera una generalización fuerte en tareas complejas. En este caso, este modelo ha obtenido los mejores resultados en el aprendizaje a partir de conjuntos de datos mezclados “All dataset together” o el conjunto de datos del cerebro “Brain”.
- Swin Transformer ha demostrado ser una opción más versátil que MaxVit en el contexto de esta investigación. Aplicando técnicas de aprendizaje por Transfer Learning y de entrenamiento de extremo a extremo, demuestra mayor rendimiento en múltiples escenarios mientras que en el entrenamiento sin modelo preentrenado obtiene resultados similares. El enfoque arquitectónico del Swin Transformer es capaz de captar mayores detalles a nivel local que el MaxVit, y por este motivo es probable que el Swin Transformer sea más flexible que el MaxVit siendo capaz de adaptar el aprendizaje de otros modelos como es el modelo preentrenado IMAGENET1K_V1. Este modelo se beneficia particularmente de partir del entrenamiento de otros modelos y aplicarlos a su contexto a través del mecanismo de atención. Esto es posible debido al formar capas jerárquicas, aplicando a la autoatención en escalas más pequeñas que MaxVit, y logrando ajustar más eficiente las características finas o de escala menor en modelos ya preentrenados. Swin obtiene un excelente rendimiento todos los conjuntos de datos “All dataset together”, “Brain”, “Kidney” y “Lung” ajustando los parámetros a partir del modelo preentrenado IMAGENET1K_V1.

Para continuar con el análisis, ahora se deberá estudiar **la estrategia de entrenamiento**. Se han seleccionado los resultados obtenidos de haber entrenado el conjunto de datos mezclados “All dataset together”, que consiste en mezclar todos los conjuntos de datos. En la Figura 8-4 se ilustra los resultados de precisión para cada técnica de entrenamiento: de color verde se aplica a la técnica de entrenamiento completo sin modelo preentrenado, de color azul para transfer learning (entrenamiento de última capa con modelo preentrenado) y en color rojo, entrenamiento extremo a extremo con modelo pre entrenado.

Asimismo, se compara la misma subvariante del mismo modelo, pero con los resultados obtenidos de las distintas técnicas de aprendizaje. Desde este punto de vista, en todos los casos se observa que el entrenamiento de extremo a extremo con modelo entrenado se obtiene mejor resultado que en las otras técnicas, demostrando que es capaz de aprovechar características preentrenadas de otros entrenamientos y aplicarlo con mayor efectividad a las tareas de detección y clasificación médica.

En cuanto al ajuste de parámetros de extremo a extremo en lugar de ajustar únicamente la última capa, en cualquier caso, sigue siendo determinante en este tipo de tareas. A partir de este hecho,

entrenar la última capa podría ser suficiente para obtener resultados confidentes, mientras que el entrenamiento de extremo a extremo obtendría resultados óptimos en escenarios más complejos como el de mezclar todos los conjuntos de datos, pero a costa de un mayor costo computacional. Es más, incluso comparando un modelo como es la arquitectura base del Vision Transformer (ViT) que es más ***ineficiente*** en términos computaciones y cuyo algoritmo es ***teóricamente*** menos efectivo, aún obtiene mejor rendimiento en resultado que los modelos Swin Transformer y MaxViT. Sin embargo, el entrenamiento de extremo a extremo obtiene mayores resultados indiferentemente del modelo seleccionado.

Como conclusión, hay un claro dominio en el aprendizaje, se observa que el entrenamiento de extremo a extremo con modelo entrenado se obtiene un mejor resultado y resalta la importancia de lo influyente y crucial que es la técnica de aprendizaje capaz de aprovechar las características preentrenadas a gran escala.

En el siguiente apartado se deberá investigar y comparar más a fondo las razones del bajo rendimiento de ViT, así como las diferencias entre Swin y MaxVit.

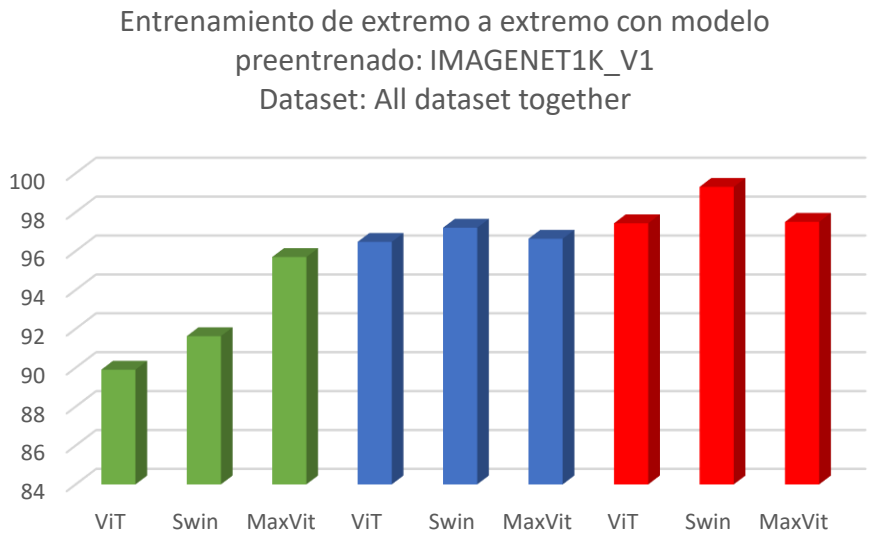


Figura 8-4: Entrenamiento de extremo a extremo con modelo preentrenado.

8.3 Comparaciones de resultados y conclusiones.

En este apartado se realizará un análisis comparativo de los modelos basados en los resultados obtenidos en el apartado 8.2 para diferentes métodos de entrenamiento y conjuntos de datos. Se tendrá en cuenta las evaluaciones a partir del entrenamiento con modelo preentrenado por los siguientes motivos: por un lado, porque en esta investigación se ha visto una predominancia en la técnica de aprendizaje, siendo una de las claves para obtener la mejor efectividad; y, por otro lado, porque en un contexto de mayor escalabilidad y empresarial es más atractivo un entrenamiento con modelos preentrenados.

A continuación, se cita textualmente una afirmación de Alfredo Ramos, vicepresidente la plataforma de IA Clarifai, que respalda el aprendizaje con modelos preentrenados antes que empezar el entrenamiento desde cero:

"*Los modelos previamente entrenados pueden reducir el tiempo de desarrollo de las aplicaciones de IA hasta en un año y lograr ahorros de costos de cientos de miles de dólares.*" Fuente: [81].

A partir de aquí, teniendo en cuenta la relevancia de realizar el aprendizaje con modelos preentrenados, se debe considerar un listado de características:

Complejidad computacional y Eficiencia (se recomienda revisar el apartado 7.5 "Visión general de las arquitecturas y elección para puesta en práctica" donde se describe los recursos en memoria y operaciones aritméticas de los modelos). Swin Transformer es un modelo más eficiente en términos computacionales al reducir la cantidad de operaciones, pero a costa de ocupar mayor uso de memoria al aplicar la jerarquía de mapas de características en su mecanismo de atención, mientras que MaxVit utiliza menos memoria a costa de mayor número de operaciones mediante la combinación novedosa de atención global y local. Según las limitaciones de hardware se conviene un modelo u otro, por ejemplo, un sistema altamente paralelo se beneficiaría más de MaxVit al poder distribuir a mayor escala las operaciones aritméticas o la carga de trabajo. Otro ejemplo, un sistema con una memoria VRAM limitada podría ralentizar el entrenamiento en un modelo de Swin Transformer, al tener un cuello de botella entre la RAM principal, la VRAM compartida y 0dedicada de la GPU generaría una latencia adicional en el caso de que el modelo y los datos de entrenamiento fuese mayor que la capacidad de almacenamiento de la VRAM.

Por lo general, se podría proponer a MaxVit como un modelo más eficiente desde el punto de vista computacional. Habría que investigar más a fondo si la diferencia en eficiencia computacional marca un gran impacto como para descartar uno u otro modelo, ya que a nivel de efectivad Swin Transformer es ligeramente superior.

Capturas de Detalles. En este punto hay que tener muy en cuenta el mecanismo de atención de cada modelo. Swin Transformer emplea un algoritmo de ventanas deslizadas de autoatención local, y genera capas jerárquicas al aplicar operaciones de reducción de resolución, lo que permite una extracción a nivel local y a escala global. Sin embargo, MaxVit utiliza un mecanismo de atención de autoatención local que posteriormente relaciona cada ventana de forma global, pero con la diferencia de que incluye etapas de generalización global mayor que Swin Transformer, por lo que se especializa más en enfocarse en capturar detalles a mayor lejanía espacial. A partir de este punto, hay que considerar que los tumores y enfermedades estudiadas en esta investigación son anomalías que se generan como un conjunto de pixeles en una localización única, por lo que no es estrictamente relevante tener un enfoque que permite generalizar dependencias globales. En cambio, se requiere de un enfoque de atención más local, capaz de detectar detalles finos, por lo que se saca las siguientes conclusiones: **desde el contexto de detección de enfermedades**, Swin Transformer podría ser mejor opción al ser capaz de capturar dependencias locales y finas, en contra del MaxVit que considera los detalles en un plano más genérico. **Dicha conclusión cabe resaltar que se enfoca en las tareas de enfermedades médicas bidimensionales**, ya que en cualquier otro contexto MaxVit podría presentar mejor efectividad.

Adaptabilidad y Flexibilidad. Esta característica tiene estrecha relación con la técnica de aprendizaje. Swin Transformer, al tener un enfoque más fino en su mecanismo de atención es capaz de aprender a partir de otras tareas y posteriormente aplicarla a la tarea de detección de enfermedades. No obstante, MaxVit tiene también esta capacidad, pero no logra tanta efectividad como Swin Transformer, probablemente porque no es capaz de aprender con tanta efectividad los detalles finos que se generan de forma local. Esta característica es muy importante para un contexto empresarial, proporcionando la escalabilidad de un modelo predictivo, así como para poder adaptarlas a tareas más complejas. En este punto se debe aclarar que el ajuste fino de los parámetros es requisito para la tarea de esta

investigación, y los resultados obtenidos así como sus conclusiones se deben por tener un conjunto de datos que se requiere la captura de detalles locales, por lo que habría que investigar si MaxVit obtiene mayor rendimiento en la adaptabilidad y flexibilidad, pero en una tarea de comprensión del contexto global.

MaxVit podría no captar los detalles finos al tener un enfoque más global, siendo crucial para marcar la diferencia en entrenamientos de modelos preentrenados, y también para aprender de conjunto de datos más simples como el del pulmón “Lung” o riñón “Kidney”. Además, Swin Transformer ha demostrado ser capaz de aprender de conjuntos de datos tanto complejos como simples, mientras que MaxVit ha sobresalido de los más complejos. Sin embargo, ha tenido un ligero rendimiento inferior para aprender en los conjuntos de datos más complejos en modelos preentrenados como “All dataset together” mientras que en los conjuntos de datos más simple tuvo ligeras mejoras (aunque no supera a Swin) como pulmón “Lung” o riñón “Kidney”.

Capítulo 9. Conclusiones y Trabajo Futuro

Las enfermedades, especialmente los tumores cancerígenos provocan incontables muertes en la humanidad por año, y para los pacientes que logran superar la enfermedad, posiblemente mantengan secuelas del tratamiento o la huella del cáncer. Una vez comprendida la estadística escalofriante sobre el cáncer, se hace evidente que la detección temprana es una tarea compleja pero necesaria para mitigar y eliminar en la medida de lo posible el daño directo o colateral de las enfermedades, prolongando la esperanza de vida de los pacientes. El trabajo elaborado en este documento pretende aportar conocimiento o entendimiento del problema a la causa en base a la experiencia de casos reales a partir de herramientas a disposición de los desarrolladores e investigadores.

De esta manera, se han desarrollado distintos modelos predictivos basados en tecnologías de vanguardia para imágenes bidimensionales denominada *Vision Transformer* y algunos de sus variantes. Se ha utilizado imágenes de resonancia magnética y de tomografía computarizada, tanto separado como juntos en un entrenamiento, entendiendo que el propio modelo debe ser capaz de realizar su tarea de predicción sin implicar el formato de la imagen en la decisión. Además, la naturaleza de los conjuntos de datos es distintas, por ejemplo, en el caso del cerebro se requerirá un modelo que capture detalles finos para poder clasificar los tipos de cáncer, mientras que en el caso del riñón y pulmón consistirá en la detección y clasificación de casos más genéricos.

Es relevante comprender las diferencias y propósitos de las variantes de este enfoque arquitectónico que se centra en el mecanismo de atención. Durante la investigación de puede identificar dos modelos predominantes: MaxVit y Swin Transformer. Partiendo de la base del objetivo del proyecto, se recogen los mejores resultados que se obtuvieron en las evaluaciones respecto al conjunto de datos donde se mezclan todos las enfermedades no correlacionadas y formatos de imágenes médicas de distinta naturaleza. Estos resultados obtenidos fueron de gran efectividad y rendimiento: en cuanto a MaxVit se logró una efectividad de 97.5 %, y en cuanto a Swin Transformer se logró hasta un extraordinario 99.3 %.

A pesar de que los resultados fueron de alto rendimiento en MaxVit y en el modelo base de Vision Transformer, Swin Transformer destaca en la mayoría de los resultados. En este proyecto se ha trabajado en la técnica del algoritmo de cada arquitectura, y en base a este estudio teórico y los resultados prácticos obtenidos, se concluye que Swin Transformer es el modelo más efectivo de los puestos a prueba para detectar tumores demostrando características excepcionales en el estudio de las enfermedades médicas. El enfoque de este modelo destaca es capturar detalles locales o cercanas entre sí por lo que es especialmente fuerte para aprender patrones de masas cancerígenas que son manifestaciones anómalas en los tejidos de forma conjunta y no dispersa globalmente en la imagen. En cambio, MaxVit, sí tiene el propósito de analizar globalmente los detalles en las imágenes, siendo su especialidad probablemente en otras tareas distintas a las estudiadas en esta investigación.

Se plantea algunas consideraciones para reforzar estos argumentos. Una de ellas podría ser mezclar conjuntos de datos perteneciente a una misma clase, por ejemplo, de cáncer de pulmón. La hipótesis que se plantea es que las imágenes del conjunto de datos utilizada en este proyecto tienen los bordes grises, y se plantea buscar otro conjunto de datos de pulmón donde los bordes no sean grises, y mezclarlos para que el mecanismo de atención no se fije en los detalles de los bordes, y sí en el tumor. Esto le resultaría especialmente útil a los modelos cuyo mecanismo de atención sea global, como MaxVit. Otro trabajo futuro que se plantea es el desarrollo de un modelo mixto entre MaxVit y Swin, capaz de aprovechar las ventajas de cada modelo, aunque pudiera ser una tarea que requiera una complejidad alta en términos de diseño y tiempo en desarrollarse. Por último, un posible trabajo futuro es añadir otro tipo de formato de entrada que no sean imágenes, como son la secuencia de ADN explicadas en el capítulo 2 en otros enfoques, pero también requeriría una adaptación en el modelo.

Anexo I. Pruebas

Para los modelos que han presentado una baja precisión, se probó utilizando un mini-lote (*batch size*) más pequeño que, por lo general, han presentado resultados de buen rendimiento, probablemente conduciendo a un entrenamiento más estable logrando una mejor convergencia. Sin embargo, esto no se ha observado en el rendimiento del modelo ViT, que no ha obtenido los resultados esperados aun aplicando este ajuste de hiperparámetro en el mini-lote de entrenamiento. Además, en el entrenamiento con modelos preentrenados ha resultado de utilidad incrementar o decrementar la tasa de aprendizaje para un ajuste de parámetros más fino, siendo estos dos hiperparámetros de gran relevancia e impacto en los entrenamientos. El resto de hiperparámetros, aunque también son importantes, no han sido tan determinantes para los resultados finales.

Por simplicidad, se presentará el resultado y las gráficas en cada modelo para los entrenamientos de ajuste de parámetros en la última capa realizados sobre el conjunto de datos donde se mezclan todas las clases únicamente en el entrenamiento con mayor efectividad, omitiéndose de esta manera aquellas evaluaciones realizadas y necesarias donde se ajustaban los hiperparámetros. En concreto, corresponde a los resultados de cada modelo de la Tabla 8-2 del apartado 8.2, a la columna “Transfer Learning: Entrenamiento de solo última capa con modelo preentrenado: IMAGENET1K_V1”

No se presentan los datos de entrenamiento realizados en el entrenamiento de extremo a extremo con ajuste en todos los parámetros porque el entrenamiento era de duración corta en algunos conjuntos de datos, y de pocas épocas donde no había contenido gráfico de suficiente valor como para poder ser analizado, por la poca información que proporciona. Tampoco se considera presentar los gráficos de los entrenamientos sin modelo preentrenado por simplicidad, y también, debido a los resultados dados han sido de poca relevancia.

Prueba del MaxVit.

Tiempo de entrenamiento 4.0 horas, 16.0 minutos, 14.83 segundos

Marca de tiempo para la mejor precisión de validación: 3.0 horas, 49.0 minutos, 14.58 segundos

Máxima precisión relativa de entrenamiento: 98.57 %

Máxima precisión relativa de entrenamiento de prueba: 96.63 %

Máxima precisión relativa de entrenamiento de validación: 96.79 %

Media de la Precisión de validación en la última época: 97.17 %

Mini-lotes: 8

Épocas: 30 (se interrupte antes porque el margen de mejora es mínimo en contra del tiempo consumido por época)

Máximo relativo de validación en la época: 19

Número de clases en el modelo: 10

Tamaño de imagen y canal: 3x224x224

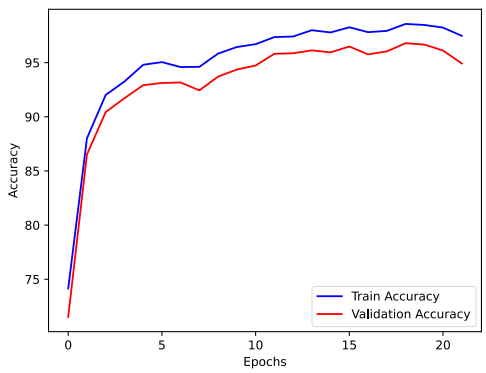
Tasa de aprendizaje: 0.0001

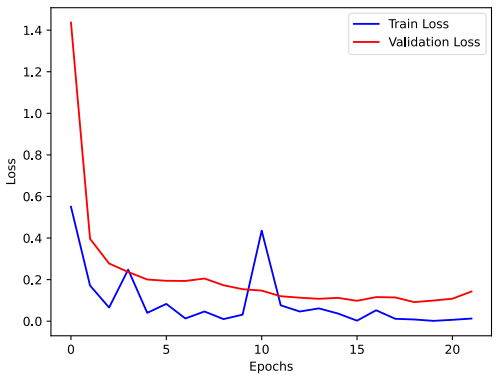
Porcentaje de uso del conjunto de datos para entrenamiento: 70.0%

Porcentaje de uso del conjunto de datos para entrenamiento de prueba: 15.0%

Porcentaje de uso del conjunto de datos para validación: 15.0%

Porcentaje de uso del conjunto de datos descartados: 0.0%





Prueba del Swin Transformer.

Tiempo de entrenamiento: 3.0 horas, 54.0 minutos, 10.85 segundos

Marca de tiempo para la mejor precisión de validación: 3.0 horas, 44.0 minutos, 21.06 segundos

Máxima precisión relativa de entrenamiento: 97.44 %

Máxima precisión relativa de entrenamiento de prueba: 97.17 %

Máxima precisión relativa de entrenamiento de validación: 96.96 %

Media de la Precisión de validación en la última época: 97,15 %

Mini-lotes: 16

Épocas: 200 (se interrupte antes porque el margen de mejora es mínimo en contra del tiempo consumido por época)

Máximo relativo de validación en la época: 27

Número de clases en el modelo: 10

Tamaño de imagen y canal: 3x224x224

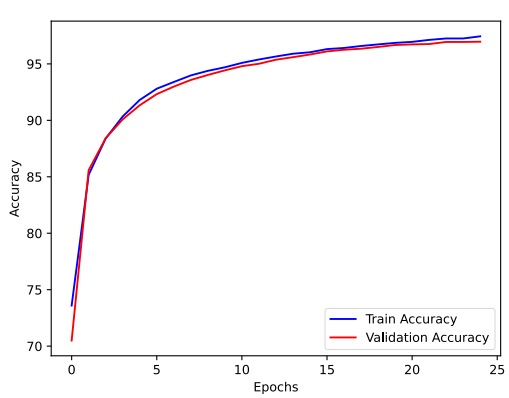
Tasa de aprendizaje: 0.0001

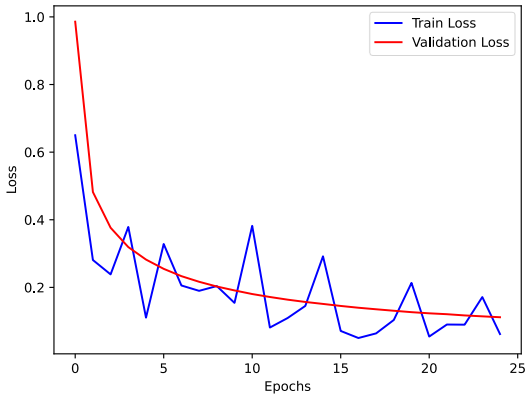
Porcentaje de uso del conjunto de datos para entrenamiento: 70.0%

Porcentaje de uso del conjunto de datos para entrenamiento de prueba: 15.0%

Porcentaje de uso del conjunto de datos para validación: 15.0%

Porcentaje de uso del conjunto de datos descartados: 0.0%





Prueba del Vision Transformer.

Tiempo de entrenamiento: 5.0 horas, 33.0 minutos, 32.20 segundos

Marca de tiempo para la mejor precisión de validación: 5.0 horas, 22.0 minutos, 6.09 segundos

Máxima precisión relativa de entrenamiento: 97.73 %

Máxima precisión relativa de entrenamiento de prueba: 96.47 %

Máxima precisión relativa de entrenamiento de validación: 96.70 %

Media de la Precisión de validación en la última época: 96.80 %

Mini-lotes: 16

Épocas: 50 (se interrupte antes porque el margen de mejora es mínimo en contra del tiempo consumido por época)

Máximo relativo de validación en la época: 37

Número de clases en el modelo: 10

Tamaño de imagen y canal: 3x224x224

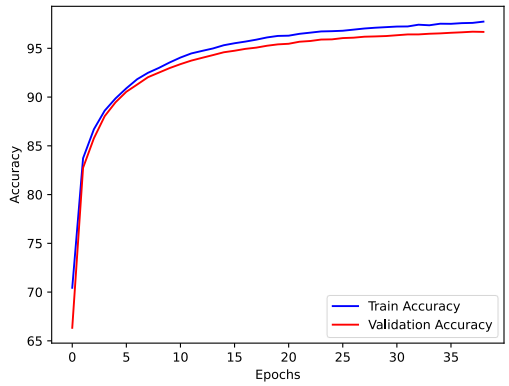
Tasa de aprendizaje: 0.0001

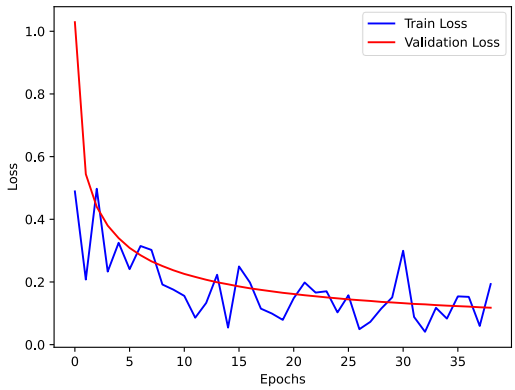
Porcentaje de uso del conjunto de datos para entrenamiento: 70.0%

Porcentaje de uso del conjunto de datos para entrenamiento de prueba: 15.0%

Porcentaje de uso del conjunto de datos para validación: 15.0%

Porcentaje de uso del conjunto de datos descartados: 0.0%





Otras pruebas, sin especificar hiperparámetros por simplicidad, para un conjunto de datos específico en el entrenamiento de ajuste de parámetros de última capa, ya que el entrenamiento extremo a extremo proporciona poca información en las gráficas:

Prueba del Vision Transformer.

Conjunto de datos: Pulmón

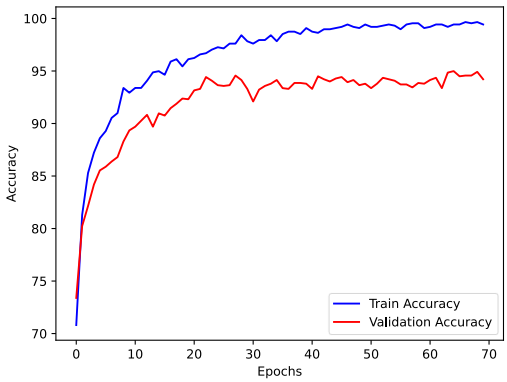
Tiempo de entrenamiento total: 19.0 minutos, 55 segundos

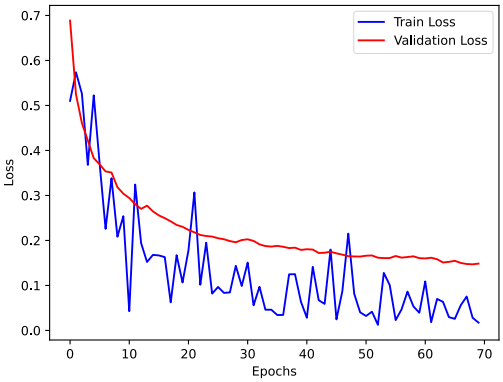
Marca de tiempo para la mejor precisión de validación: 6.0 minutos, 42.26 segundos

Máxima precisión relativa de entrenamiento: 99.65 %

Máxima precisión relativa de entrenamiento de prueba: 96.33 %

Máxima precisión relativa de entrenamiento de validación: 94.98 %





Conjunto de datos: Cerebro

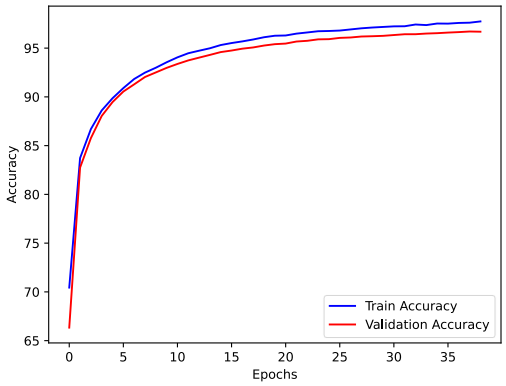
Tiempo de entrenamiento total: 2.0 horas, 42.0 minutos, 13.06 segundos

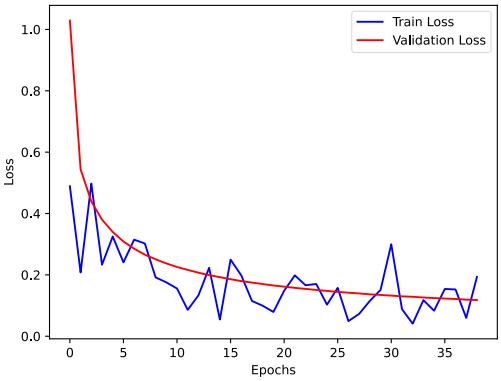
Marca de tiempo para la mejor precisión de validación: 2.0 horas, 28.0 minutos, 3.78 segundos

Máxima precisión relativa de entrenamiento: 96.39 %

Máxima precisión relativa de entrenamiento de prueba: 94.4 %

Máxima precisión relativa de entrenamiento de validación: 94.79 %





Conjunto de datos: Riñón

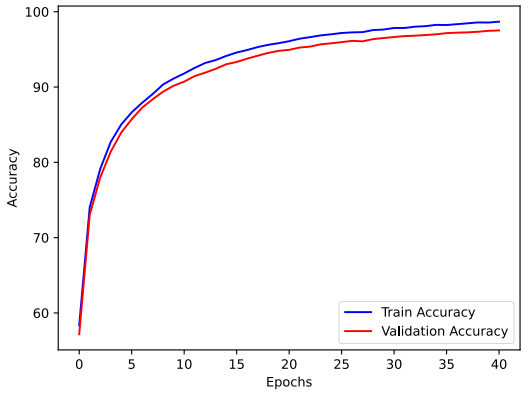
Tiempo de entrenamiento total: 2.0 horas, 51.0 minutos, 40.70 segundos

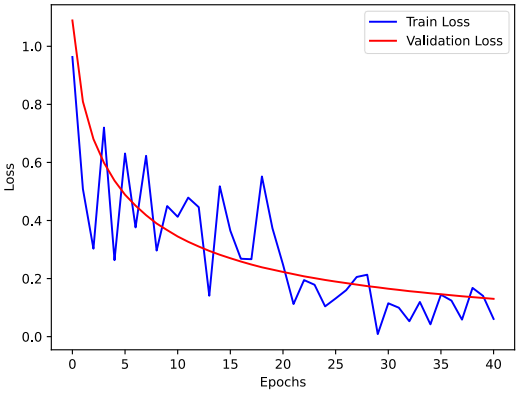
Marca de tiempo para la mejor precisión de validación: 2.0 horas, 46.0 minutos, 14.09 segundos

Máxima precisión relativa de entrenamiento: 98.68 %

Máxima precisión relativa de entrenamiento de prueba: 97.85.4 %

Máxima precisión relativa de entrenamiento de validación: 97.51 %





Prueba del MaxVit.

Conjunto de datos: Pulmón

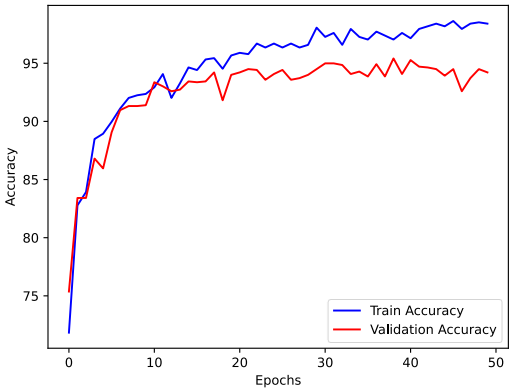
Tiempo de entrenamiento total: 19.0 minutos, 50 segundos

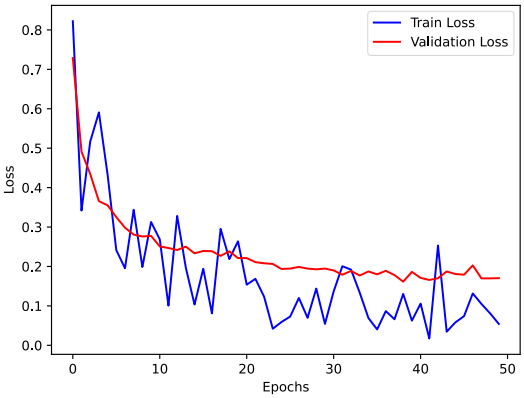
Marca de tiempo para la mejor precisión de validación: 12.0 minutos, 55.16 segundos

Máxima precisión relativa de entrenamiento: 97.73 %

Máxima precisión relativa de entrenamiento de prueba: 96.47 %

Máxima precisión relativa de entrenamiento de validación: 96.70 %





Conjunto de datos: Cerebro

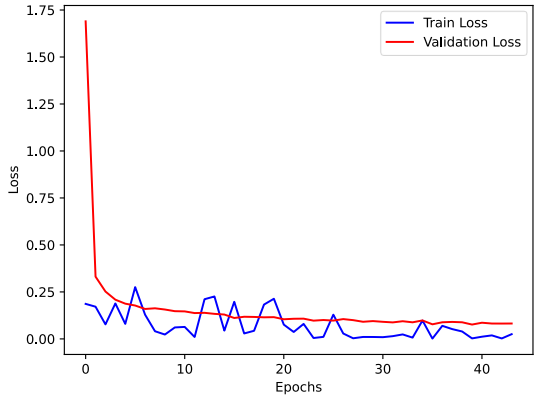
Tiempo de entrenamiento total: 3.0 horas, 40.0 minutos, 6.36 segundos

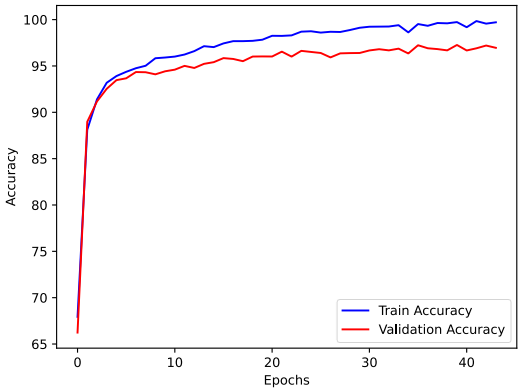
Marca de tiempo para la mejor precisión de validación: 3.0 horas, 18.0 minutos, 25.21 segundos

Máxima precisión relativa de entrenamiento: 99.84 %

Máxima precisión relativa de entrenamiento de prueba: 96.93 %

Máxima precisión relativa de entrenamiento de validación: 97.26 %





Conjunto de datos: Riñón

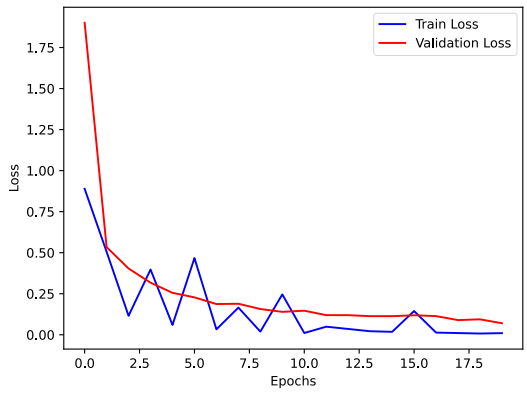
Tiempo de entrenamiento total: 1.0 h, 41.0 m, 18.69 segundos

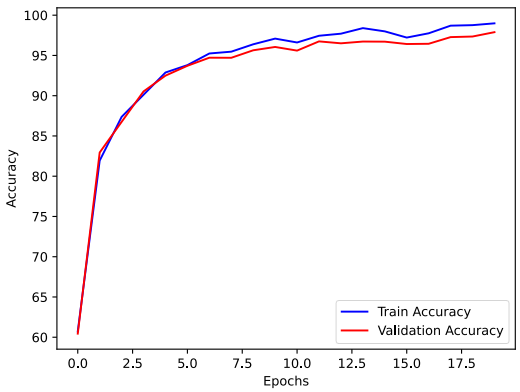
Marca de tiempo para la mejor precisión de validación: 1.0 h, 39.0 m, 54.43 segundos

Máxima precisión relativa de entrenamiento: 98.98 %

Máxima precisión relativa de entrenamiento de prueba: 98.07 %

Máxima precisión relativa de entrenamiento de validación: 97.89 %





Prueba del Swin Transformer.

Conjunto de datos: Pulmón

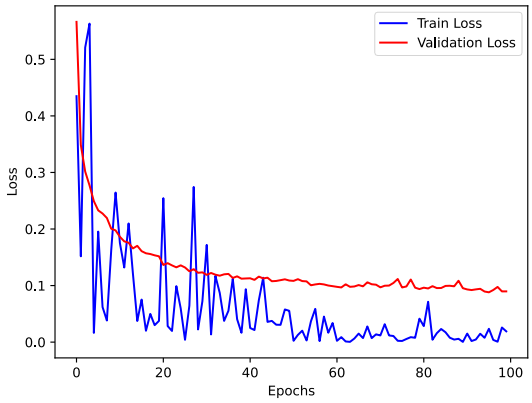
Tiempo de entrenamiento total: 36.0 minutos, 16.65 segundos

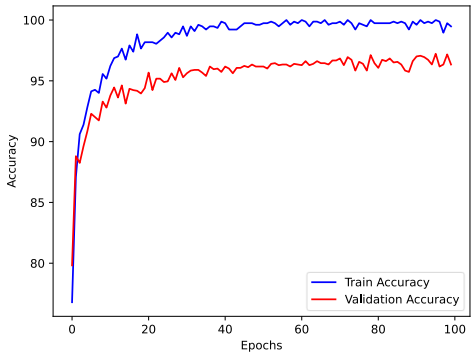
Marca de tiempo para la mejor precisión de validación: 22.0 minutos, 21.01 segundos

Máxima precisión relativa de entrenamiento: 97.73 %

Máxima precisión relativa de entrenamiento de prueba: 96.47 %

Máxima precisión relativa de entrenamiento de validación: 96.70 %





Conjunto de datos: Cerebro

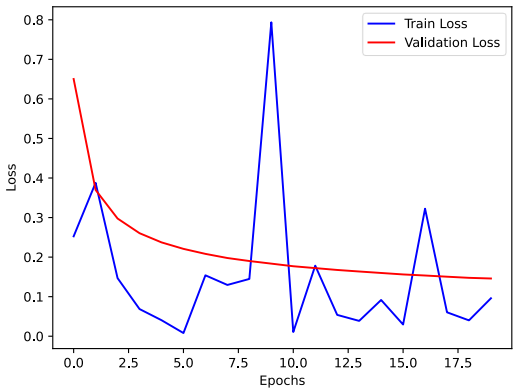
Tiempo de entrenamiento total: 1.0 horas, 30.0 minutos, 11.22 segundos

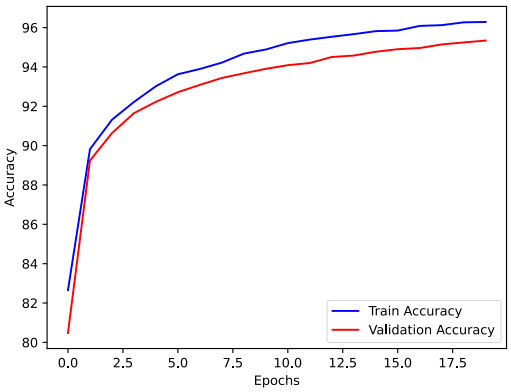
Marca de tiempo para la mejor precisión de validación: 1.0 horas, 28.0 minutos, 20.58 segundos

Máxima precisión relativa de entrenamiento: 96.28 %

Máxima precisión relativa de entrenamiento de prueba: 95.2 %

Máxima precisión relativa de entrenamiento de validación: 95.34 %





Conjunto de datos: Riñón

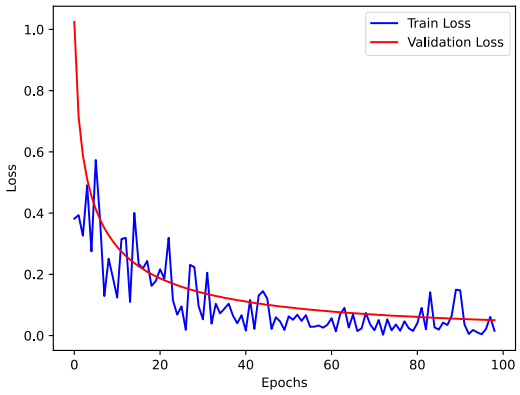
Tiempo de entrenamiento total: 7.0 horas, 6.0 minutos, 45.37 segundos

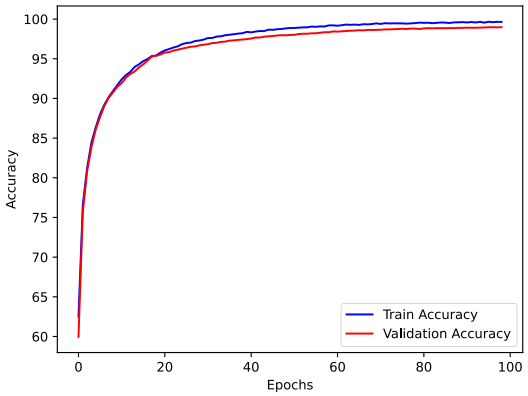
Marca de tiempo para la mejor precisión de validación: 6.0 horas, 31.0 minutos, 12.79 segundos

Máxima precisión relativa de entrenamiento: 97.73 %

Máxima precisión relativa de entrenamiento de prueba: 96.47 %

Máxima precisión relativa de entrenamiento de validación: 96.70 %





Anexo II. Competencias específicas

A continuación, se procede a citar algunas de las Competencias específicas comunes a la rama de informática. del plan de carrera de Ingeniería Informática Plan 41, de la tabla 3.5 en la siguiente fuente [82]:

CI2: "*Capacidad para planificar, concebir, desplegar y dirigir proyectos, servicios y sistemas informáticos en todos los ámbitos, liderando su puesta en marcha y su mejora continua y valorando su impacto económico y social.*". Durante el trabajo se ha tenido en cuenta aspectos como la planificación del proyecto, la puesta en marcha y evaluación continua de los modelos, así como continuas mejoras en funcionalidades necesarias para el desarrollo del *software*, y considerando el impacto en la sociedad a nivel sanitario.

CI6: "*Conocimiento y aplicación de los procedimientos algorítmicos básicos de las tecnologías informáticas para diseñar soluciones a problemas, analizando la idoneidad y complejidad de los algoritmos propuestos.*". Se ha debido estudiar y analizar distintos algoritmos del modelo Transformer y sus variantes, para aplicarlo a la tarea de detección de patrones en imágenes médicas.

CI15 "*Conocimiento y aplicación de los principios fundamentales y técnicas básicas de los sistemas inteligentes y su aplicación práctica.*". Se ha debido construir un modelo de Software capaz de aplicar y ejecutar distintos modelos del modelo Transformer y sus variantes. Además, también ha debido de ajustar los parámetros de los modelos de forma práctica en base de la experiencia para maximizar sus resultados con el propósito de cumplir sus objetivos.

CI16: "*Conocimiento y aplicación de los principios, metodologías y ciclos de vida de la ingeniería del software.*". Se ha seleccionado una metodología CRISP-DM como marco de trabajo para poder poner en marcha el proyecto de investigación y la mejora continua del software en base a las evaluaciones de los modelos.

Anexo III. Definiciones básicas

Se procede a explicar muy brevemente algunos términos presentes en el algoritmo de aprendizaje:

Hiperparámetros se refiere a los parámetros de un modelo neuronal que se ajusta manualmente con el objetivo de controlar y revisar el proceso de aprendizaje a través del entrenamiento y resultado, iterativamente

Paso (*step*) se refiere a una actualización de pesos en toda la red neuronal, implica por tanto calcular el vector gradiente de la función de error con respecto a los parámetros de la red.

Época (*epoch*) se refiere una pasada completa de los datos de entrada y hasta su actualización de pesos. En una época, se realiza múltiples números de pasos (*steps*).

Tamaño de lote o mini-lote (*batch size*) se refiere al número de tokens de entrada en un paso que se realiza en cada época. Por ejemplo, para un tamaño total en un conjunto de datos de 1000 token, se puede configurar un tamaño de lote de 100 tokens, y por tanto se realizaría 10 pasos en cada época.

Paso hacia adelante (*step forward*) se refiere a la entrada de datos a la red neuronal para producir una salida predictiva

Retropropagación (*backpropagation*) se refiere al cálculo de las derivadas parciales de la función de error al final del modelo, para posteriormente ajustar los pesos desde el final de la red neuronal y propagar el cálculo de ajuste hasta el inicio.

Tasa de aprendizaje (*learning rate*) se refiere a un hiperparámetro que controla la magnitud de ajuste de los pesos de una red neuronal. Es una variable escalar, un valor alto puede hacer que el entrenamiento sea divergente e inestable, un valor bajo puede ser que el entrenamiento sea excesivamente lento.

Datos de entrenamiento se refiere al conjunto de datos que se utiliza para ajustar los pesos durante el entrenamiento.

Datos de validación se refiere al conjunto de datos que se utiliza para evaluar su rendimiento durante el entrenamiento. En este proyecto no se ha utilizado datos de validación durante los *steps* para aligerar el tiempo de cómputo, pero en su defecto se dará mayor énfasis a los datos de test de error.

Datos de prueba (*test*) o datos de error se refiere al conjunto de datos que no se ha utilizado durante el entrenamiento, por tanto, es una medida de la capacidad de aprendizaje real pues el modelo deberá ser capaz de predecir los resultados de datos no vistos. Se ha aplicado al final de cada época para poder generar gráficos visuales y observar su curva de aprendizaje, así como su máximo local.

Se procede, por último, lugar, a explicar muy brevemente un componente fundamental del aprendizaje profundo, el Perceptrón Multicapa:

El perceptrón multicapa (*MLP - Multilayer Perceptron*) se desarrolló a partir del perceptrón simple propuesto en la década de 1950. Se trata de una red neuronal formada por una capa de entrada (*input layer*), una o múltiples capas ocultas secuenciales contactadas total o parcialmente (*hidden layers*), y una capa de salida (*output layer*), tal y como se muestra a la derecha de la Figura III-1. Son capaces de resolver problemas no lineales. Cada neurona de la red es la sumatoria de todos los pesos de sus entradas, se le añade un sesgo y al resultado se le aplica una función de activación comúnmente siendo una función no lineal. Los parámetros de red como los pesos (*weights*) y el sesgo neuronal (*bias*) son los que deberán ser entrenados mediante el entrenamiento del modelo, visibles a la izquierda de la Figura III-2. Estos parámetros pueden ser inicializados por un modelo ya entrenado, entonces se estaría aplicando una técnica conocida como *Transfer Learning* o sin carga de parámetros donde se puede inicializar los parámetros aleatoriamente o con valor predeterminado "cero".

A partir de aquí, nacerán otras arquitecturas derivadas de estos conceptos como son las redes convolucionales o las redes residuales. Históricamente, como se ha explicado en el capítulo 2.1 "Avance tecnológico del aprendizaje profundo", la inteligencia artificial nace en la década de 1950 con el propósito que las neuronas artificiales replicaran el comportamiento de las neuronas biológicas, de tal manera que una neurona simple artificial tiene una estructura como es la neurona biológica. En respecto al enfoque arquitectónico, los modelos de inteligencia artificial también se replica los rasgos cognitivos de tal manera que proponen imitar las conexiones con neuronas tanto en capas contiguas siendo un ejemplo en las redes convolucionales como en las conexiones no contiguas, saltándose las conexiones intermedias, siendo un ejemplo en modelos propuestos como en las redes residuales [83].

En este proyecto, el perceptrón multicapa se verá incluido como parte o internamente conectado dentro de las capas de los modelos empleados en el aprendizaje automático. Debido a esta implementación, y a su vez, la posibilidad de derivar otros modelos y complementar esta arquitectura en otras como son las redes convolucionales o las redes residuales, el perceptrón multicapa tendrá un papel relevante en los transformadores de visión.

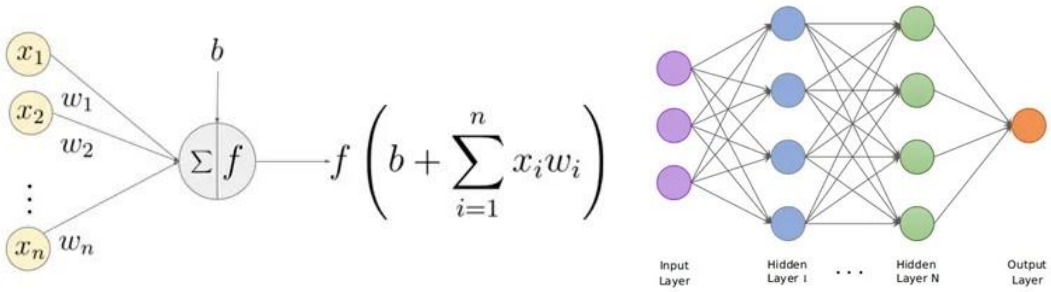


Figura III-3: Perceptrón Multicapa. Fuente: J. Durán [84].

Referencias

[1] A. N. y. J. Marcano, «Las mamografías apoyadas con inteligencia artificial detectan 20% más el cáncer de mama,» *El Mercurio,* 3 Agosto 2023.

[2] *10 Ejemplos de Inteligencia Artificial en Salud,* Big Data Marketer, 2023.

[3] Alexandra, «Diferencias entre TAC y Resonancia,» *Salud Blogs Mafre,* 2022.

[4] «Conceptos básicos de ayuda de CRISP-DM,» IBM, 17 Agosto 2021. [En línea]. Available: https://www.ibm.com/docs/es/spss-modeler/saas?topic=dm-crisp-help-overview.

[5] «Cáncer,» *Organización Mundial de la Salud,* 2 de febrero 2022.

[6] K. Fukushima, «Cognitron: A self-organizing multilayered neural network,» de *Biol. Cybernetics*, 1975.

[7] F. Ramírez, «Historia de la IA: Frank Rosenblatt y el Mark I Perceptrón, el primer ordenador fabricado específicamente para crear redes neuronales en 1957,» *Telefónica Tech,* 2018.

[8] «El origen de Deep Learning,» Universidad de Alcalá, 2022. [En línea]. Available: https://master-deeplearning.com/origen-deep-learning/.

[9] B. House, «2012: A Breakthrough Year for Deep Learning,» Deep Sparse, 17 Julio 2019. [En línea]. Available: https://medium.com/neuralmagic/2012-a-breakthrough-year-for-deep-learning-2a31a6796e73.

[10] Na8, «Breve Historia de las Redes Neuronales Artificiales,» Aprende Machine Learning, 12 Septiembre 2018. [En línea]. Available: https://www.aprendemachinelearning.com/breve-historia-de-las-redes-neuronales-artificiales/.

[11] A. Vaswani, S. Bengio, E. Brevdo, F. Chollet, A. Gomez, S. Gouws y D. Zhong, «Attention is All You Need,» de *31st Conference on Neural Information Processing Systems (NIPS 2017)*, Long Beach, CA, USA, 2017.

[12] A. Jacinto, «HISTORIA DEL APRENDIZAJE AUTOMÁTICO: LA LÍNEA DE TIEMPO COMPLETA,» *StartechUP,* 2022.

[13] R. Merritt, «¿Qué es un Modelo Transformer?,» NVidia, 2022. [En línea]. Available: https://la.blogs.nvidia.com/2022/04/19/que-es-un-modelo-transformer/.

[14] E. Sandu, «Inteligencia artificial en la conducción autónoma,» Metaverso.pro, [En línea]. Available: https://metaverso.pro/blog/inteligencia-artificial-en-la-conduccion-

autonoma/#:~:text=%C2%BFQu%C3%A9%20es%20la%20conducci%C3%B3n%20aut%C3%B3noma,segundo%20y%20conducen%20con%20seguridad..

[15] «El Campus de Tafira acoge la prueba piloto de la primera guagua autónoma de Canarias,» Universidad de Las Palmas de Gran Canaria, 31 Marzo 2023. [En línea]. Available: https://www.ulpgc.es/noticia/2023/03/31/campus-tafira-acoge-prueba-piloto-primera-guagua-autonoma-canarias.

[16] S. M. Weiss, C. A. Kulikowski, S. Amarel y A. Safir, «A model-based method for computer-aided medical decision-making,» *Artificial Intelligence,* vol. 11, pp. 145-172, 1978.

[17] J. M. Juarez., «Tres cosas que saber sobre la Inteligencia Artificial en Medicina,» Blog de IA, Computación y Medicina, Marzo 2018. [En línea]. Available: https://webs.um.es/jmjuarez/1-queesinteligenciaartificialmedicina/.

[18] *Inteligencia artificial en el campo de la salud,* Wikipedia, La enciclopedia libre, 2023.

[19] C. D. Cecco, «¿Cómo puede la inteligencia artificial mejorar la salud de los latinoamericanos?,» 2 de Septiembre 2021. [En línea].

[20] Redacción APD, «Efectos de la inteligencia artificial en la medicina y sus aplicaciones más novedosas,» *APD,* 23 Febrero 2021.

[21] J. M. Medina y J. S. Pérez, «High accuracy brain tumor classification with EfficientNet and magnetic resonance images,» de *n Advances in Signal Processing and Artificial Intelligence (ASPAI' 2023)*, Tenerife, 2023.

[22] D. Reyes y J. Sánchez, «Performance of convolutional neural networks for the classification of brain tumors using magnetic resonance imaging,» *Heliyon,* vol. 10, nº 3, 2 Febrero 2024.

[23] A. Khan, Z. Rauf, A. R. Khan, S. Rathore, S. H. Khan, N. S. Shah, U. Farooq, H. Asif, A. Asif, U. Zahoora, R. U. Khalil, S. Qamar, U. H. Asif, F. B. Khan, A. Majid y Gwak, «A Recent Survey of Vision Transformers for Medical Image,» arXiv, 2023. [En línea]. Available: https://arxiv.org/abs/2312.00634.

[24] J. B. Rollins, *Metodología Fundamental para la Ciencia de Datos,* IBM Corporation, 2015.

[25] J. C. R. K. T. K. T. R. C. S. R. W. Pete Chapman, «CRISP-DM 1.0,» Agosto 2020. [En línea]. Available: https://www.kde.cs.uni-kassel.de/wp-content/uploads/lehre/ws2012-13/kdd/files/CRISPWP-0800.pdf.

[26] P. Haya, «La metodología CRISP-DM en ciencia de datos,» *Instituto de Ingeniería del Conocimiento.*

[27] S. D. Quintero, *Jupyter,* Aprende Python.

[28] *PyTorch,* Wikipedia, La enciclopedia libre, 2024.

[29] C. d. Wikipedia, *Matplotlib,* Wikipedia, La enciclopedia libre, 2023.

[30] c. d. Wikipedia, «Unidad de procesamiento gráfico,» Wikipedia, La enciclopedia libre, 22 octubre 2023. [En línea]. Available: https://es.wikipedia.org/w/index.php?title=Unidad_de_procesamiento_gr%C3%A1fico&oldid=154789486.

[31] c. d. Wikipedia, «Arquitectura de Von Neumann,» Wikipedia, 7 noviembre 2023. [En línea]. Available: https://es.wikipedia.org/w/index.php?title=Arquitectura_de_Von_Neumann&oldid=155131967.

[32] «CUDA Zone,» NVIDIA Corporation, 2024. [En línea]. Available: https://developer.nvidia.com/cuda-zone.

[33] *NVIDIA cuDNN,* NVIDIA Corporation, 2024.

[34] «Diagnóstico por imágenes,» 21 Enero 2021. [En línea]. Available: https://medlineplus.gov/spanish/diagnosticimaging.html.

[35] «Exploraciones con tomografía computarizada (TC) para el cáncer,» *Instituto Nacional del Cáncer,* 2019.

[36] «Imagen por Resonancia Magnética (IRM),» National Institute of Biomedical Imaging and Bioengineering (NIBIB), [En línea]. Available: https://www.nibib.nih.gov/espanol/temas-cientificos/imagen-por-resonancia-magn%C3%A9tica-irm.

[37] *How Does an MRI Scan Work?,* NIBIB gov, 2013.

[38] «Cáncer de riñón: Estadísticas,» *American Society of Clinical Oncology ,* 2023.

[39] «Las piedras en los riñones,» Instituto Nacional de la Diabetes y las Enfermedades Digestivas y Renales, [En línea]. Available: https://www.niddk.nih.gov/health-information/informacion-de-la-salud/enfermedades-urologicas/piedras-rinones.

[40] *Información sobre los quistes renales (simples y complejos),* Gopa Cirujanos; S.L, 2023.

[41] «Cáncer de riñón,» *Clínica Universidad de Navarra,* 2023.

[42] J. G. PÉREZ DE LARRAYA, «Tumor cerebral,» Universidad de Navarra . [En línea].

[43] «Afecciones que tratamos: Gliomas, astrocitomas y glioblastomas,» Johns Hopkins Medicine International, [En línea]. Available: https://www.hopkinsmedicine.org/international/espanol/conditions-treatments/neurosurgery/gliomas.

[44] «Glioma,» Mayo Clinic, 7 Marzo 2024. [En línea]. Available: https://www.mayoclinic.org/es/diseases-conditions/glioma/symptoms-causes/syc-20350251#:~:text=El%20glioma%20es%20una%20multiplicaci%C3%B3n,y%20las%20ayudan%20a%20funcionar..

[45] «Meningioma,» Instituto Nacional del Cáncer, 26 Marzo 2020. [En línea]. Available: https://www.cancer.gov/rare-brain-spine-

tumor/espanol/tumores/meningioma#:~:text=El%20meningioma%20es%20un%20tumor,grado%20son%20muy%20poco%20comunes..

[46] J. E. Nora, J. A. D. Lares-Franco y M. M. Portra, «¿Qué es el tumor de la hipófisis?,» Portal Clínic, 4 septiembre 2020. [En línea].

[47] DrTango, Inc, «Cáncer pulmonar,» MedlinePlus en español [Internet]. Bethesda (MD): Biblioteca Nacional de Medicina de EE.UU, 2 Febrero 2024. [En línea].

[48] R. PÉREZ y MARÍA, «Cáncer de pulmón,» Universidad de Navarra. [En línea].

[49] C. d. Wikipedia, *Kaggle,* Wikipedia, La enciclopedia libre, 2022.

[50] M. N. Islam, M. Hasan, M. Hossain, M. Alam, G. Rabiul, M. Z. Uddin y A. Soylu, «CT Kidney Dataset: Normal-Cyst-Tumor and Stone,» Kaggle, 2022. [En línea]. Available: https://www.kaggle.com/datasets/nazmul0087/ct-kidney-dataset-normal-cyst-tumor-and-stone.

[51] *About DICOM: Overview,* The Medical Imaging Technology Association (MITA).

[52] O. S. Naren, «Multi Cancer Dataset,» 2023. [En línea]. Available: https://www.kaggle.com/datasets/obulisainaren/multi-cancer.

[53] «Glioma,» Mayo Clinic, 2023. [En línea]. Available: https://www.mayoclinic.org/es/diseases-conditions/glioma/symptoms-causes/syc-20350251.

[54] A. Cohen-Gadol, «Perspectiva de los pacientes con meningioma,» 2023. [En línea]. Available: https://www.aaroncohen-gadol.com/es/pacientes/meningioma-es/supervivencia/vivir-con-meningioma/#:~:text=Un%20meningioma%20es%20una%20afecci%C3%B3n,tienden%20a%20crecer%20muy%20lentamente..

[55] «Pituitary Adenomas,» Cleveland Clinic, 2022. [En línea]. Available: https://my.clevelandclinic.org/health/diseases/15328-pituitary-adenomas.

[56] H. F. Al-Yasriy, «The IQ-OTH/NCCD lung cancer dataset,» 2019. [En línea]. Available: https://doi.org/10.34740/KAGGLE/DS/672399.

[57] H. F. Al-Yasriy, M. S. Al-Husieny, F. Y. Mohsen, E. A. Khalil y Z. S. Hassan, «Diagnosis of Lung Cancer Based on CT Scans Using CNN,» de *Materials Science and Engineering*, 2020.

[58] H. F. Kareem, M. S. A.-Husieny, F. Y. Mohsen, E. A. Khalil y H. Z. S., «Evaluation of SVM performance in the detection,» *Indonesian Journal of Electrical Engineering and Computer Science,* vol. 21, nº 3, pp. 1731-1738, 2021.

[59] J. M. C. Ramírez, «ViT - Vision Transformer,» Linkedin, 2023.

[60] Arvind, «Attention Mechanism in Vision Models,» *Medium,* 2021.

[61] G. Santini, «AI in Medical Imaging: The Kidney Tumor Segmentation Challenge,» Keosys, 2019.

[62] «AlexNet,» *Wikipedia,* 27 January 2024.

[63] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, V. Usadel y N. Houlsby, «An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale,» de *Conference on Computer Vision and Pattern Recognition*, 2021.

[64] C. d. Wikipedia, «Transformador de visión,» Wikipedia, 22 de julio 2023. [En línea]. Available: https://es.wikipedia.org/wiki/Transformador_de_visi%C3%B3n.

[65] «Vision Transformers (ViT) Explained,» Pinecone Systems, San Francisco.

[66] H. hettiarachchi, «Unveiling Vision Transformers: Revolutionizing Computer Vision Beyond Convolution,» Medium, 12 Agosto 2023. [En línea]. Available: https://medium.com/@hansahettiarachchi/unveiling-vision-transformers-revolutionizing-computer-vision-beyond-convolution-c410110ef061.

[67] D. Shah, «Vision Transformer: What It Is & How It Works,» V7labs, 2022. [En línea]. Available: https://www.v7labs.com/blog/vision-transformer-guide.

[68] *Perceptrón multicapa,* Wikipedia, La enciclopedia libre, 2023.

[69] J. M. C. Ramírez, «ViT - Vision Transformer,» Linkedin, 2023.

[70] PyTorch, «Models and pre-trained weights,» Linux Foundation, 2017. [En línea]. Available: https://pytorch.org/vision/stable/models.html.

[71] S.-H. Tsang, «Review: Vision Transformer (ViT),» Medium, 2022.

[72] Z. Liu, Y. Lin, Y. Cao, H. Hu y Y. Wei, «Swin Transformer: Hierarchical Vision Transformer using Shifted Windows,» de *IEEE/CVF International Conference on Computer Vision (ICCV)*, Montreal, QC, Canada, 2021.

[73] Z. Liu, H. Hu, Y. Lin, Z. Yao, Z. Xie, Y. Wei, J. Ning, Y. Cao, Z. Zhang y L. Dong, *Swin Transformer V2: Scaling Up Capacity and Resolution,* University of Science and Technology of China; Xian Jiaotong University; Tsinghua University; Huazhong University of Science and Technology.

[74] *Segmentación semántica: Tres cosas que es necesario saber,* MATLAB .

[75] Rishigami, *Swin Transformer (Tensorflow),* Github, 2021.

[76] Z. Tu, H. Talebi, H. Zhang, F. Yang, P. Milanfar, A. Bovik y Y. Li, «MaxViT: Multi-Axis Vision Transformer,» de *European Conference on Computer Vision*, 2022.

[77] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov y L.-C. Chen, *MobileNetV2: Inverted Residuals and Linear Bottlenecks,* Google Inc., 2019.

[78] «Adam,» Cornell University, Noviembre 2020. [En línea]. Available: https://optimization.cbe.cornell.edu/index.php?title=Adam.

[79] «CrossEntropyLoss getting value > 1,» Pytorch Forums, Septiembre 2023. [En línea]. Available: https://discuss.pytorch.org/t/crossentropyloss-getting-value-1/188115.

[80] T. Contributors, «Alexnet,» Pytorch 1.11.0 documentation, 2022. [En línea]. Available: https://pytorch.org/vision/stable/models/generated/torchvision.models.alexnet.html#torchvision.models.AlexNet_Weights.

[81] A. Lee, «¿Qué Es un Modelo de IA Previamente Entrenado?,» Nvidia, 11 Enero 2023. [En línea]. Available: https://la.blogs.nvidia.com/blog/que-es-un-modelo-de-ia-previamente-entrenado/.

[82] «Grado en Ingeniería Informática,» Escuela de Ingeniería Informática, 29 Febrero 2024. [En línea]. Available: https://www.eii.ulpgc.es/sites/default/files/2024-05/20240229%20Grado%20en%20Ingenier%C3%ADa%20Inform%C3%A1tica%20-%20rev03.pdf.

[83] M. Rivera, «La Red Residual (Residual Network, ResNet),» Agosto 2019. [En línea]. Available: http://personal.cimat.mx:8181/~mrivera/cursos/aprendizaje_profundo/resnet/resnet.html.

[84] J. Durán, «Todo lo que Necesitas Saber sobre el Descenso del Gradiente Aplicado a Redes Neuronales,» Medium, 2019. [En línea]. Available: https://medium.com/metadatos/todo-lo-que-necesitas-saber-sobre-el-descenso-del-gradiente-aplicado-a-redes-neuronales-19bdbb706a78.