ULPGC
Universidad de
Las Palmas de
Gran Canaria

ESCUELA DE
INGENIERÍA INFORMÁTICA

Trabajo de Fin de Grado

Análisis de Neuroimágenes para el diagnóstico temprano de la enfermedad del Alzheimer

TITULACIÓN: Grado en Ingeniería Informática

AUTOR: Eric Cabrera Cruz

TUTORIZADO POR:
Pablo Carmelo Fernández López
Carmen Paz Suárez Araujo

Fecha [Julio/2025]

ULPGC Universidad de Las Palmas de Gran Canaria | EII ESCUELA DE INGENIERÍA INFORMÁTICA

TFT04

SOLICITUD DE DEFENSA DE TRABAJO DE FIN DE TÍTULO

D. Eric Cabrera Cruz, autor del trabajo Análisis de Neuroimágenes para el diagnóstico temprano de la enfermedad del Alzheimer, correspondiente a la titulación Grado de Ingeniería Informática.

SOLICITA

que se inicie el procedimiento de defensa del mismo, para lo que se adjunta la documentación requerida, haciendo constar que

[X] se autoriza / [ ] no se autoriza la grabación en audio de la exposición y turno de preguntas.

Asimismo, con respecto al registro de la propiedad intelectual/industrial del TFT, declara que:

[ ] Se ha iniciado o hay intención de iniciarlo (defensa no pública).

[X] No está previsto.

Y para que así conste firma la presente. (fecha en firma electrónica)

El/La estudiante

Fdo.————————

A rellenar y firmar **obligatoriamente** por el/la/los/las tutores

En relación con la presente solicitud, se informa:(firmar donde corresponda)

Positivamente (en caso de detección de copia, esta firma quedará invalidada)	Negativamente (justificación en TFT05)
Fdo.————	Fdo.————

DIRECTOR DE LA ESCUELA DE INGENIERÍA INFORMÁTICA

Agradecimientos

Quería agradecer a mis tutores Pablo Carmelo Fernández López y Carmen Paz Suárez Araujo por acompañarme a lo largo de este proyecto, así como al investigador Ylermi Cabrera León por sus innumerables consejos durante el desarrollo del mismo.

Por otro lado, también a mis familiares y amigos por todo su apoyo y comprensión. Y especialmente a mi fallecida perrita Luna que indiferentemente de dónde esté, me da fuerzas para continuar adelante.

Resumen

La demencia cortical, enfermedad de Alzheimer (EA) es uno de los mayores desafíos de salud actuales. Este TFT desarrolla, optimiza un sistema inteligente basado en la arquitectura profunda Densenet, de ayuda a la detección temprana de la EA mediante la clasificación de neuroimágenes, como las IRM, imagen de resonancia magnética, así como predicciones de la evolución de la neuropatología. El dataset, procede de la base de datos ADNI. Consiste en las IRM de 819 pacientes, con situaciones cognitivas diferentes: sanos, con deterioro cognitivo leve y con la EA, y en tiempos distintos, t = 0, t = 6 meses, t = 12 meses.

El objetivo final de este estudio es determinar la eficiencia de esta propuesta de aprendizaje profundo, frente a otras soluciones con redes neuronales superficiales usando esencialmente test neuropsicológicos.

Abstract

Cortical dementia, Alzheimer's disease (AD), is one of the major current health challenges. This final degree project develops and optimizes an intelligent system based on the deep architecture DenseNet to support the early detection of AD through the classification of neuroimages, such as MRIs (Magnetic Resonance Imaging), as well as predictions of the progression of neuropathology. The dataset comes from the ADNI (Alzheimer's Disease Neuroimaging Initiative) database. It consists of MRI scans of 819 subjects, with different cognitive conditions: healthy, mild cognitive impairment, and Alzheimer's disease, at different time points: t = 0, t = 06 months, and t =12 months.

The ultimate goal of this study is to determine the efficiency of this deep learning approach compared to other solutions using shallow neural networks that primarily rely on neuropsychological tests.

Índice general

Índice de figuras

Índice de Tablas

Capítulo 1

Introducción

> Una inversión en conocimiento paga el mejor interés
>
> Benjamin Franklin.

Según la Real Academia Española (RAE) [1], la demencia se define como un "estado caracterizado por el deterioro progresivo e irreversible de las facultades mentales". La forma más común de demencia, especialmente en personas mayores, es la demencia cortical, cuya principal manifestación es la enfermedad de Alzheimer (AD, por sus siglas en inglés).

1.1. Preámbulo

Con el fin de agilizar la comprensión del Trabajo de Fin de Grado desarrollado, en este apartado se presentan los fundamentos teóricos y técnicos necesarios para contextualizar dicho proyecto. De este modo, se abordarán conceptos clínicos claves relacionados con el Alzheimer y el deterioro cognitivo leve (MCI, por sus siglas en inglés), así como una descripción de la base de datos ADNI (Alzheimer's Disease Neuroimaging Initiative), que constituye la fuente principal de datos utilizada. Además, se introducirán los conceptos básicos de las redes neuronales convolucionales (CNN), modelo computacional central en el enfoque propuesto.

1.1.1. La enfermedad del Alzheimer

La enfermedad de Alzheimer (AD), denominada así en honor al psiquiatra alemán Alois Alzheimer, constituye la forma más frecuente de demencia. Se trata de una patología neurodegenerativa de progresión lenta, cuya principal característica es la presencia de placas neuríticas y ovillos neurofibrilares. Estas alteraciones son consecuencia de la acumulación del

péptido beta-amiloide (Aβ), especialmente en las regiones más afectadas del cerebro, como el lóbulo temporal medial y las estructuras neocorticales [2] Este proceso de deterioro se puede ver esquematizado en la figura 1.1.

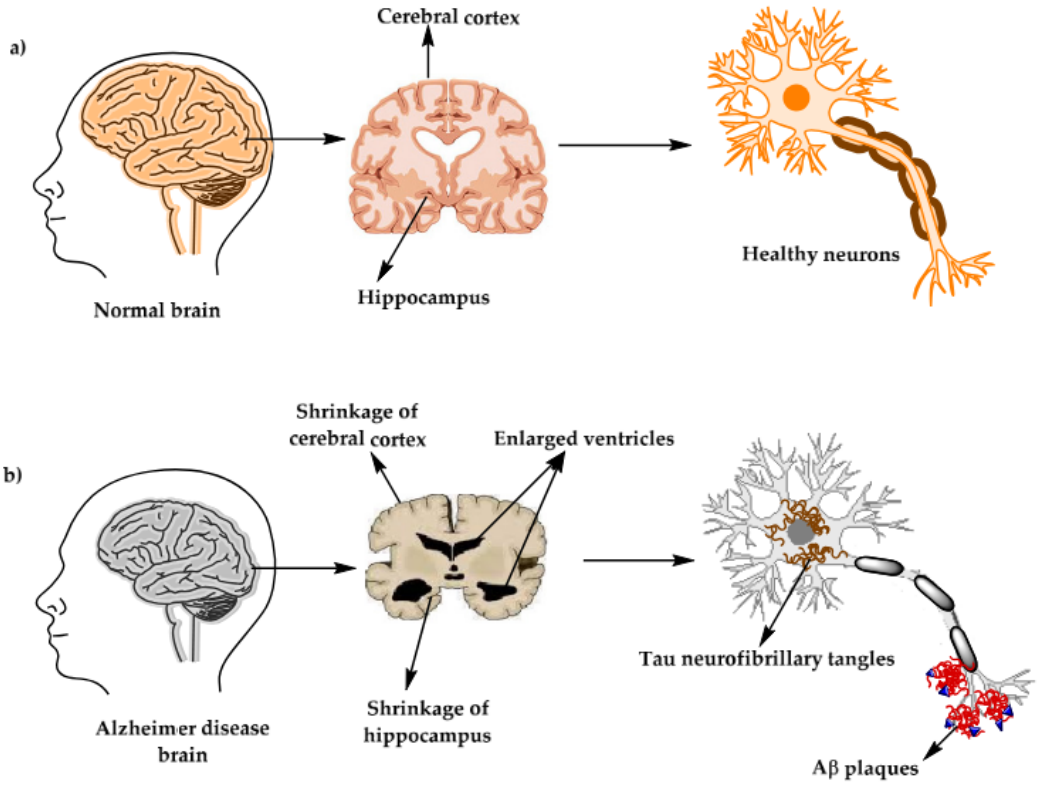


Figura 1.1: La estructura fisiológica del cerebro y las neuronas en, (a) un cerebro sano y, (b) un cerebro con enfermedad de Alzheimer. [2]

La AD ha emergido como la forma más prevalente de falla mental en humanos debido al radical aumento en la esperanza de vida, constituyendo así la causa más común de demencia [3]. Se estima que para el año 2050 alrededor de 131 millones de personas en todo el mundo padecerán esta enfermedad [4], lo que la posiciona como uno de los mayores desafíos de salud pública a nivel global.

A pesar de su creciente prevalencia, las causas fundamentales y los mecanismos patológicos del Alzheimer aún no han sido completamente comprendidos . En la actualidad, su diagnóstico se basa principalmente en pruebas neurocognitivas y en la evaluación del historial clínico del paciente. No obstante, también se utilizan imágenes por resonancia magnética (MRI, por sus siglas en inglés) para analizar patrones morfométricos cerebrales, con el fin de identificar biomarcadores característicos de la enfermedad, como alteraciones en regiones específicas del cerebro, entre ellas, por ejemplo el hipocampo [5].

En los últimos años, se han desarrollado diversos métodos que emplean los datos obtenidos

mediante MRI para detectar tanto la enfermedad de Alzheimer (AD) como el deterioro cognitivo leve (MCI). Este último se caracteriza por una disminución leve pero medible de las capacidades [6], situándose en un punto intermedio entre el envejecimiento cognitivo normal y el Alzheimer . Se estima que más del 33 % de las personas con MCI progresarán hacia la enfermedad del Alzheimer en un período de cinco años o más [7] y [8].

Aunque la enfermedad de Alzheimer no tiene cura, existen tratamientos que pueden retrasar su aparición y ralentizar el deterioro cognitivo, especialmente si se inician en las etapas tempranas del trastorno [9] como se observa de forma esquematizada en la figura 1.2. Por esta razón, el diagnóstico anticipado tanto del MCI como del Alzheimer es fundamental para influir en el avance de la enfermedad, ralentizando este último, y mejorando significativamente la calidad de vida de los pacientes.

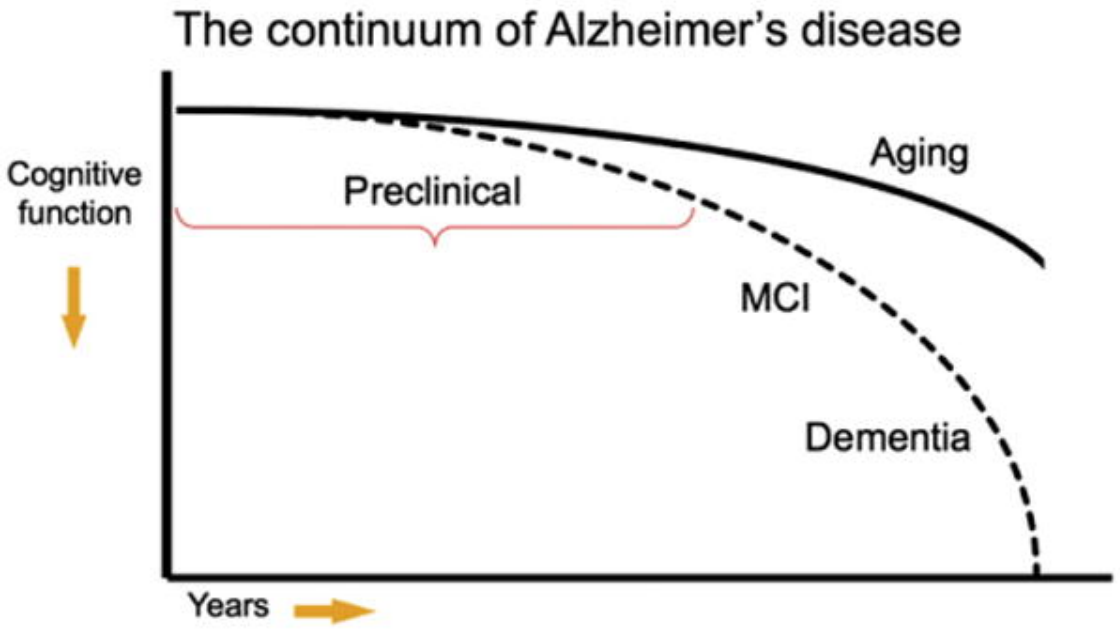


Figura 1.2: Comparación entre el deterioro cognitivo debido al envejecimiento frente al producido por la demencia tipo AD con el paso de los años [10]

Los síntomas de la AD varían según el estado de la enfermedad. El signo más común y temprano suele manifestarse como pérdida de la memoria a corto plazo, frecuentemente acompañado de alteraciones en el lenguaje. Sin embargo, en las etapas iniciales los síntomas pueden ser muy sutiles o incluso inexistentes, lo que dificulta su detección temprana. En algunos casos, la enfermedad puede presentarse de forma asintomática, retrasando aún más el diagnóstico [11].

Actualmente, el diagnóstico anticipado de la enfermedad se realiza a través de exámenes neuropsicológicos [12]. Este proceso depende en gran medida del análisis clínico manual realizado por profesionales de la salud, lo cual puede resultar tedioso y lento, especialmente ante la creciente cantidad de pacientes que presentan síntomas compatibles con esta patología.

Sin embargo, el volumen de datos y el número de pacientes supone un reto creciente para el personal sanitario. Esta situación implica un riesgo significativo, ya que un diagnóstico inadecuado o tardío puede empeorar el pronóstico del paciente a medida que la enfermedad avanza [13].

1.1.2. Base de datos ADNI

ADNI es un estudio que apoya activamente la investigación y el desarrollo de tratamientos que ralenticen o detengan la progresión de la AD, y cuyos objetivos principales son mejorar el diagnóstico clínico de la enfermedad y proporcionar datos a investigadores de todo el mundo. De este modo, investigadores en más de 60 centros clínicos de Estados Unidos y Canadá recopilan datos sobre el envejecimiento normal, el deterioro cognitivo leve y la demencia, con el fin de estudiar la progresión de la enfermedad en el cerebro humano. [14]

El estudio ADNI comenzó en 2004 y ha estado recopilando datos de forma continua a lo largo de las distintas fases del mismo. ADNI4 es la fase actual del estudio (2022-2027). Estas fases las podemos encontrar de forma esquematizada en la tabla 1.1.

Características del Estudio	**ADNI-1**	**ADNI-GO**	**ADNI-2**	**ADNI-3**	**ADNI-4**
Objetivo Principal	Desarrollar biomarcadores como medidas de resultado para ensayos clínicos	Examinar biomarcadores en etapas más tempranas de la enfermedad	Desarrollar biomarcadores como predictores del deterioro cognitivo y como medidas de resultado	Estudiar el uso de PET con tau y técnicas de imágenes funcionales en ensayos clínicos	Mejorar la generalizabilidad de la investigación sobre la enfermedad de Alzheimer
Financiación	$40 millones federales (NIA), $27 millones industria y fundaciones	$24 millones de fondos de la Ley de Recuperación Estadounidense	$40 millones federales (NIA), $27 millones industria y fundaciones	$40 millones federales (NIA), hasta $20 millones industrias y fundaciones	$147 millones federales (NIA)
Duración/fecha de inicio	5 años / octubre 2004	5 años / septiembre 2009	5 años / septiembre 2011	5 años / septiembre 2016	5 años / septiembre 2022
Cohorte	200 personas de edad avanzada, 400 MCI, 200 AD	Existentes ADNI-1 + 200 EMCI	Existentes en ADNI-1 + 200 EMCI	Existentes en ADNI-1, ADNI-GO, ADNI-2 + 133 personas de edad avanzada, 100 EMCI, 150 MCI, 150 AD	Existentes en ADNI-1, ADNI-GO, ADNI-2, ADNI-3 + 120 controles ancianos, 200 MCI, 100 AD/DEM

Tabla 1.1: Características de los Estudios ADNI adaptada de la fuente [14]

El conjunto de datos utilizado, que se compone de imagénes en formato .nii, las cuales corresponden a diferentes estudios realizados sobre 819 pacientes, proviene de la base de datos ADNI1, este conjunto de datos de ADNI es una colección integral y ampliamente utilizada que incluye datos longitudinales clínicos, de neuroimagen, genéticos y de otros

biomarcadores. Abarca diversos tipos de información, como imágenes cerebrales estructurales, funcionales y moleculares, biomarcadores en biofluidos, evaluaciones cognitivas, datos genéticos e información de carácter demográfico sobre los pacientes [14].

Para realizar las experiencias de este proyecto, se han extraído pruebas MRI de 819 pacientes en diferentes estados cognitivos: individuos sanos (CN, por sus siglas en inglés), con deterioro cognitivo leve (MCI) y con Alzheimer, evaluados en distintos momentos de tiempo, y en correspondencia con las diferentes visitas que han realizado dichos pacientes (t = 0, t = 6 meses, t = 12 meses).

1.1.3. Las redes neuronales convolucionales

Las redes neuronales convolucionales conocidas como CNN por sus siglas en inglés, también llamadas "ConvNets", "Redes Neuronales Artificiales Invariantes al Desplazamiento" o "Redes Neuronales Artificiales Invariantes al Espacio", son redes neuronales profundas de tipo feed-forward regularizadas, en las que se realiza una operación de convolución [13].

Se ha utilizado un tipo de red CNN para este proyecto debido a que han demostrado su utilidad para reconocer patrones y anomalías relacionados con la enfermedad de Alzheimer en imágenes de resonancia magnética del cerebro. Las MRI se emplean cada vez más para el diagnóstico clínico de la AD y del MCI por que ayudan a diferenciar las alteraciones neuropatologicas asociadas a estas enfermedades [15].

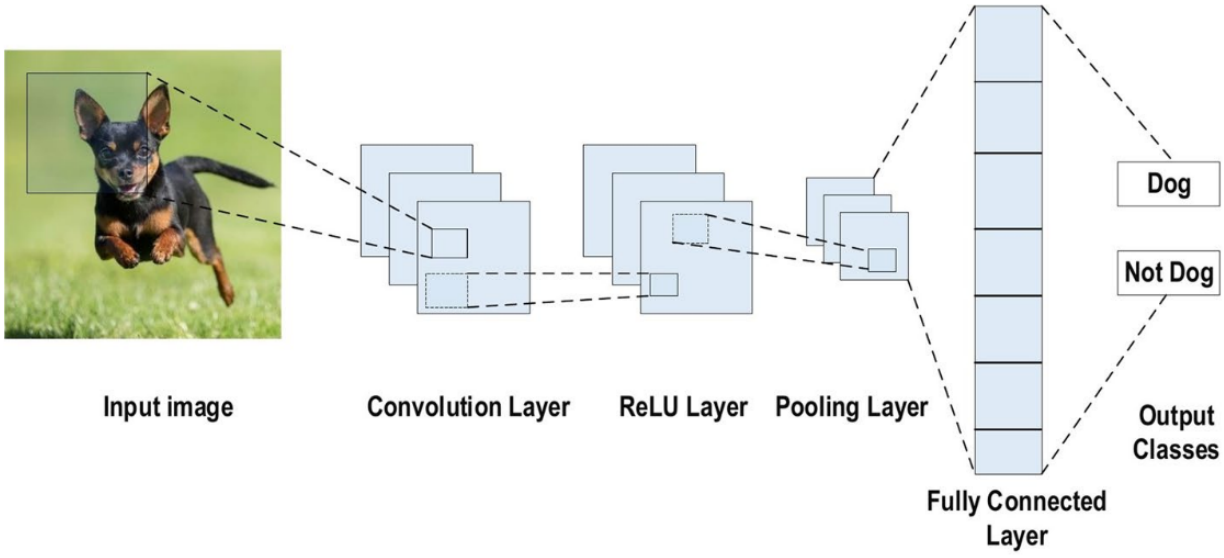


Figura 1.3: Ejemplo básico de CNN en tratamiento de imágenes. Obtenida de la fuente [16]

Las ventajas de emplear una CNN en nuestro proyecto son:

- ✔ Las CNN son bastante robustas frente a traslaciones, rotaciones y escalados de los datos de entrada, una característica muy útil para el reconocimiento de imágenes y videos, así

como para la clasificación de imágenes. Las capas de agrupamiento (pooling) ubicadas después de las capas convolucionales son la principal razón de esta robustez [13]. Esto es esencial debido a la naturaleza de clasificación de imágenes que necesita este proyecto.

- ✔ El aprendizaje simultáneo de las capas de extracción de características y de la capa de clasificación permite, a su vez, que la salida del modelo esté altamente estructurada y dependa en gran medida de las características extraídas [16].

1.1.4. DenseNet

Se ha escogido DenseNet (Dense Convolutional Network) como la arquitectura principal de este proyecto por sus ventajas, su éxito en la clasificación de imágenes naturales y en el diagnóstico del Alzheimer [17]. Asimismo existen estudios donde se demuestra que la arquitectura DenseNet rinde mejor que otras arquitecturas en el problema de clasificación del Alzheimer [11].

DenseNet es una arquitectura de red donde cada capa está conectada directamente con todas las capas posteriores en la red (dentro de su mismo DenseBlock ejemplo en la figura1.4). Para cada capa, los mapas de características de todas las capas anteriores se tratan como entradas separadas, mientras que sus propios mapas de características se pasan como entradas a todas las capas posteriores [18].

Mientras que otras arquitecturas CNN mas tradicionales, de N capas tienen N matrices de conexiones, DenseNet cuenta con $\frac{L(L+1)}{2}$ matrices de conexiones [18].

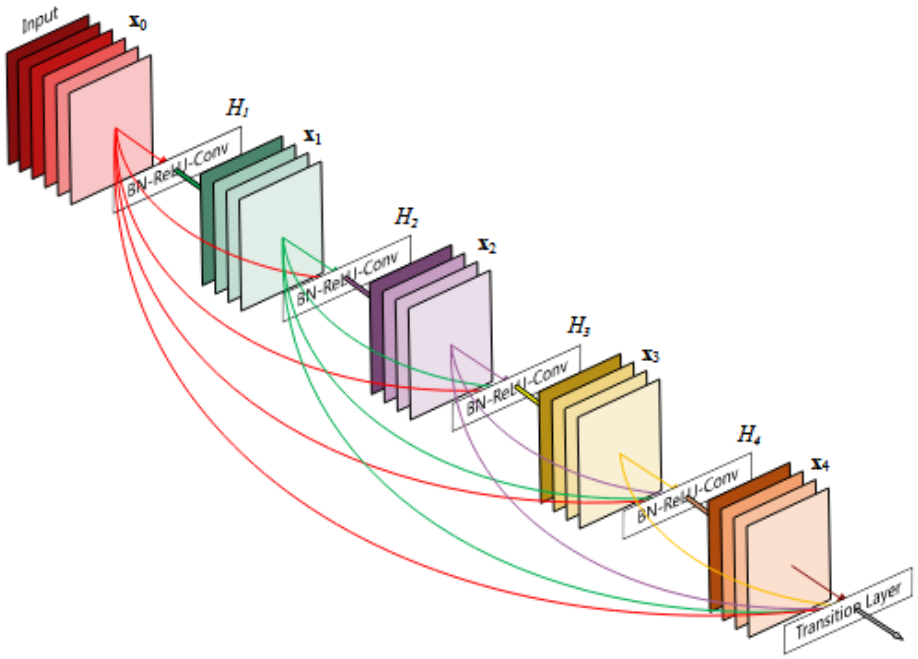


Figura 1.4: Ejemplo de un DenseBlock de 5 capas con una tasa de crecimiento de k=4. Cada capa toma como entrada todos los mapas de características anteriores [18].

Ventajas de la arquitectura DenseNet:

- ✔ Alivia el problema del gradiente desvanecido.
- ✔ Fortalece la propagación de características.
- ✔ Fomenta la reutilización de características y reducen sustancialmente el número de parámetros[18].

1.2. Estado actual y objetivos iniciales

1.2.1. Motivación y antecedentes

La motivación de este proyecto surge de una combinación de factores personales, académicos y sociales. Por un lado, deseaba aplicar mis conocimientos técnicos en un ámbito que trascendiera lo puramente académico, buscando que el resultado de este trabajo pudiera tener un impacto real en la sociedad. En este sentido, me incliné por el desarrollo de sistemas aplicados al ámbito médico, y en particular por mi interés en la creación de modelos de clasificación de datos e imágenes.

Asimismo, la elección de este proyecto responde a una propuesta presentada por la Dra.

Carmen Paz Suárez Araujo y el Dr. Pablo Carmelo Fernández López, cuyo enfoque despertó mi interés, tanto por su relevancia como por el desafío técnico que representa.

La decisión de centrarme específicamente en la enfermedad de Alzheimer no fue casual. Una persona muy cercana a mí ha vivido de primera mano el proceso de deterioro de un familiar afectado por esta enfermedad. Acompañar de cerca esa experiencia me permitió comprender el profundo impacto no sólo en el paciente, sino también en su entorno más cercano. El Alzheimer no se limita a la pérdida de memoria o a la dificultad para realizar tareas básicas; implica la progresiva pérdida de la identidad, de los recuerdos que conforman quiénes somos. Envejecer es parte de la vida, y aunque con los años nuestras capacidades disminuyen, la compañía de nuestros seres queridos suele ser el mayor consuelo. Sin embargo, perder la capacidad de reconocerlos, de recordarlos, supone (a mi entender) una forma de muerte en vida, donde la desconexión con uno mismo es total.

Por todo ello, este proyecto representa para mí una oportunidad para contribuir, aunque sea modestamente, al avance en la detección temprana de esta enfermedad. Espero que el trabajo aquí desarrollado pueda ser un pequeño paso hacia soluciones más eficaces que ayuden a mejorar la vida de quienes se enfrentan al Alzheimer, así como la de quienes los acompañan en ese difícil camino.

1.2.2. Objetivos

Los objetivos inicialmente previstos y desarrollados a lo largo del proyecto han sido principalmente tres. A continuación se describen los mismos, así como los trabajos adicionales que se han realizado, y que han supuesto una modificación y aumento en el alcance definitivo de este TFG.

- ✔ **Objetivo 1:** El objetivo uno, consiste en desarrollar un modelo de inteligencia artificial basado en DenseNet capaz de procesar imágenes de resonancia magnética (MRI) enfocadas en el diagnóstico neurológico, específicamente enfocado en la detección de Enfermedad de Alzheimer (AD) y Deterioro Cognitivo Leve (MCI). A lo largo del desarrollo de este objetivo se planteó el uso de un modelo preentrenado, bajo está premisa, se realizó una comparación entre un modelo DenseNet preentrenado y uno no preentrenado para evaluar su desempeño y su posterior uso en los siguientes objetivos.
- ✔ **Objetivo 2:** Analizar los resultados obtenidos del modelo mediante un estudio comparativo con otras soluciones computacionales basadas en redes neuronales artificiales (RNA), utilizando, esencialmente, datos basados en el resultado de tests neuropsicológicos para el diagnóstico temprano de la Enfermedad de Alzheimer (AD). Lo anterior ha permitido evaluar su precisión, eficacia y utilidad como herramienta complementaria o alternativa en la práctica clínica.
- ✔ **Objetivo 3**: Analizar imágenes de MRI de pacientes en tres fases temporales: línea base (BL), 12 meses (M12), y 24 meses (M24), para identificar patrones que permitan detectar la evolución del Deterioro Cognitivo Leve (MCI) y la Enfermedad de Alzheimer (AD).

Dado que no se pudo contar con la totalidad de las pruebas MRI en los momentos temporales que nos propusimos estudiar (BL, M12 y M24), se optó por reducir las fases temporales a Screening (SC), 6 meses (M06), y 12 meses (M12). Asimismo, los resultados de estos diagnósticos fueron utilizados para realizar una comparación de un modelo que utiliza estos datos de historial frente a con un modelo equivalente en el que no se incorpora el historial clínico completo.

1.3. Aportaciones del trabajo

1.3.1. Principales aportaciones

Este proyecto ha sido realizado en colaboración con la División de Neurociencia Computacional del Grupo de Investigación de Computación Inteligente, Percepción y Big Data (CIPERBIG) de la Universidad de las Palmas de Gran Canaria, para posteriormente anexarlo con sus futuras investigaciones, así como para evaluar y observar el comportamiento de los diferentes modelos aplicados en el ámbito de estudio. De este modo, desde el punto de vista técnico-científico, este proyecto constituye una contribución crucial para el aumento de la eficiencia, de posteriores investigaciones, y supone una aportación de datos y métricas potencialmente útiles para futuras implementaciones.

Por otro lado, la principal aportación de este proyecto trata sobre el diagnóstico temprano del Alzheimer, haciendo uso de MRI dado que permite distinguir las alteraciones neuropatológicas vinculadas a estas enfermedades [15].

Desde el punto de vista socioeconómico, se ha desarrollado un sistema inteligente automático que ayuda al diágnostico temprano del Alzheimer, siendo este un punto crítico para el posterior tratamiento del apciente, y correspondiéndose esta con la etapa más importante para retasar la aparición de los síntomas de la enfermedad y el deterioro cognitivo de la persona, tal como se mostró en la figura 1.2.

1.3.2. Competencias específicas

Este proyecto guarda estrecha relación con las competencias asociadas a las ramas de a Computación Inteligente, Aprendizaje Automático e Ingeniería del Software. Dentro de las competencias específicas de la titulación del Grado de Ingeniería Informática, destacan:

CI1: Esta competencia trata sobre la capacidad para diseñar, desarrollar, seleccionar y evaluar aplicaciones y sistemas informáticos, asegurando su fiabilidad, seguridad y calidad, conforme a principios éticos y a la legislación y normativa vigente. Dicha competencia se adecua a la naturaleza del trabajo, dado a que para la realización del modelo de aprendizaje fue necesario implementar, diseñar y evaluar un sistema que se comportase de manera correcta.

CI7: Esta competencia trata sobre el conocimiento, diseño y utilización de forma eficiente de los tipos y estructuras de datos más adecuados a la resolución de un problema. Este

apartado, cobra importancia dado a que un sistema inteligente necesita datos para entrenar, esos datos, deben ser estudiados, ordenados y estructurados de la manera más idónea para posteriormente alimentar al sistema.

CI15: Esta competencia trata sobre el conocimiento y aplicación de los principios fundamentales y técnicas básicas de los sistemas inteligentes y su aplicación práctica. Esta competencia se cumple, ya que ha sido esencial la comprensión de los distintos modelos y técnicas de evaluación y validación de Aprendizaje Automático, así como la implementación de la Arquitectura DenseNet.

CI6: Esta competencia trata sobre el 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. Las distintas operaciones con los datos como su procesamiento de las MRI a imágenes dió lugar a soluciones que se iban implementando a medida que se avanzaba en el proyecto.

1.3.3. Alineamiento con los objetivos de desarrollo sostenible

ODS	Grado de relación			
	0 No procede	1 Bajo	2 Medio	3 Alto
1 Fin de la Pobreza	x			
2 Hambre cero	x			
3 Salud y Bienestar				x
4 Educación de calidad	x			
5 Igualdad de género	x			
6 Agua limpia y saneamiento	x			
7 Energía Asequible y no contaminante	x			
8 Trabajo decente y crecimiento económico			x	
9 Industria, Innovación e Infraestructuras			x	
10 Reducción de las desigualdades	x			
11 Ciudades y comunidades sostenibles	x			
12 Producción y consumo sostenibles	x			
13 Acción por el clima	x			
14 Vida submarina	x			
15 Vida de ecosistemas terrestres	x			
16 Paz, justicia e instituciones sólidas	x			
17 Alianzas para lograr objetivos			x	

Tabla 1.2: Grado de relación del TFT con los objetivos de desarrollo sostenible.

1.3.3.1. Salud y Bienestar

https://www.un.org/sustainabledevelopment/es/health/

Este proyecto se alinea de forma directa con el Objetivo de Desarrollo Sostenible (ODS) número 3: Salud y Bienestar, propuesto por la Organización de las Naciones Unidas en la Agenda 2030. Este objetivo promueve ”3.b Apoyar las actividades de investigación y desarrollo de vacunas y medicamentos para las enfermedades transmisibles y no transmisibles que afectan primordialmente a los países en desarrollo ”, y dentro de sus metas específicas incluye la necesidad de fortalecer la capacidad de los países para la detección temprana, prevención y tratamiento de enfermedades no transmisibles como las neurodegenerativas. (Meta 3.d)

Desde esta perspectiva, el presente trabajo tiene como objetivo contribuir a esa mejora del bienestar a través del desarrollo de un sistema automatizado de ayuda al diagnóstico, que emplea técnicas de aprendizaje profundo para analizar neuroimágenes. La integración de herramientas de inteligencia artificial en el ámbito médico no solo puede reducir la carga de trabajo de los profesionales sanitarios, sino también incrementar la velocidad y exactitud del diagnóstico, favoreciendo una atención médica más equitativa, oportuna y efectiva.

1.3.3.2. Trabajo decente y crecimiento económico

https://www.un.org/sustainabledevelopment/es/economic-growth/

Este proyecto presenta un grado de relación medio con el Objetivo de Desarrollo Sostenible (ODS) número 8: Trabajo decente y crecimiento económico, en particular con la meta 8.2, que plantea “lograr niveles más elevados de productividad económica mediante la diversificación, la modernización tecnológica y la innovación, entre otras cosas centrándose en los sectores con gran valor añadido y un uso intensivo de la mano de obra”.

La propuesta desarrollada en este trabajo contribuye a esta meta al integrar un sistema inteligente basado en aprendizaje profundo para el análisis automatizado de neuroimágenes en el diagnóstico temprano del Alzheimer. Esta tecnología representa una forma concreta de modernización e innovación dentro del sector sanitario, permitiendo aumentar la productividad del personal médico al reducir el tiempo requerido en la evaluación de imágenes médicas y en la toma de decisiones clínicas.

Lejos de reemplazar al profesional, el sistema actúa como una herramienta de apoyo, lo que optimiza el uso de los recursos humanos cualificados, permitiéndoles centrarse en tareas de mayor valor añadido, como el tratamiento personalizado o la atención directa al paciente. Además, esta transformación tecnológica puede impulsar el desarrollo de nuevos perfiles laborales especializados en el cruce entre medicina, tecnología e inteligencia artificial, favoreciendo la creación de empleo cualificado y la sostenibilidad económica de los sistemas sanitarios en contextos de alta demanda asistencial.

1.3.3.3. Industria, Innovación e Infraestructuras

https://www.un.org/sustainabledevelopment/es/infrastructure/

Este proyecto guarda una relación media con el Objetivo de Desarrollo Sostenible (ODS) número 9: Industria, Innovación e Infraestructuras, especialmente con su meta 9.1, que promueve "desarrollar infraestructuras fiables, sostenibles, resilientes y de calidad [...] para apoyar el desarrollo económico y el bienestar humano, haciendo especial hincapié en el acceso asequible y equitativo para todos".

La propuesta técnica presentada en este trabajo representa una contribución directa al avance de las infraestructuras digitales en el ámbito médico. El desarrollo de un sistema inteligente para la detección temprana del Alzheimer mediante el análisis de neuroimágenes constituye una forma de infraestructura tecnológica avanzada, orientada a fortalecer el sistema de salud, hacerlo más resiliente y mejorar su capacidad de respuesta ante enfermedades neurodegenerativas.

Este tipo de soluciones no solo optimizan la eficiencia clínica, sino que también democratizan el acceso a herramientas diagnósticas precisas y modernas, especialmente en contextos donde la disponibilidad de especialistas es limitada. De este modo, se favorece un acceso más equitativo a la salud, alineado con los principios de sostenibilidad y bienestar social.

Además, este tipo de innovación tecnológica impulsa el desarrollo de una industria emergente en la intersección entre inteligencia artificial y biomedicina, que tiene el potencial de convertirse en una infraestructura clave para el futuro de la atención médica a escala global.

1.3.3.4. Alianzas para lograr objetivos

https://www.un.org/sustainabledevelopment/es/infrastructure/

Este proyecto presenta una relación media con el Objetivo de Desarrollo Sostenible (ODS) número 17: Alianzas para lograr objetivos, especialmente con sus metas técnologicas 17.6 y 17.7. Por un lado, la herramienta desarrollada en este Trabajo de Fin de Grado propone una tecnología accesible para todos, sin necesidad de ser un especialista. Aunque no fue creada con el propósito de sustituir a profesionales, sino como herramienta de ayuda a los facultativos, puede utilizarse como una herramienta en contextos donde estos no estén disponibles, lo que contribuye significativamente a su acceso y difusión global, en línea con lo planteado en la meta 17.6.

Además, el uso de este proyecto en el ámbito de la investigación en neuroimagen o detección de enfermedades como el Alzheimer, no solo contribuye al avance científico, sino que sienta las bases para cooperación Sur-Sur y triangular, permitiendo que investigadores de diferentes países colaboren, compartan datos, algoritmos y resultados, fortaleciendo así los mecanismos de innovación tecnológica con un enfoque inclusivo y accesible.

Capítulo 2

Desarrollo de una solución computacional para la detección temprana del AD

Para abordar la solución computacional se dividió el problema a resolver en 4 componentes interrelacionados, que en su conjunto conforman la solución integral.

1. Preprocesamiento del conjunto de datos: En este componente se llevaron a cabo una serie de desarrollos con el objetivo de generar una fuente de datos confiable y sólida para la entrada del sistema.
2. Sistema de Proceso: En este componente se llevó a cabo un programa que contenía el modelo DenseNet121, asimismo , se desarrollaron funciones específicas para su configuración, entrenamiento y evaluación, originando las métricas correspondientes a sus ejecuciones.
3. Resultados: Este componente se basa en el estudio detallado de los resultados, para su posterior extracción de conclusiones respaldadas por el funcionamiento del sistema desarrollado.
4. Validación: Este componente se centra en el análisis de los resultados para verificar el correcto funcionamiento del clasificador de imágenes y su eficiencia. Asimismo, se enfoca en el desarrollo de métodos de validación como el cross-site validation, para comprobar la integridad y robustez del sistema.

Todo estos componentes y sus interacciones entre ellos pueden apreciarse en la figura 2.1

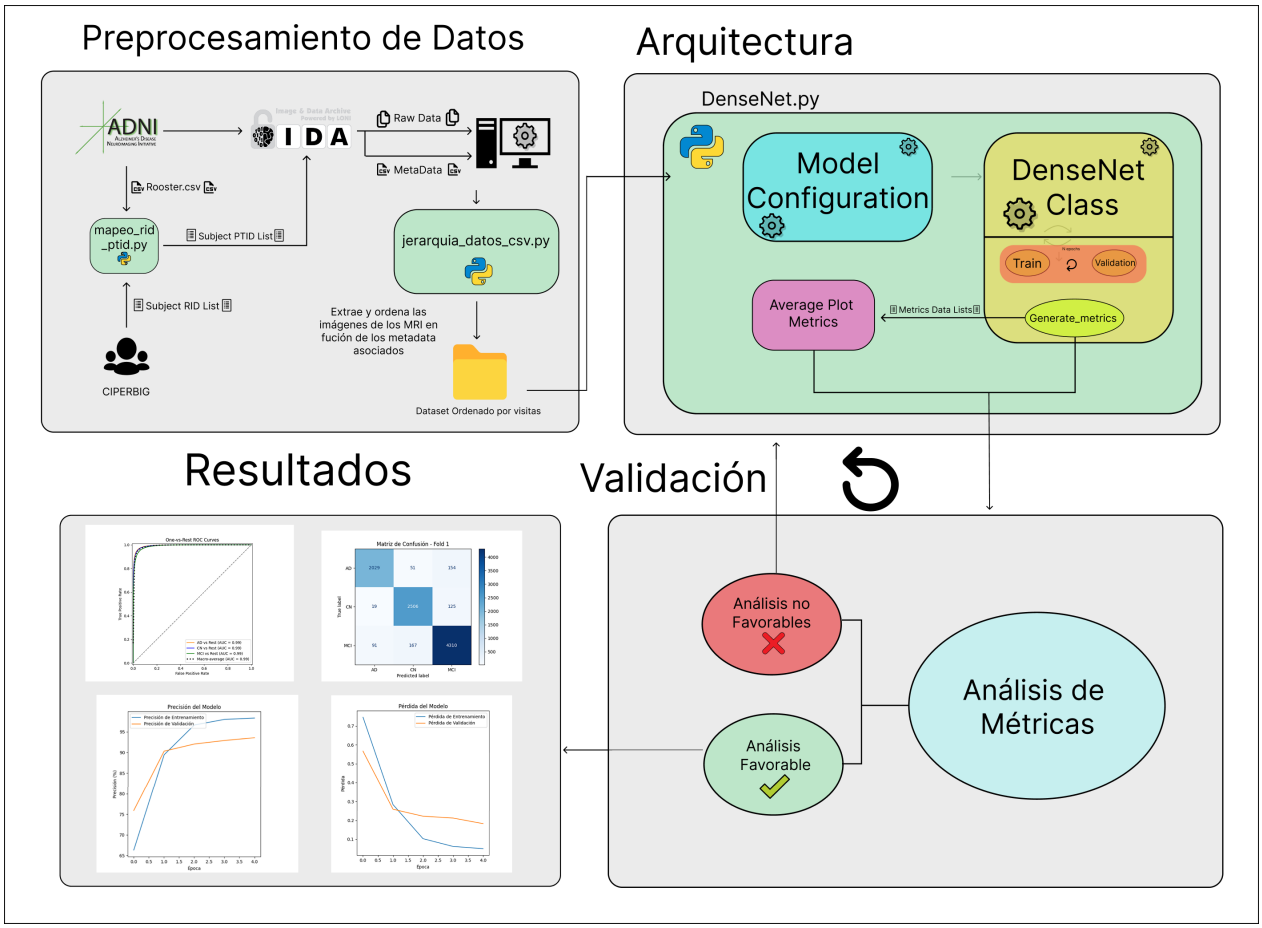


Figura 2.1: Esquema gráfico del desarrollo de la solución computacional.

A lo largo de este capítulo, se explicarán con detalle los dos primeros componentes, mientras que los dos últimos se abordarán en el capítulo siguiente.

2.1. Metodología

Dado que este TFG se centra en el desarrollo de un sistema de inteligencia artificial para el análisis de neuroimágenes, se adoptará una metodología estructurada para asegurar un enfoque sistemático y efectivo en cada etapa del proceso.

Dicha metodología usada es la espiral, donde cada iteración estará compuesta de las siguientes fases:

- ✔ **Fase de Definición del Problema**: Donde identificamos y analizamos de forma exhaustiva el problema, o tarea, del problema que se desea abordar. Así como la determinación de los recursos tecnológicos y de datos necesarios para llevar a cabo dicha tarea y desarrollo asociado.

- ✔ **Fase de Diseño del Sistema**: Se llevará a cabo un estudio de la base de datos ADNI para identificar y diseñar una estructura de datos que posteriormente, tras una serie de operaciones se integrará con la arquitectura de DenseNet. Además se estudiarán las características y funcionalidades de esta arquitectura en el contexto del análisis de neuroimágenes, así como su integración con las técnicas de aprendizaje profundo (Deep Learning).
- ✔ **Fase de Construcción y Realización de Pruebas del Sistema:** En donde comenzaremos con el desarrollo e implementación de la arquitectura neuronal, así como con su entrenamiento, utilizando el conjunto de datos de MRI. Durante esta etapa, se ajustarán los hiperparámetros del modelo para optimizar su rendimiento y se realizarán pruebas para evaluar su comportamiento. Asimismo, se implementarán técnicas de validación para garantizar que el modelo no esté sobre-ajustado y que su capacidad de generalización sea adecuada.
- ✔ **Fase de Estudio de Resultados y Comparación Cualitativa:** Se compararán los resultados con los métodos de diagnóstico convencionales y otras soluciones existentes en la bibliografía.

2.2. Especificaciones de la infraestructura tecnológica utilizada en el desarrollo

Para el desarrollo de este proyecto los miembros del grupo de investigación CIBERPIG han puesto a disposición un equipo para el correcto desempeño de la evaluación del TFG. Este equipo forma parte de un puesto de trabajo que se encuentra en el laboratorio 7 del edificio de Informática y Matemáticas, perteneciente al grupo de investigación. Asimismo, también se ha usado mi portátil personal para el desarrollo del mismo. En la figura 2.1 podemos ver la comparativa del hardware de ambos.

Categoría	**HP ENVY Desktop TE02-0xxx**	**Laptop Aspire A515-56**
Procesador	12th Gen Intel® Core ™ i7-12700, 2100 MHz, 12 procesadores principales, 20 procesadores lógicos	11th Gen Intel(R) Core(TM) i7-1165G7 @ 2.80GHz
Gráfica	NVIDIA GeForce RTX 3060, 12GB	Intel(R) Iris(R) Xe Graphics (128 MB)
Sistema Operativo	Windows 11 Home, versión 23H2	Windows 10 Home
Memoria RAM	32 GB	8 GB
Almacenamiento	1 TB	477 GB SSD

Tabla 2.1: Comparativa de características entre HP ENVY Desktop y Laptop Aspire

2.2.1. Librerías utilizadas en el desarrollo

En este TFG se utilizó Python 3.9.21, debido a su compatibilidad con las versiones recientes de las principales bibliotecas de aprendizaje automático empleadas.

Entre las herramientas más relevantes utilizadas en el entorno de desarrollo se encuentran:

- ✔ PyTorch (versión 1.12.0 y 2.6.0): Esta biblioteca fue fundamental para la construcción e implementación de modelos de redes neuronales profundas. Se utilizó junto con CUDA 11.3 para aprovechar la aceleración por GPU. Enlace a la documentación: [19]
- ✔ Scikit-learn (versión 1.6.1): Biblioteca que proporciona herramientas eficientes para el análisis predictivo y modelos clásicos de machine learning. Enlace a la documentación: [20]
- ✔ Torchvision (versión 0.21.0): Complemento de PyTorch especializado en visión por computadora, con modelos preentrenados y transformaciones comunes. Esta contiene el modelo DenseNet121 utilizado. Enlace a la documentación: [21]
- ✔ Nibabel (versión compatible con Python 3.9): Librería que permite leer, escribir y manipular archivos de neuroimagen en formato NIfTI. NIfTI (Neuroimaging Informatics Technology Initiative) es un formato de archivo ampliamente utilizado en el ámbito de las neuroimágenes, que se utiliza principalmente para almacenar imágenes cerebrales obtenidas por técnicas como la resonancia magnética (MRI) [22] cuyo origen proviene del formato de archivo ANALYZE™ 7.5.[23].Enlace a la documentación: [24]

2.3. Método para el tratamiento inteligente de los datos

El preprocesamiento del conjunto de datos es una parte fundamental en cualquier Sistema Inteligente, ya que dicho proceso es el responsable de obtener los datos que son entrada del sistema y con lo que realizará su entrenamiento y sacará los mapas de características. El objetivo final de este componente es obtener un conjunto de datos bien estructurados y etiquetados que actúe de entrada al modelo. Al tratarse de un modelo CNN bidimensional (2D), hemos de trabajar primero con el MRI, que viene en el formato médico .nii para extraer las imágenes del mismo, ordenarlas en función de su plano(axial, coronal y sagital) y diagnóstico, y para agregarlas al fichero de datos que constituirá la fuente de datos de nuestro proyecto.

De este modo, el esquema final del preprocesamiento de datos se corresponde al mostrado en la figura 2.2:

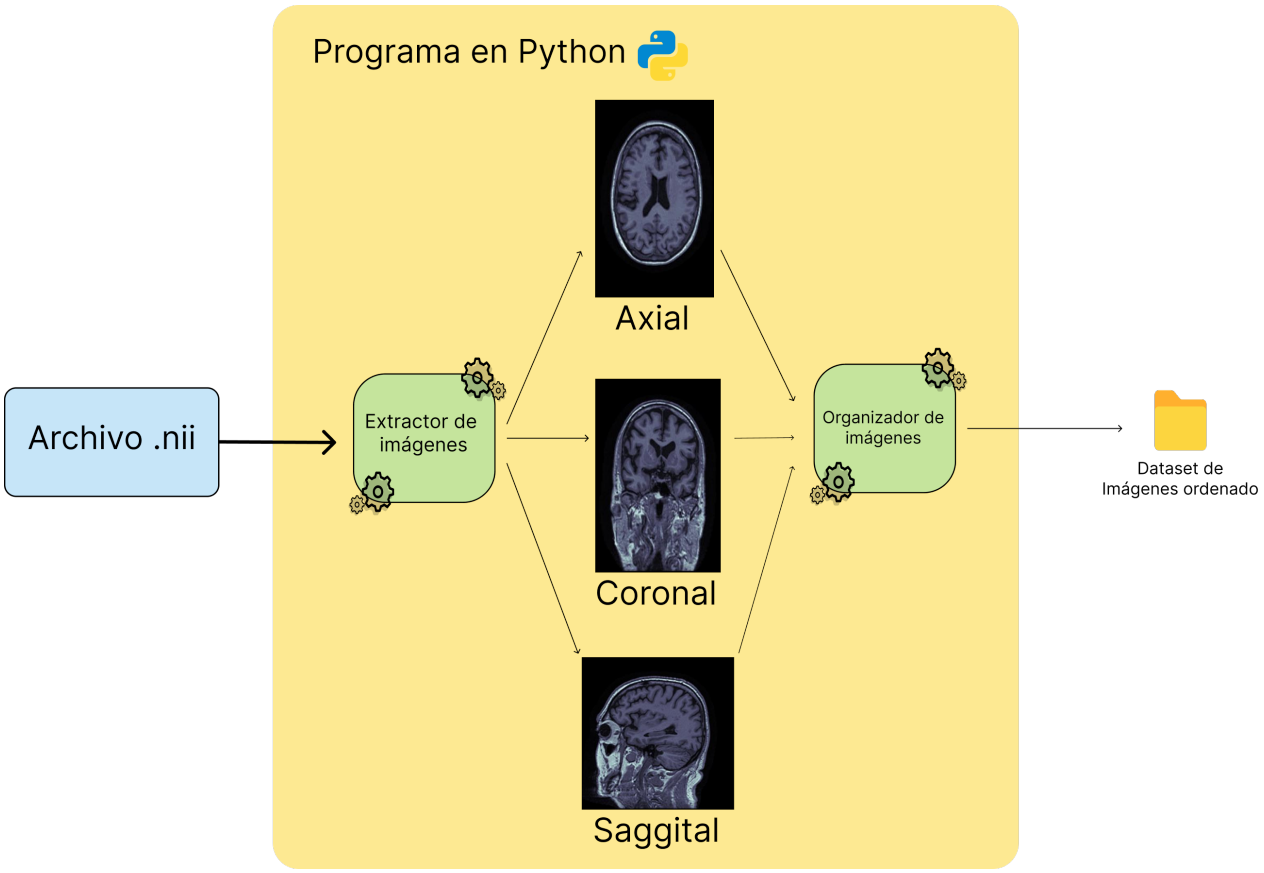


Figura 2.2: Esquema gráfico de la implementación y proceso de preprocesamiento del conjunto de datos.

En la primera fase del proyecto se utilizó un conjunto de datos de prueba de 9 pacientes procedentes de ADNI . Primero se tuvo que definir, la estructura de los directorios que queríamos para nuestra fuente de datos final, es decir, la estructura de los datos al final del preprocesamiento de los mismos. De este modo , se concluyó que la mejor manera de organizar los datos era siguiendo la estructura que podemos observar en la figura 2.3.

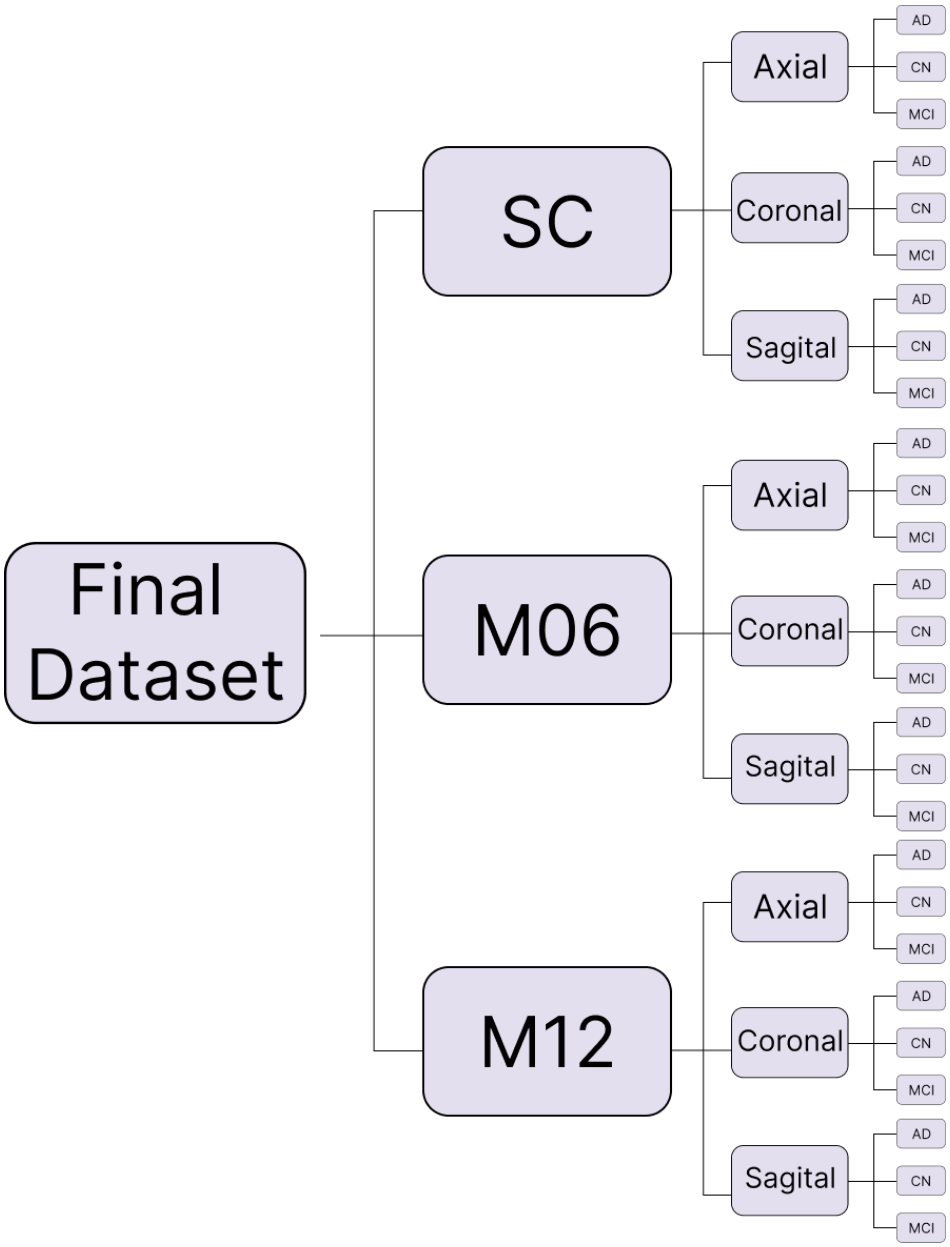


Figura 2.3: Estructura de Directorios final de Datos.

Esta estructura se adapta a nuestros objetivos, ya que nos permite discernir entre lo que

llamaremos fases de tiempo y planos de visualización.

Por fases de tiempo nos referimos a los distintos momentos en el seguimiento clínico del paciente: screening, sexto mes (m06) y doceavo mes (m12). Estas fases representan las marcas de tiempo de las visitas realizadas por el paciente y su correspondiente estudio asociado a dicha visita. Por otro lado, los planos de visualización, hacen referencia a las distintas orientaciones espaciales en las que se segmentan las imágenes de resonancia mágnetica MRI (Axial, Coronal y Sagital).

Discernir entre las distintas fases del tiempo es importante ya que usualmente las investigaciones en el ámbito médico suele venir en base al período de tiempo (BS) Baseline, en nuestro caso, dado a las limitaciones de los datos de ADNI, y para adaptarnos a los 819 pacientes del conjunto de datos de trabajo utilizado en el estudio [25], se optó por escoger el Screening(SC), ya que no tenemos muestras del período Baseline.

Esta limitación se debe a que, en el estudio ADNI, los nuevos participantes son evaluados inicialmente en una visita de "cribado" (*screening*). Si no son excluidos, ya sea por no cumplir con los criterios de inclusión/exclusión o por pérdida en el seguimiento, avanzan a una segunda visita denominada "línea base" (*baseline*), donde se completan evaluaciones y tareas adicionales del estudio.

En cuanto al diagnóstico durante las fases de cribado y línea base, la visita de cribado incluye un subconjunto de evaluaciones planificadas (como pruebas neuropsicológicas, neuroimagen, etc.) que permiten determinar si el participante cumple los criterios necesarios para su inclusión.

Una vez superada esta etapa, se lleva a cabo la evaluación de línea base, que incorpora pruebas neuropsicológicas adicionales, estudios de imagen y toma de muestras biológicas. En la mayoría de los casos, el diagnóstico asignado durante el cribado coincide con el determinado en la línea base, siendo así una solución ideal para nuestra limitación [14].

Ahora que hemos definido la estructura final, afrontamos la siguiente cuestión. Extraer las imagenes de los archivos MRI .nii. Los datos,a pesar de ser un dataset de prueba, eran extensos, y de cara a la implementación final con el dataset de 819 pacientes, era necesaria la creación de un software que automatizase esta tarea, y que cumpliese estos objetivos:

- ✔ Crear un Dataset Final clasificado y ordenado en función de las visitas.
- ✔ Filtrar por el conjunto de datos de ADNI el archivo .nii y sus metadatos para extraer y clasificar sus imagénes asociadas.

Para lograr esto, el primer pasó que se realizó fue estudiar y analizar los datos del conjunto de ADNI. Observando lo siguiente: El conjunto de datos ADNI sigue siempre la estructura de directorios mostrada en la figura 2.4.

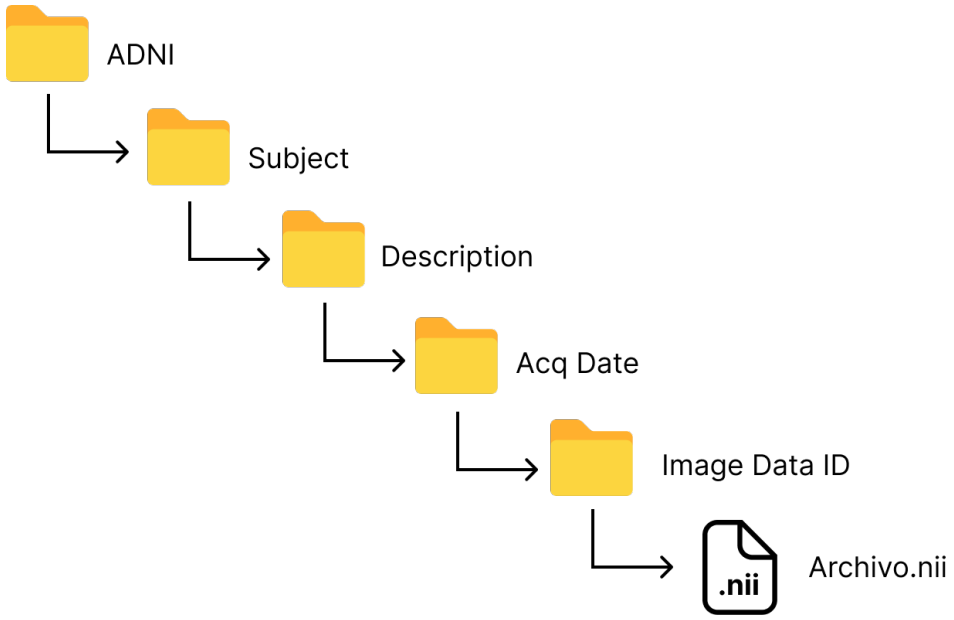


Figura 2.4: Figura que muestra de manera gráfica la estructura que sigue el directorio descargado de ADNI.

Como observamos, tenemos una serie de directorios que siguen un formato específico. Por tanto, para automatizar el proceso de búsqueda del archivo .nii necesitamos conocer los siguientes datos de búsqueda:

- ✔ **Identificador del sujeto:** representado por la etiqueta `Subject`.
- ✔ **Identificador de la visita:** representado por la etiqueta `visitIdentifier`.
- ✔ **UID de la imagen:** representado por la etiqueta `/Image Data UID`.
- ✔ **Fecha de adquisición de la imagen:** representada por la etiqueta `Acq Date`.
- ✔ **Etiqueta de los datos procesados:** representada por la etiqueta `Description`.
- ✔ **Grupo de Investigación:** representado por la etiqueta `researchGroup`.

Con estos datos de búsqueda, podremos encontrar el MRI (archivo.nii) correspondiente a cada uno de los pacientes en estudio, extraer las imágenes del. nii y clasificar las imágenes en base a estos. Por otro lado, los datos en ADNI, siempre van acompañados de un fichero .csv de metadatos, este fichero es crítico para nuestro cometido, ya que sin él no es posible obtener los datos de búsqueda, ni organizarlos, ni asignar las etiquetas correctas a los mismos.

Por ende, concluimos que para extraer las imágenes y clasificarlas necesitamos:

- ✔ El archivo .nii
- ✔ Los metadatos asociados a cada uno.

De este modo se desarrolló un software para este cometido. En primer lugar, para localizar el archivo .nii debiamos obtener los metadatos del mismo, por lo que se creó una función

llamada process_from_csv que recorre el .csv, extrae los datos clave fila por fila y los mandaba a otra función llamada `process_neuro_images`. Ambas implementaciones se pueden observar en la figura 2.5 y en la figura 2.6.

```python
def process_from_csv(csv_path):
    df = pd.read_csv(csv_path)

    for _, row in df.iterrows():
        subject_identifier = row["Subject"]
        visit_identifier = row["Visit"]
        image_uid = row["Image Data ID"]
        original_date = row["Acq Date"]
        fecha_dt = datetime.strptime(original_date, "%m/%d/%Y")
        date_acquired = fecha_dt.strftime("%Y-%m-%d")
        data_label = row["Description"]
        research_group = row["Group"]

        print(f"\nProcesando: {subject_identifier}, {image_uid}")
        process_neuro_images(subject_identifier, visit_identifier, image_uid, date_acquired, data_label, research_group)
```

Figura 2.5: Implementación de filtro sobre los metadatos

Como podemos ver en la figura 2.5, leemos y recorremos las columnas del archivo csv para extraer los valores correspondientes a aquellas cuyo nombre coincide con los datos de interés comentados anteriormente.

```python
    date_folder = None
    for folder in os.listdir(os.path.join(subject_folder, processed_folder)):
        if folder.startswith(date_acquired):
            potential_path = os.path.join(subject_folder, processed_folder, folder, image_uid)
            if os.path.isdir(potential_path):
                date_folder = folder
                break
    if not date_folder:
        print(f"Error: No se encontró la carpeta de la fecha {date_acquired} en {subject_folder}/{processed_folder}")
        return

    image_folder = os.path.join(subject_folder, processed_folder, date_folder, f"{image_uid}")
    nii_files = [f for f in os.listdir(image_folder) if f.endswith(".nii")]

    if not nii_files:
        print(f"Error: No se encontró el archivo .nii en {image_folder}")
        return

    nii_path = os.path.join(image_folder, nii_files[0])

    neuro_image = load_neuro_image(nii_path)
```

Figura 2.6: Implementación de la búsqueda automática del archivo .nii y su extracción de imágenes

Tras esto, como observamos en la figura 2.6 recorremos la estructura de directorios en función de los datos asociados, correspondientes a los extraídos del archivo .csv, para poste-

riormente extraer las imágenes y organizarlas siguiendo la estructura explicada en la figura 2.3

De cara a la implementación asociada con el conjunto final de trabajo, compuesta por los 819 pacientes se siguió el siguiente esquema.

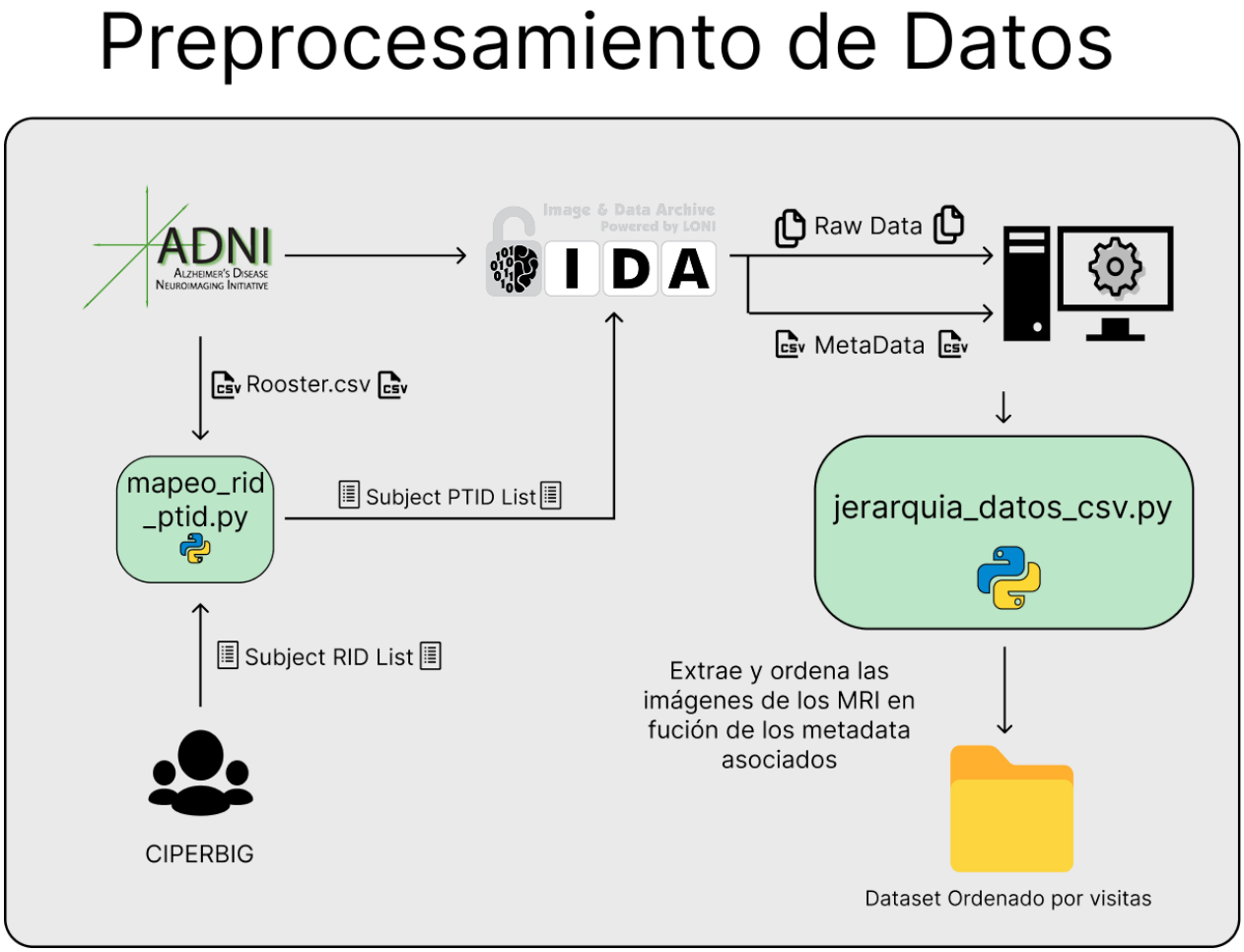


Figura 2.7: Esquema de implementación del preprocesado de datos simplificado

Ahora que tenemos un proceso de obtención y separación de las imágenes automatizado, se dio comienzo a la siguiente fase. En esta segunda fase, trabajamos con el conjunto completo de datos, que ha sido utilizado en el trabajo de investigación [25] publicado por el equipo de CIPERBIG, no obstante, dicho trabajo no constaba de datos en imágenes, sino con otro tipo de datos. Por ello, se entró en contacto con el equipo para obtener el Subject_ID de los pacientes, con los que posteriormente se intruduciría en la IDA [26] para obtener todos los datos correspondientes a estos, para posteriormente extraer sus imágenes y construir la fuente de Datos definitiva. No obstante, en este estudio se trabajó con otro parámetro de ADNI, el RID. Este parámetro no es válido para la búsqueda avanzada de la base de datos IDA [26]

de ADNI. Tras un estudio más detallado, se descubrió que el SUBJECT_IDENTIFIER, es un parámetro compuesto por: idHospital_S(de Subject)_RID

Tras esta correlación se desarrolló una función, llamada mapeo_rid_ptid.py capaz de mapear la lista de RID dada por el equipo de investigación, para ello ahora hacía falta una lista con todos los parametros para buscar el PTID(Subject Identifier) correspondiente al RID, siendo esta el ROOSTER.csv publicado en ADNI, publicado en la sección FAQ por parte de los investigadores al cargo.

```
51  # Cargar el CSV
52  df = pd.read_csv(input_file)
53
54  filtered_df = df[(df["RID"].isin(rids)) & (df.iloc[:, 0] == "ADNI1")]
55
56  filtered_df = filtered_df.drop_duplicates()
57  # Guardar a un nuevo archivo
58  filtered_df.to_csv(r"C:\Users\Eric\Desktop\Novus Initium\Ars Discendi\TFG\TFG_ADNI.csv", index=False)
59  ptid_list = filtered_df["PTID"].tolist()
60  ptid_text = ",".join(ptid_list)
61
62  # Guardar el TXT
63  with open(r"C:\Users\Eric\Desktop\Novus Initium\Ars Discendi\TFG\TFG_PTID.txt", "w") as f:
64      f.write(ptid_text)
65
66  print("Archivo filtrado guardado como TFG_ADNI.csv")
67  print("Archivo de PTIDs guardado como TFG_PTID.txt")
68
```

Figura 2.8: Implementación en python del mapeado de RID sobre el fichero ROOSTER.csv

Asimismo, el tutor me otorgó el conjunto de datos ADNI1_Complete_1Yr_1.5T_4_26_2025 perteneciente a las colecciones de datos preprocesadas por IDA [26], sin embargo, esta colección carecía de todos los pacientes. Por ende, se implementó en el mapeo_rid_ptid.py, una variable que almacenaba los pacientes que faltaban, obteniendo así su PTID correspondiente.

A partir de aquí surgieron dos vertientes, que posteriormente se unirían para crear la fuente de datos final. Tras la descarga masiva de información mediante la herramienta de búsqueda avanzada del portal IDA[26], obtuvimos un conjunto de datos en crudo que incluía información heterogénea, parte de la cual no era útil para los fines de este proyecto, ya que se encontraba en formatos distintos al que se requería. Por un lado, fue necesario realizar un cribado exhaustivo de los metadatos, con el objetivo de conservar únicamente aquellos que coincidían con el formato y criterios establecidos. Por otro lado, fue preciso llevar a cabo un filtrado equivalente sobre los datos de imagen, localizando y separando aquellos archivos que correspondían específicamente al formato .nii, el cual constituye el estándar de trabajo en este estudio.

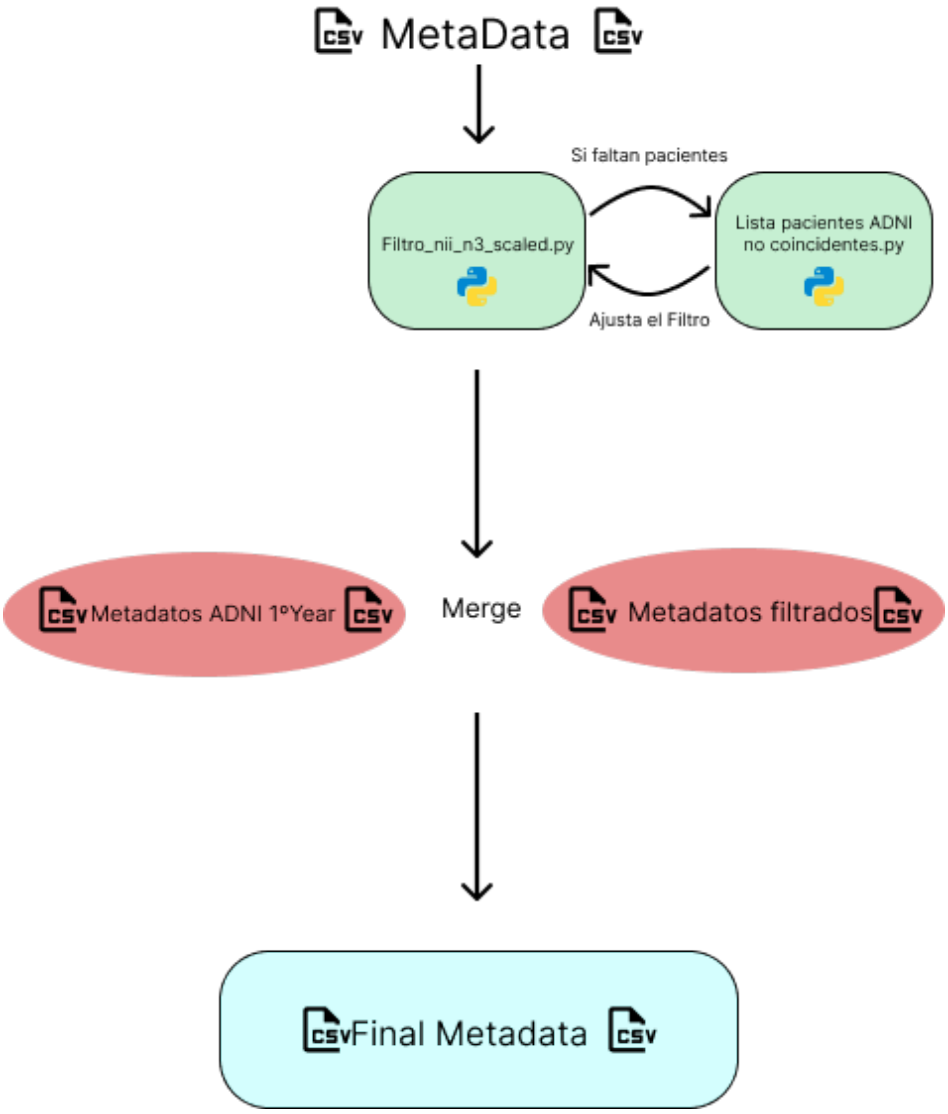


Figura 2.9: Esquema simplificado del proceso de filtro sobre los metadatos

El primer paso para obtener los metadatos de todos los pacientes, es filtrar los metadatos del conjunto total de los pacientes restantes por tipo de estudio (Columna Description), ya que necesitamos que sean los mismos tipos que el conjunto ADNI1_Complete_1Yr_1.5T_4_26_2025, de este modo, se implementó un programa python llamado filtrar_csv_description.py

```
3
4   input_file = r"C:\Users\Eric\Desktop\Novus Initium\Ars Discendi\TFG\Csvs Metadata\Sin filtro\m06,sc,m12_4_29_2025.csv"
5
6
7   df = pd.read_csv(input_file)
8
9
10  valores_validos = [
11      "MPR; GradWarp; B1 Correction; N3; Scaled",
12      "MPR; GradWarp; B1 Correction",
13      "MPR; GradWarp; N3; Scaled"
14      "MPR; GradWarp; N3",
15      "MPR; ; N3; Scaled",
16      "MPR; ; N3; Scaled_2",
17      "MPR; GradWarp; B1 Correction; N3; Scaled_2",
18      "MPR; GradWarp; B1 Correction; N3",
19      "MPR-R; GradWarp; B1 Correction; N3; Scaled",
20  ]
21
22  filtered_df = df[df['Description'].isin(valores_validos)]
23
```

Figura 2.10: Implementación en python del filtro de los metadatos, donde se definen los tipos de estudios válidos.

Como podemos ver, en la figura 2.10 los valores válidos del tipo de estudio correspondientes a los archivos .nii son los siguientes:

- ✔ ”MPR; GradWarp; B1 Correction; N3; Scaled”,
- ✔ ”MPR; GradWarp; B1 Correction”,
- ✔ ”MPR; GradWarp; N3; Scaled”
- ✔ ”MPR; GradWarp; N3”,
- ✔ ”MPR; ; N3; Scaled”,
- ✔ ”MPR; ; N3; Scaled_2”,
- ✔ ”MPR; GradWarp; B1 Correction; N3; Scaled_2”,
- ✔ ”MPR; GradWarp; B1 Correction; N3”,
- ✔ ”MPR-R; GradWarp; B1 Correction; N3; Scaled”,

De esta forma, recorremos el csv en busca de aquellas imágenes que pertenezcan a este tipo de estudio (Que son las de formato .nii). No obstante, esta solución mostró otra limitación del dataset, ya que no se han realizado los mismos tipos de estudio a todos los pacientes, por lo que había que ajustar el filtro. Para poder hacerlo, se desarrolló una función adicional que permite identificar a aquellos pacientes que no presentan ninguno de estos tipos de estudio, posteriormente, se analizarían y se añadirían los diferentes tipos de estudios de éstos, para poder tener mínimamente una imagen pertenecientes a estos, de manera que al final, tengamos una muestra mínima de cada paciente. Tras ello, se ajusto el filtro. Dando lugar a un .csv con todos los metadatos de los .nii de los pacientes. Finalmente, se fusionó este .csv con el .csv perteneciente al ADNI1_Complete_1Yr_1.5T_4_26_2025, obteniendo como resultado, un .csv de todos los metadatos asociados a todos los .nii de los 819 pacientes.

A continuación, había que repetir el mismo proceso, pero con los datos, es decir, encontrar todos los archivos .nii dentro de los datos sin filtrar pertenecientes a los mismos tipos de estudios comentados anteriormente con el fin de asegurar que los mismos archivos .nii estén reflejados en el archivo .csv previamente generado. De este modo, se procedió tal como muestra la figura 2.11.

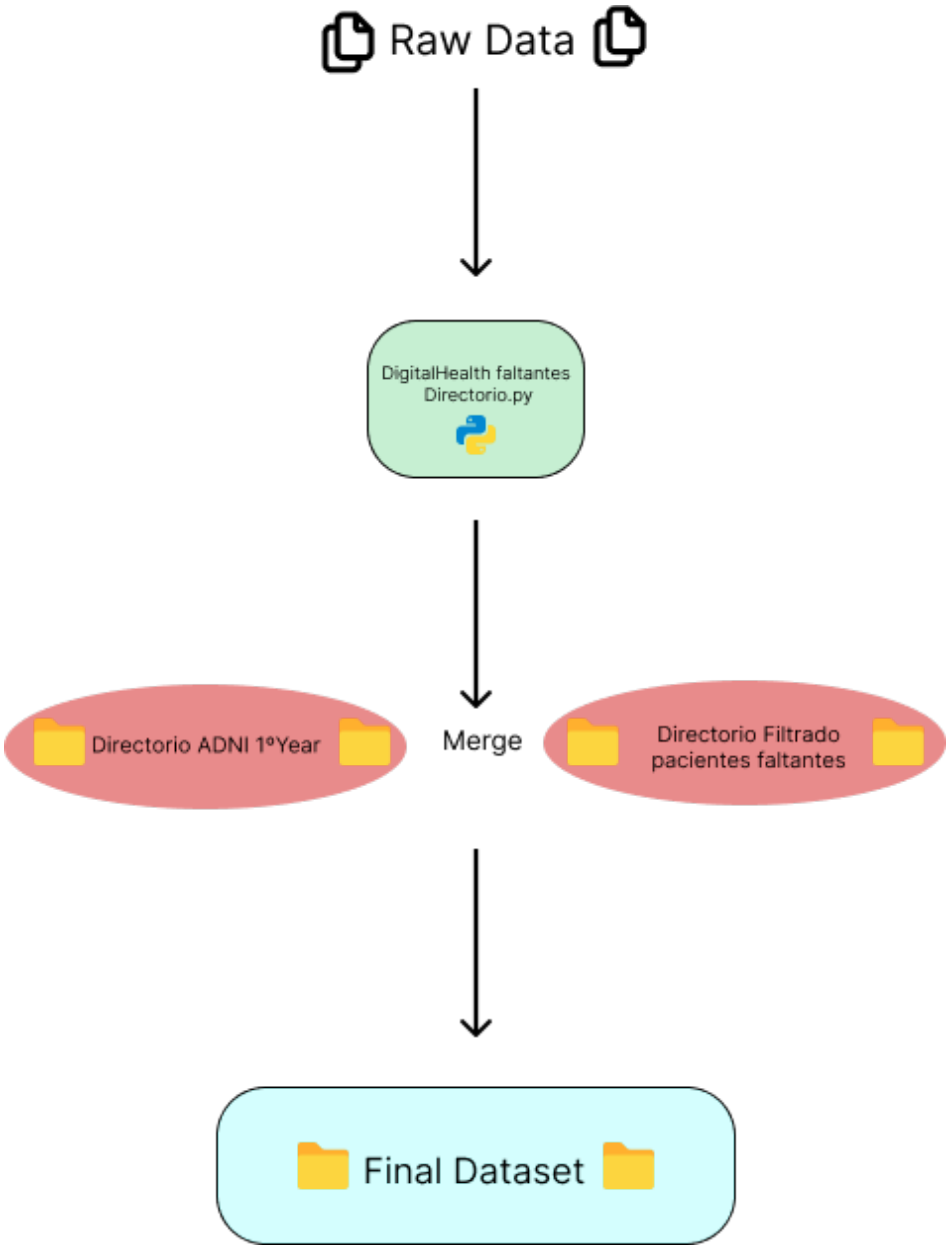


Figura 2.11: Esquema simplificado del proceso de filtro sobre los datos

Siguiendo la misma lógica de antes, se ideó e implementó un programa que buscase en los directorios de los pacientes restantes los .nii pertenecientes a los tipos de estudios correspon-

dientes.

```python
for sujeto in os.listdir(origen):
    path_sujeto = os.path.join(origen, sujeto)
    if not os.path.isdir(path_sujeto):
        continue

    for descripcion in os.listdir(path_sujeto):
        path_descripcion = os.path.join(path_sujeto, descripcion)
        if not os.path.isdir(path_descripcion):
            continue

        if descripcion in descripciones_validas:
            destino_path = os.path.join(destino, sujeto, descripcion)
            if not os.path.exists(destino_path):
                shutil.copytree(path_descripcion, destino_path)
                print(f"Copiado: {path_descripcion} -> {destino_path}")
```

Figura 2.12: Fragmento de código en Python que recorre subdirectorios con el objetivo de filtrar los datos en base a los tipos de estudio válidos

Este bloque de código mostrado en la figura 2.12 recorre recursivamente un directorio de origen que contiene subcarpetas por sujeto, y dentro de estas, subcarpetas con distintas descripciones de imágenes médicas. El objetivo es identificar y copiar únicamente aquellas subcarpetas cuyo nombre (que representa el tipo de estudio) coincida con una lista predefinida de tipos de estudio válidos. Estas carpetas seleccionadas se copian a un nuevo directorio de destino, conservando la estructura de carpetas por sujeto. Tras esto, tenemos una carpeta con todos los pacientes faltantes y sus respectivos .nii. Finalmente hacemos una fusión con el directorio ADNI1_Complete_1Yr_1.5T_4_26_2025 obteniendo así, el directorio Final Dataset, el cual será al que le pasaremos al programa principal, para extraer, clasificar y ordenar todos los .nii por visitas.

De este modo concluye el proceso de preprocesamiento de datos, dando como resultado un conjunto de datos final organizado por visitas, cada una de las cuales está estructurada por planos visuales (axial, coronal y sagital). La tabla 2.2 presenta un resumen detallado de las características del dataset generado.

Directorio	Visita	Plano	Diagnóstico	Nº Imágenes
Dataset	SC	Axial	AD	23198
			MCI	45126
			CN	26194
		Coronal	AD	23198
			MCI	45126
			CN	26194
		Sagital	AD	23198
			MCI	45126
			CN	26194
	M06	Axial	AD	20250
			MCI	40183
			CN	23883
		Coronal	AD	20250
			MCI	40183
			CN	23883
		Sagital	AD	20250
			MCI	40183
			CN	23883
	M12	Axial	AD	15402
			MCI	34003
			CN	19246
		Coronal	AD	15402
			MCI	34003
			CN	19246
		Sagital	AD	15402
			MCI	34003
			CN	19246

Tabla 2.2: Información del Dataset

Teniendo un total de 742.455 imágenes.

2.4. Arquitectura: Modelo DenseNet

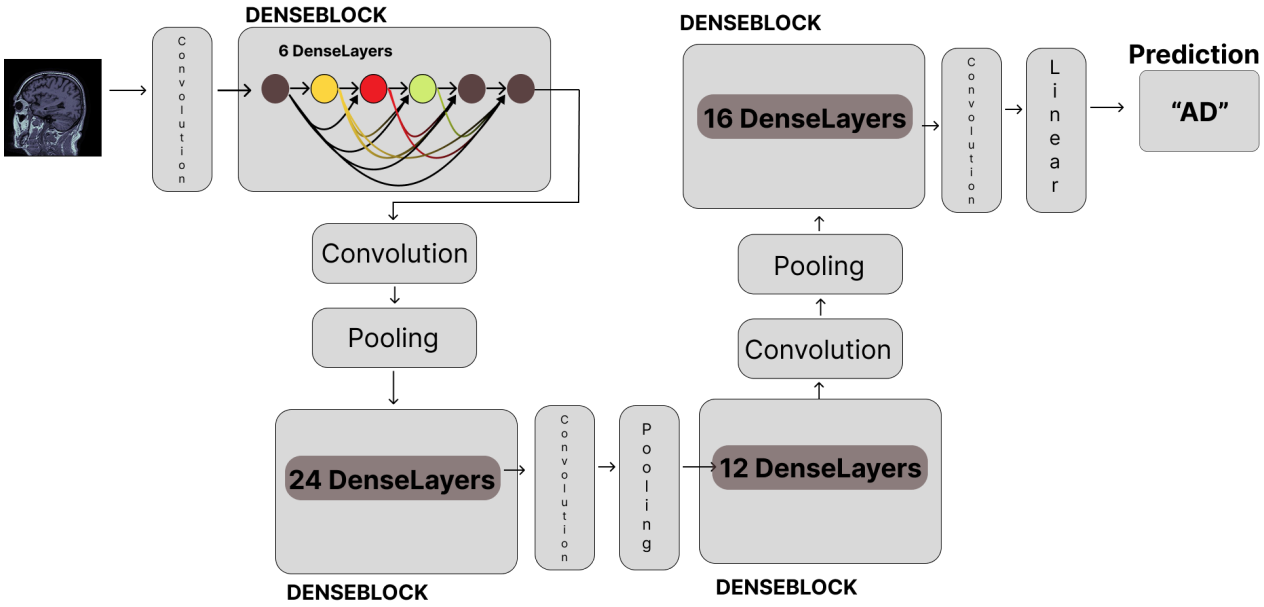


Figura 2.13: Esquema de la arquitectura

Para el desarrollo del proyecto hemos usado la arquitectura DenseNet formada por cuatro bloques densos de seis, doce, veinticuatro y dieciséis capas densas pudiéndolas apreciar en la figura 2.13. Esta arquitectura coincide con la presentada en el paper, DenseNet121 [18]

La primera capa de la arquitectura consiste en una capa de convolución 7x7 que captura los patrones grandes y reduce la resolución de la imagen. Las capas densas siguen la estructura por capas BN-ReLU-Conv. En primer lugar se aplica normalización por lotes para estabilizar y acelerar el entrenamiento, seguida de una función de activación no lineal ReLU. A continuación, aunque cada capa solo produce k mapas de características de salida, normalmente tiene muchos más mapas de características de entrada [18]. Por ello se añade una capa denominada bottleneck, que sirve para reducir el número de mapas de características de entrada y mejorar la eficiencia computacional [18]. Tras esto vuelve a pasar por las capas BN-ReLU-Conv, salvo que esta última es la convolución principal 3x3 que genera un número k(grow_rate) de nuevas características. La salida de esta capa da como resultado una serie de características nuevas y refinadas que se concatenan con todas las características previas, formando un conjunto nuevo de características que sirve como entrada para las siguientes capas densas.

Las capas que conectan los DenseBlocks se les llaman capas de transición, las cuales realizan operaciones de convolución y agrupamiento. Las capas de transición de nuestro proyecto están formadas por una capa de normalización por bloques y una capa de convolución 1x1, acompañadas de una capa de pooling de tamaño 2x2.

Al final de la arquitectura nos encontramos con una clasificación compuesta por una capa de convolución 1x1 y la capa lineal completamente conectada softmax, de 3 salidas (AD, MCI, CN).

Desde el punto de vista del software hemos implementado dicha arquitectura a través de una clase llamada DenseNet. Donde escalamos las imágenes para que tengan todas la misma dimensión de 224x224, les aplicamos un Median Filter, ya que en el estudio [27] se demuestra que supone una mejora en el modelo y finalmente las transformamos en un tensor. En cuanto al entrenamiento, se define la función de pérdida como Cross Entropy Loss, adecuada para tareas de clasificación multiclase y para el optimizador de los pesos, se utiliza el algoritmo Adam, configurado con la tasa de aprendizaje $0{,}55 \times 10^{-4}$.

2.5. Implementación de la solución computacional

De cara a la resolución de los objetivos se llevaron a cabo una serie de experimentos, todos ellos con diferentes configuraciones.

2.5.1. Elección de hiperparámetros

Durante el desarrollo del proyecto se hizo necesario hacer un pequeño estudio paramétrico de selección de parámetros, tales como el learning rate y el batch size. Para decidir los valores de éstos, se experimentó con varias configuraciones distintas de cara a encontrar aquella que se adaptase bien al problema ternario. Tras estas pruebas el learning rate óptimo fue de 1×10^{-5}, con el cual, la red presentó un desempeño relativamente mejor respecto al valor 1×10^{-4} figura 2.14, como hemos comentado un valor muy bajo supone un entrenamiento excesivamente lento, por lo que se decidió elegir el punto medio entre ambos valores, dando como resultado la selección final de 1×10^{-4}, mostrando un equilibrio entre eficiencia y velocidad de entrenamiento.

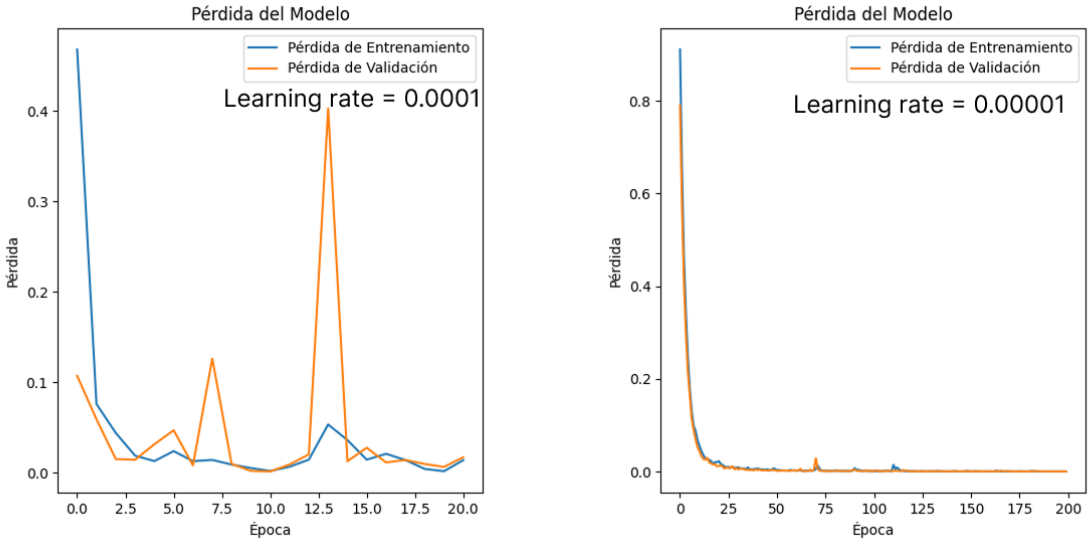


Figura 2.14: Métricas comparativas de los learning rates obtenidas el conjunto de datos de pruebas iniciales

Por otro lado, el batch se sometió bajo el mismo procedimiento, y el valor que reflejó mejores resultados fue el 64 tal y como podemos ver en la comparación de la figura 2.15

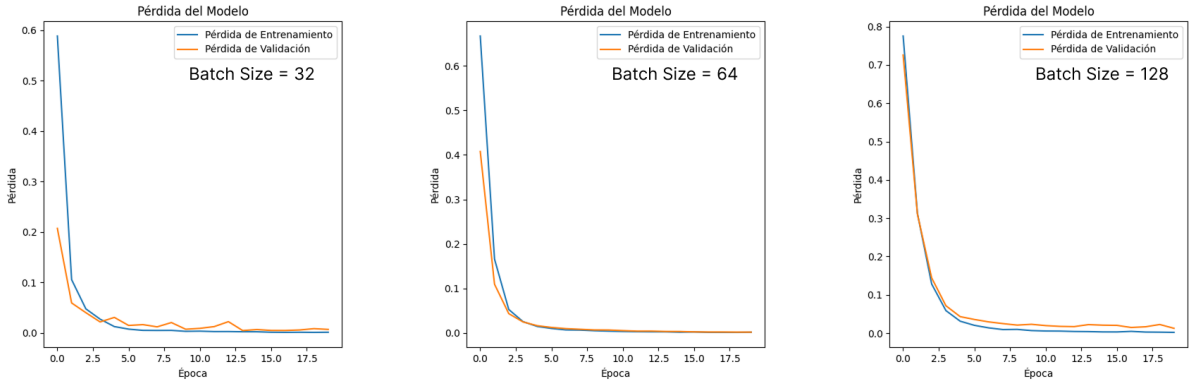


Figura 2.15: Métricas comparativas de los diferentes batch sizes obtenidas el conjunto de datos de pruebas iniciales

2.5.2. Elección del modelo base

Además del estudio de hiperparámetros, también realizó una serie de pruebas sobre qué arquitectura elegir para las pruebas significativas, dichas pruebas se llevaban a cabo para ver si una arquitectura preentrenada, como la DenseNet121, tenía mejor rendimiento que una DenseNet sin preentrenar, es decir, esta comparación tenía como objetivo determinar si el conocimiento previo aprendido por una red preentrenada aportaba ventajas significativas frente a un modelo entrenado desde cero en nuestro contexto específico. De este modo, se construyeron las dos arquitecturas con los mismos hiperparámetros y fuente de datos, con un kfold de dos y con cinco épocas. Esto concluyó de forma adicional en nuestro primer objetivo, ya que el resultado de este nos permitía decidir qué arquitectura se adaptaba mejor a nuestro problema, obteniéndose un modelo base desde el cual continuar con las siguientes fases del estudio.

2.5.3. Configuración del modelo

Para llevar a cabo los distintos experimentos de forma organizada y reproducible, se diseñó un script que automatiza todo el proceso de entrenamiento y evaluación del modelo. Este código, nos permite iterar fácilmente sobre diferentes fuentes de datos correspondientes a los tres planos anatómicos (axial, coronal y sagital), así como modificar parámetros críticos como el número de particiones del k-fold, la tasa de aprendizaje o el tamaño del batch. 2.16

```
724     for fold, (train_idx, test_idx) in enumerate(kfold.split(image_paths, labels)):
725         print(f"\n====== Fold {fold+1}/{k_folds} ======")
726 
727         train_paths = [image_paths[i] for i in train_idx]
728         train_labels = [labels[i] for i in train_idx]
729         test_paths = [image_paths[i] for i in test_idx]
730         test_labels = [labels[i] for i in test_idx]
731 
732         densenet_model = DenseNetModel(data_dir=data_dir, num_classes=3, num_epochs=5,
733                                        learning_rate=lr, batch_size=batch_size, train_paths = train_paths,
734                                        train_labels = train_labels, test_paths = test_paths, test_labels = test_labels)
735 
736         densenet_model.save_model(f"densenet_fold{fold+1}_Plano_{plano}.pth")
737 
738         densenet_model.plot_metrics(fold+1, plano, lr, batch_size)
739 
740         all_train_losses.append(densenet_model.train_losses)
741         all_val_losses.append(densenet_model.val_losses)
742         all_train_accuracies.append(densenet_model.train_accuracies)
743         all_val_accuracies.append(densenet_model.val_accuracies)
744 
745         fpr, tpr, roc_auc = densenet_model.plot_roc_curve(fold+1, plano, lr, batch_size)
```

Figura 2.16: Fragmento de la implementación de la configuración del modelo dinámica

La lógica del código se basa en una estructura iterativa que recorre cada ruta de los planos de imagen definidos previamente. Para cada uno de ellos, se ejecuta el proceso completo de entrenamiento utilizando validación cruzada estratificada (Stratified K-Fold), permitiendo conservar la proporción de clases en cada partición. Durante cada iteración, se almacenan automáticamente las métricas de la pérdida, la precisión y las curvas ROC (tanto de tipo 1 versus Rest como binaria), así como las matrices de confusión por fold. Asimismo, al finalizar

la iteración se guardan los resultados del entrenamiento, test y validación, para realizar la media de todos los folds y plasmarlos en las distintas métricas.

2.5.4. Validación Cruzada

Un modelo de aprendizaje automático es tan bueno como la calidad de los datos con los que se entrenó, por ende hemos de asegurarnos de que el modelo reciba y entrene con datos que nunca ha visto. Para afirmar esto, hacemos uso del método validación cruzada (Cross-Validation).

Nosotros usamos la variante de Stratified kfold., dado a que está variante se adapta al desbalance natural de los problemas médicos entre clases visible en la figura 2.2

A diferencia de la validación cruzada estándar, Stratified K-Fold garantiza que cada una de las particiones (folds) mantiene aproximadamente la misma proporción de muestras por clase que el conjunto de datos original. Según la documentación oficial: Esta es una variante del método k-fold que devuelve particiones estratificadas: cada conjunto contiene aproximadamente el mismo porcentaje de muestras de cada clase objetivo que el conjunto completo. [28] 2.17

Es por ello, que se evita que los conjuntos de entrenamiento o validación queden dominados por las clases mayoritarias (MCI en nuestro caso), lo cual podría sesgar los resultados y afectar negativamente a la generalización del modelo. Finalmente, se decidió optar por un split donde el 80 % del conjunto de datos de entrenamiento del fold, sería dedicado al propio entrenamiento, mientras que el 20 % de estos serían asignados al conjunto de validación.

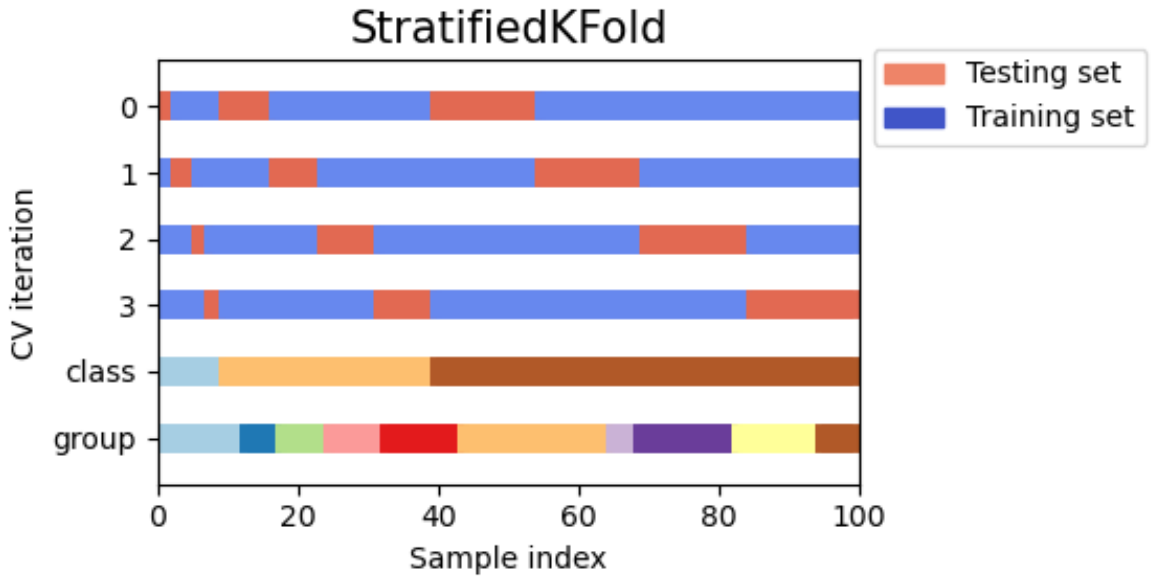


Figura 2.17: Esquema de trabajo del proceso de validación cruzada

Capítulo 3

Resultados y discusión

3.1. Diagnóstico ternario con datos screening

En este apartado se presentan los resultados del modelo de inteligencia artificial propuesto en este proyecto, el cual estará orientado al procesamiento de MRI para la clasificación ternaria de los estados cognitivos característicos de la enfermedad del Alzheimer,que como hemos destacado anteriormente son: enfermedad de Alzheimer (AD), deterioro cognitivo leve (MCI) y cognitivamente normal (CN).

Durante la construcción de este modelo se realizó un experimento adicional con el fin de seleccionar la arquitectura central del sistema. En este se comparó la arquitectura DenseNet121, el cual es un modelo preentrenado, frente a una arquitectura DenseNet121 sin preentrenamiento, a la cual nos referiremos como DenseNet, para ver si la inicialización de pesos de forma aleatoria o no, presentaba mejor rendimiento en nuestro problema. Este experimento culminó en la elección de la arquitectura DenseNet121 como modelo central de nuestro sistema.

A continuación, se presentan los resultados detallados, incluyendo los perfiles de las métricas de precisión y pérdida durante el entrenamiento, las curvas ROC bajo los enfoques 1 versus Rest y binario, así como las matrices de confusión. Estos elementos permiten analizar de manera integral tanto el comportamiento de los modelos comparados como el desempeño final del sistema completo.

3.1.1. Comparativa entre DenseNet121 y DenseNet sin preentrenar

En esta sección, haremos un análisis detallado de los resultados obtenidos de los dos modelos DenseNet121 y DenseNet. Este análisis comparativo es realizado desde los distintos planos de entrada (Axial, Coronal y Sagital) y los distintos estados cognitivos, a lo largo de este capítulo nos referiremos a estos como clases. Asimismo, dado que los resultados obtenidos

para los distintos planos son similares entre sí y conducen a las mismas conclusiones, en esta sección se presentarán únicamente los resultados de las métricas correspondientes al plano Axial. Es por esto que se podrán acceder a las demás en el Anexo (Repositorio GitHub).

Para este experimento se utilizó como fuente de entrada del sistema el conjunto de datos correspondientes a la visita SC. Este conjunto cuenta con 283,554 imágenes en total y este se divide:

- ✔ AD: Esta clase contiene 69,594 imágenes (23,198 por plano)
- ✔ MCI: Esta clase 135,378 imágenes (45,126 por plano).
- ✔ CN: Esta clase 78,582 imágenes (26,194 por plano).

Donde:

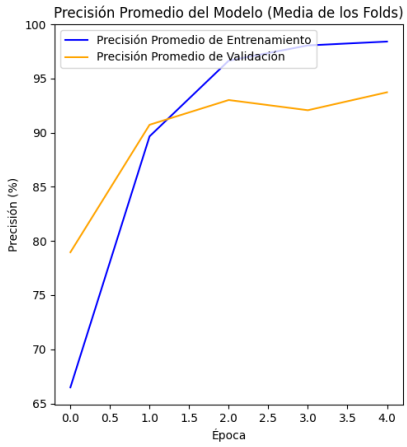
- ✔ 37807 imágenes pertenecen al conjunto de entrenamiento.
- ✔ 9452 imágenes pertenecen al conjunto de validación.
- ✔ 47259 imágenes pertenecen al conjunto de prueba.

Para este número de imágenes utilizamos un kfold de 2, donde dentro del conjunto de entrenamiento empleamos un 80 % para el entrenamiento y un 20 % para la validación. Asimismo los modelos se configuraron siguiendo los parámetros que se pueden apreciar en la tabla 3.1

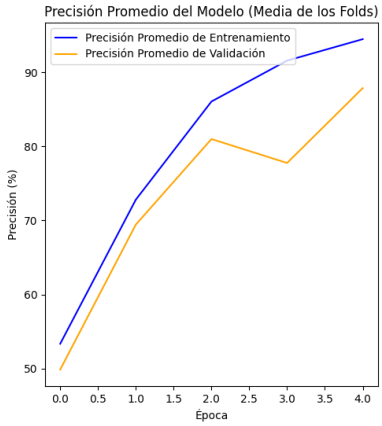
Modelo	**Tasa de Aprendizaje**	**Optimizador**	**Tamaño de Lote**	**Fuente de Entrenamiento**	**K-Fold**	**Preentrenada**
DenseNet121	1 x 10^{-4}	Adam	64	Screening	2	Sí
DenseNet	1 x 10^{-4}	Adam	64	Screening	2	No

Tabla 3.1: Tabla de Modelos y sus parámetros de entrenamiento

Perfiles de las métricas precisión y pérdida durante entrenamiento

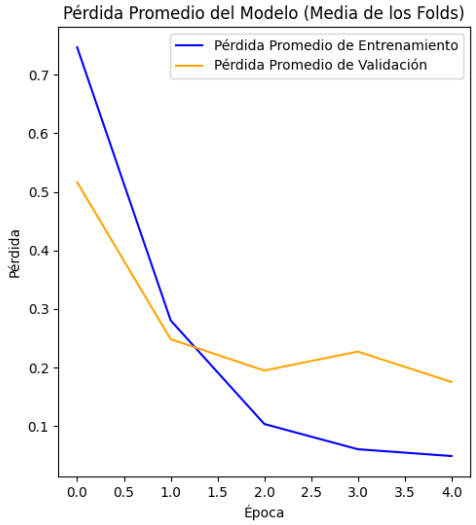


(a) Densenet121

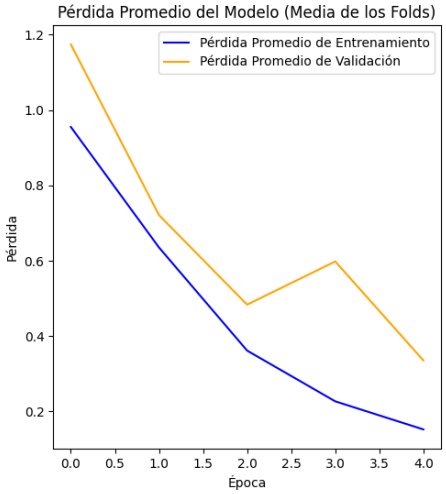


(b) DenseNet sin preentrenar

Figura 3.1: Evolución de la precisión promedio (entrenamiento y validación) de los modelos DenseNet121 y DenseNet a lo largo de 5 épocas. (a) Densenet121 y (b) DeseNet sin preentrenar.



(a) Densenet121



(b) DenseNet sin preentrenar

Figura 3.2: Evolución de la pérdida promedio (entrenamiento y validación) de los modelos DenseNet121 y DenseNet a lo largo de 5 épocas. (a) Densenet121 y (b) DenseNet sin preentrenar.

Las curvas presentadas en la figura 3.1 muestran las curvas de precisión promedio de

los modelos con y sin preentrenamiento, evaluadas como la media de los folds durante el entrenamiento.

Podemos observar que el DenseNet121 3.1a muestra un comportamiento típico de sobreajuste a partir de la época 1. Por otro lado, el modelo sin preentrenar 3.1b presenta una curva de precisión y validación donde no se muestran síntomas de sobreajuste significativo, por lo que a diferencia de Densenet121 no podemos concluir que exista sobreajuste.

En segundo lugar, la figura 3.2 se muestran las curvas de pérdida promedio para los modelos con y sin preentrenamiento, evaluadas como la media de los folds durante el entrenamiento.

Podemos observar que el modelo DenseNet121 preentrenado 3.2apresenta una curva de pérdida de validación que a partir de la época 2 diverge respecto a la de entrenamiento. Esto indica, al igual que la métrica de precisión, que el modelo presenta sobreajuste, lo cual indica que el modelo se ajusta bien a los datos de entrenamiento, pero no logra generalizar igual de bien en la validación.

En contraste, el modelo sin preentrenar 3.2b presenta una pérdida de validación que también disminuye inicialmente, pero luego muestra una mayor oscilación. Este comportamiento sugiere una menor estabilidad y posiblemente una mayor tendencia al sobreajuste.

Por lo que al final del entrenamiento ambos modelos alcanzan porcentajes de precisión levemente iguales, de modo que a partir de estas gráficas, sin tener en cuenta el resto de resultados, se puede concluir que el modelo DenseNet sin preentrenamiento presenta un mejor comportamiento, ya que no evidencia sobreajuste, a diferencia del modelo DenseNet121.

Curvas ROC y AUC

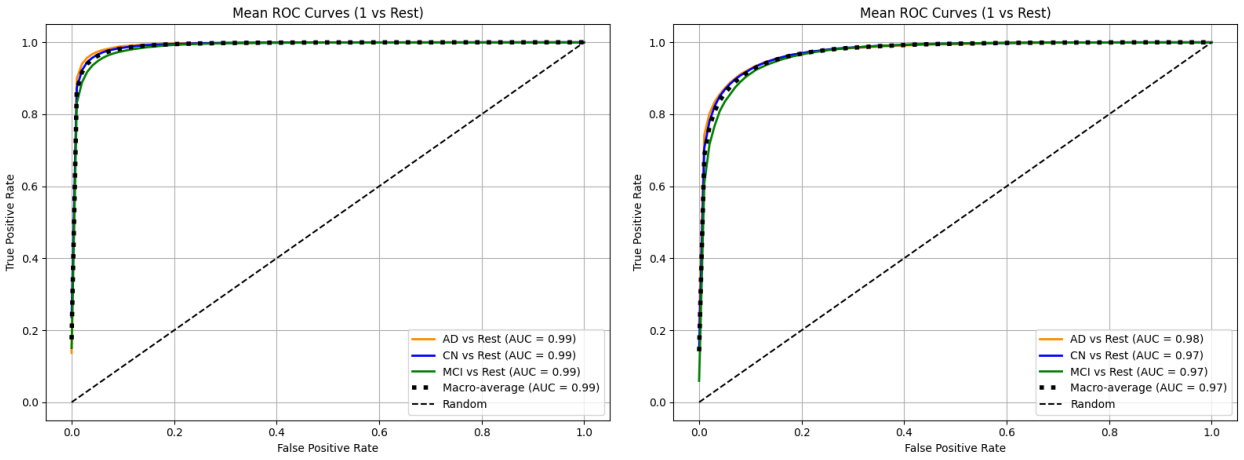
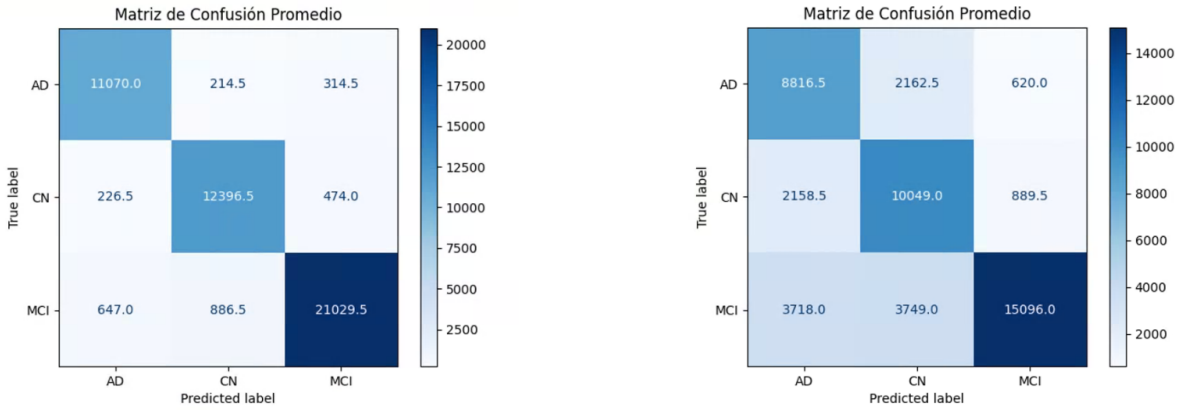


Figura 3.3: Comparación de las curvas ROC de los modelos DenseNet121 y DenseNet bajo la estrategia 1 vs Rest para ambas arquitecturas: a la izquierda, DenseNet121; a la derecha, DenseNet sin preentrenamiento.

En primer lugar, comentaremos los resultados de la métrica 1 versus Rest 3.3 . Como podemos ver DenseNet121 presenta resultados más estables en las tres clases AD versus Rest, CN versus Rest y MCI versus Rest, todos ellos con un valor de 0.99. Por otro lado, DenseNet presenta valores ligeramente peores, donde el AD versus Rest tiene un mejor rendimiento con un AD versus Rest de 0.98, mientras que las otras clases descienden a 0.97. Por lo que podemos concluir que, DenseNet121 presenta mejores resultados que el DenseNet.

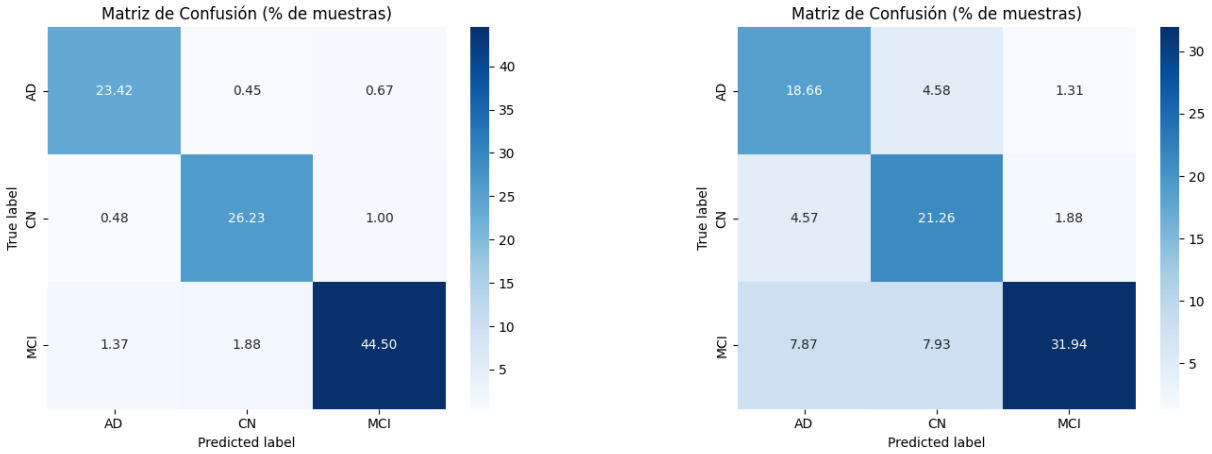
A continuación, los resultados de las métricas binarias accesibles a través de nuestro Anexo (Repositorio GitHub) presentan un comportamiento similar al anterior, específicamente las ADvsCN ADvsMCI y CNvsMCI muestran un 1.0, 0.99 y 0.99 respectivamente en el modelo DenseNet121, frente a los descendientes 0.97, 0.97 y 0.98 de AUC del modelo DenseNet

Matrices de Confusión



(a) Densenet121 (b) DenseNet sin preentrenar

Figura 3.4: Matrices de confusión promedio obtenidas del conjunto de test de los modelos DenseNet121 y DenseNet. (a) Densenet121 y (b) DenseNet sin preentrenar



(a) Densenet121 (b) DenseNet sin preentrenar

Figura 3.5: Matrices de confusión promedio por porcentajes obtenidas del conjunto de test de los modelos DenseNet121 y DenseNet. (a) Densenet121 y (b) DenseNet sin preentrenar

Los resultados obtenidos en la clasificación, los cuales están representados por la correspondientes matrices de confusión 3.4 nos muestra el desempeño de nuestros modelos en la resolución del problema.

Si observamos las diagonales de las matrices podemos percibir que el modelo DenseNet121 presenta mayor número de aciertos, que el modelo sin preentrenar, mostrando 2107.5 aciertos para la clase AD ,2532.5 aciertos para la clase CN y 4219.5 aciertos en la clase MCI, frente a los 1787.5 aciertos para la clase de AD, 2296.5 aciertos para la clase CN y 4220.5 aciertos en la clase MCI del modelo sin preentrenar.

Todo esto lo podemos reafirmar, viendo la figura 3.5 que refleja el gran porcentaje de muestras bien clasificadas.

Evaluación global de rendimiento

En función de los resultados obtenidos tras los experimentos, calculamos el desempeño en base a un conjunto de métricas ampliadas con el fin de evaluar la efectividad de los modelos entrenados las cuales son: precisión global (Accu), sensibilidad (Sens), especificidad (Spec), área bajo la curva ROC (AUC), y los índices compuestos CUI+ y CUI-. Estos indicadores permiten una evaluación integral del desempeño del modelo, considerando tanto la capacidad para detectar correctamente los casos positivos como para evitar falsos positivos, proporcionando así una visión completa de la fiabilidad y precisión del sistema desarrollado. Los resultados de estos indicadores podemos observarlos en la tabla 3.2

Modelo	Plano	Clase	Accu	Sens	Spec	PPV	NPV	CUI+	CUI-	AUC
DenseNet121	Axial	ADvsRest	0.9703	0.9544	0.9755	0.9269	0.9850	0.8846	0.9609	0.99
		CNvsRest	0.9619	0.9465	0.9678	0.9184	0.9793	0.8693	0.9477	0.99
		MCIvsRest	0.9509	0.9320	0.9681	0.9639	0.9397	0.8984	0.9097	0.99
	Coronal	ADvsRest	0.9756	0.9387	0.9876	0.9609	0.9802	0.9020	0.9680	**1**
		CNvsRest	0.9654	0.9214	0.9823	0.9523	0.9702	0.8774	0.9531	0.99
		MCIvsRest	0.9555	0.9687	0.9434	0.9398	0.9706	0.9104	0.9156	0.99
	Sagital	ADvsRest	**0.9910**	**0.9778**	**0.9953**	**0.9855**	**0.9928**	**0.9637**	**0.9882**	**1**
		CNvsRest	**0.9851**	**0.9811**	**0.9867**	**0.9658**	**0.9927**	**0.9476**	**0.9795**	**1**
		MCIvsRest	**0.9832**	**0.9798**	**0.9863**	**0.9849**	**0.9816**	**0.9650**	**0.9682**	**1**
Sin preentrenar	Axial	ADvsRest	0.8168	0.7601	0.8352	0.6000	0.9146	0.4561	0.7638	0.99
		CNvsRest	0.8104	0.7673	0.8270	0.6296	0.9026	0.4831	0.7464	0.99
		MCIvsRest	0.8101	0.6691	0.9389	0.9091	0.7564	0.6082	0.7102	0.99
	Coronal	ADvsRest	0.9319	0.9162	0.9369	0.8253	0.9717	0.7562	0.9105	**1**
		CNvsRest	0.9355	0.8732	0.9594	0.8917	0.9518	0.7786	0.9131	0.99
		MCIvsRest	0.9079	0.8812	0.9322	0.9223	0.8957	0.8128	0.8350	0.99
	Sagital	ADvsRest	0.9487	0.9541	0.9470	0.8541	0.9845	0.8149	0.9323	**1**
		CNvsRest	0.9570	0.9147	0.9732	0.9289	0.9675	0.8497	0.9415	**1**
		MCIvsRest	0.9344	0.9056	0.9606	0.9546	0.9176	0.8645	0.8815	**1**

Tabla 3.2: Comparación detallada de métricas de rendimiento para DenseNet121 y el modelo sin preentrenamiento, segmentado por plano y clase.

Tal y como hemos comentado, podemos observar que el DenseNet121 presenta un mejor rendimiento global frente a la arquitectura DenseNet. Esta mejora persiste en los tres planos analizados y en nuestras tres clases evaluadas.

Los mejores resultados obtenidos por DenseNet121 destacan especialmente en el plano sagital, donde se alcanzan valores máximos para cada clase. En particular, para la clase AD, se logra una sensibilidad de 0.9811, una NPV de 0.9953 y un CUI+ de 0.9637. Para la clase CN, se obtiene una especificidad de 0.9867, un PPV de 0.9658 y un CUI+ de 0.9476. Finalmente, para la clase MCI, el modelo alcanza un CUI+ de 0.9650 y un CUI- de 0.9682, consolidando un rendimiento robusto en la detección de todos los grupos.

Por otro lado, el modelo sin preentrenar presenta un rendimiento más irregular. Aunque en algunos casos alcanza valores competitivos como la sensibilidad de 0.9541 para la clase AD en el plano axial, la especificidad de 0.9732 de la misma clase o PPV de 0.9546 para la clase MCI en el plano sagital, estos se presentan de forma aislada y no se mantienen de manera consistente en todas las métricas ni clases. Asimismo muestra un bajo desempeño en la clase AD del plano axial, con un CUI+ de apenas 0.4561, lo que evidencia una capacidad reducida para detectar correctamente los casos positivos.

Cabe a destacar, que la figura 3.4 y 3.5 puede llevar a confusión dado a los tonos de color de la gráfica, a pesar de que el tono azul oscuro se interpreta como que el modelo ha predecido muy favorablemente la clase MCI, realmente si nos fijamos en 3.2 podemos observar que los valores de acierto no son tan dispares como indica el tono de las métricas, esto se debe al desbalanceamiento que existe entre las clases, algo muy normal en estudios médicos. No obstante, esto está compensado gracias al uso de Stratified Kfold comentado en la sección 2.5.4

Por lo tanto, se ha optado por seleccionar la arquitectura DenseNet121 como base del sistema, a pesar de la tendencia al sobreajuste observada en sus primeras etapas de entrenamiento,entendiendo este como la capacidad del modelo para memorizar los datos de

entrenamiento en lugar de generalizar. Sin embargo, resulta particularmente relevante que, a pesar de esta tendencia al sobreajuste, el modelo demuestra un rendimiento superior sobre los datos del conjunto de prueba, los cuales no fueron utilizados durante el entrenamiento. Este hecho se refleja en los valores obtenidos de AUC y en las curvas ROC, donde DenseNet121 supera al modelo DenseNet.

Es decir, aunque el modelo muestra signos de memorizar los datos de entrenamiento, su comportamiento sobre datos no vistos sugiere que dicho suceso no compromete su capacidad de generalización, sino que, por el contrario, logra predecir de forma acertada ejemplos completamente nuevos. Además, en las siguientes etapas del TFG se empleó un esquema de validación cruzada estratificada con K = 5, lo que aporta mayor robustez y consistencia a los resultados, dado que un mayor valor de K tiende a ofrecer estimaciones más fiables del desempeño del modelo, mitigando los efectos del sobreajuste.

3.1.2. Resultados del sistema inteligente

Una vez concluida la anterior fase, se produjo un nuevo experimento partiendo de la base establecida tras los resultados anteriores. Asimismo, a diferencia del anterior, en esta sección se presentarán los resultados en los distintos planos(Axial, Coronal y Sagital).

Para este experimento se utilizó de nuevo como fuente de entrada del sistema el conjunto de datos correspondientes a la visita SC. Sin embargo, para este experimento se utilizó, como hemos mencionado anteriormente, un esquema de validación cruzada estratificada con K = 5, donde dentro del conjunto de entrenamiento empleamos un 80 % para el entrenamiento y un 20 % para la validación. Como K = 5 los conjuntos se dividen de esta forma:

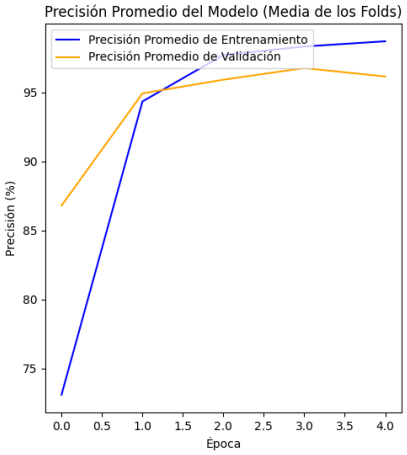
- ✔ 60,491 imágenes pertenecen al conjunto de entrenamiento.
- ✔ 15,123 imágenes pertenecen al conjunto de validación.
- ✔ 18,904 imágenes pertenecen al conjunto de prueba.

Asimismo el modelo DenseNet121 siguió los parámetros que podemos ver en la tabla 3.3

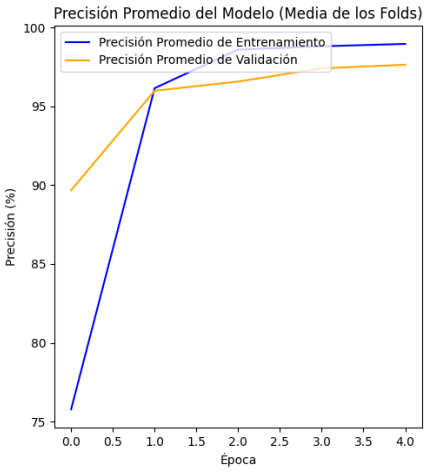
Modelo	**Tasa de Aprendizaje**	**Optimizador**	**Tamaño de Lote**	**Fuente de Entrenamiento**	**K-Fold**	**Preentrenada**
DenseNet121	1 x 10^{-4}	Adam	64	Screening	5	Sí

Tabla 3.3: Tabla del DenseNet121 y sus parámetros de entrenamiento

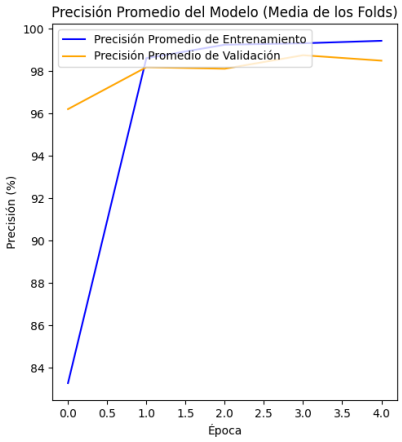
Perfiles de las métricas precisión y pérdida durante el entrenamiento



(a) Plano Axial



(b) Plano Coronal



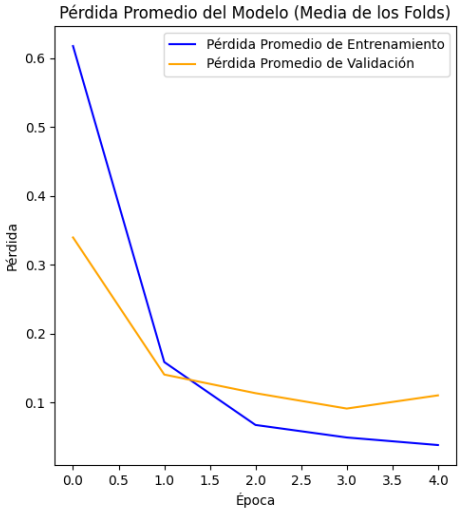
(c) Plano Sagital

Figura 3.6: Evolución de la precisión promedio (entrenamiento y validación) a lo largo de 5 épocas por plano del modelo DenseNet121 K=5. (a) Axial,(b) Coronal y (c) Sagital.

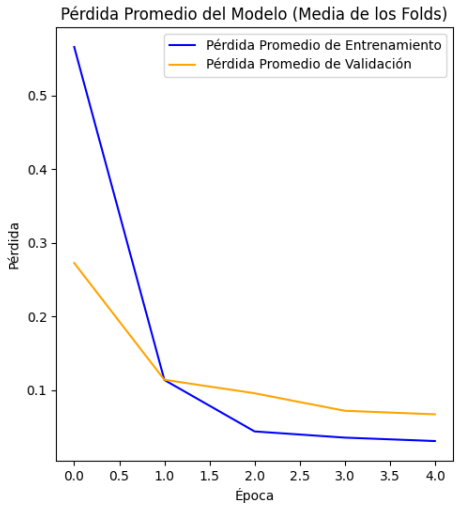
Como podemos observar en las curvas presentadas en la figura 3.6 podemos ver las curvas de precisión promedio del modelo DenseNet121 en sus distintos planos, evaluadas como la media de los cinco folds durante el entrenamiento.

Pudiendo observar que todas comparten un comportamiento similar, presentando una curva de precisión estable a lo largo de las épocas, seguida de cerca por la curva de la validación, apreciando claramente la ausencia de síntomas de sobreajuste significativo, llegando todas ellas a alcanzar valores de precisión en el entrenamiento y en la validación altos al final de la etapa de entrenamiento.

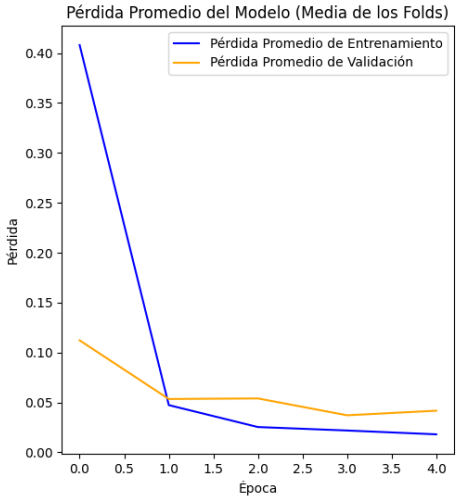
Denotando así la buena adaptación de la arquitectura a la hora de clasificar los distintos planos.



(a) Plano Axial



(b) Plano Coronal



(c) Plano Sagital

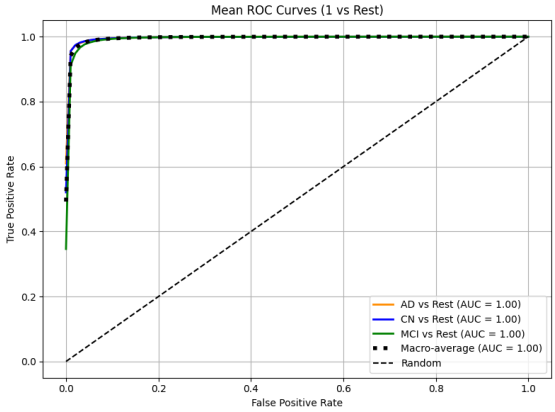
Figura 3.7: Evolución de la pérdida promedio (entrenamiento y validación) a lo largo de 5 épocas por plano del modelo DenseNet121 K=5. (a) Axial,(b) Coronal y (c) Sagital.

A continuación, la figura 3.7 se muestran las curvas de pérdida promedio del modelo DenseNet121 en sus distintos planos, evaluadas como la media de los cinco folds durante el entrenamiento. Al igual que en las curvas de precisión el modelo muestra un comportamiento similar en las curvas asociadas a los diferentes planos, mostrando una curva de pérdida que disminuye gradualmente acompañada de la curva de validación, estabilizándose entorno a las últimas épocas, al igual que las anteriores, esta no denota ningún síntoma de sobreajuste.

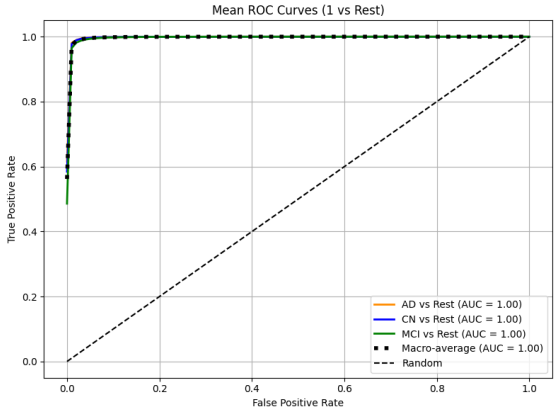
Demostrando de nuevo, que el modelo generaliza bien y se adapta bien a los diferentes

datos de los ejes de nuestro problema.

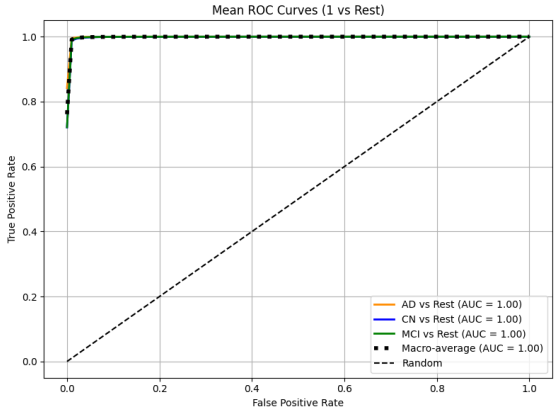
Curvas ROC y AUC



(a) Plano Axial



(b) Plano Coronal



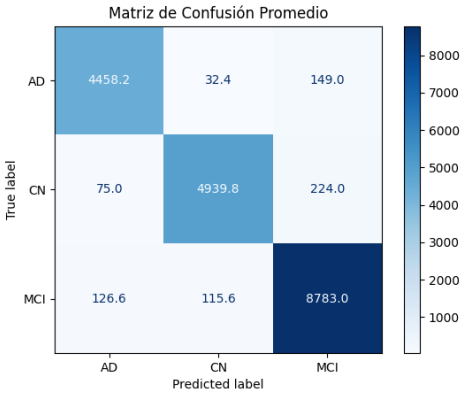
(c) Plano Sagital

Figura 3.8: Resultados de las curvas ROC obtenidas por el modelo DenseNet121 (K=5) bajo la estrategia 1 vs Rest de los tres planos. (a) Axial,(b) Coronal y (c) Sagital.

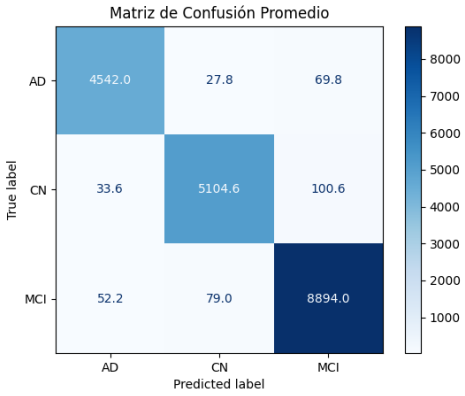
Como apreciamos en la figura 3.8 las tres figuras correspondientes a los planos Axial 3.8a , Coronal 3.8b y Sagital 3.8c , evaluadas como la media de los cinco folds durante el entrenamiento, comparten los mismos resultados, para cada uno de los ROC, en todas las combinaciones AD versus REST,CN versus REST Y MCI versus REST el valor del AUC es de 1.00. Esto indica que el modelo está clasificando perfectamente las diferentes clases de las que se compone el problema indiferentemente del plano.

Este resultado también se aprecia durante los resultados de la estrategia binaria (AD versus CN , AD versus MCI y CN versus MCI) accesibles a través de nuestro Anexo (Repositorio GitHub), denotando todas ellas, en todos los planos de un AUC total de 1. Esto hace hincapié en lo comentado anteriormente.

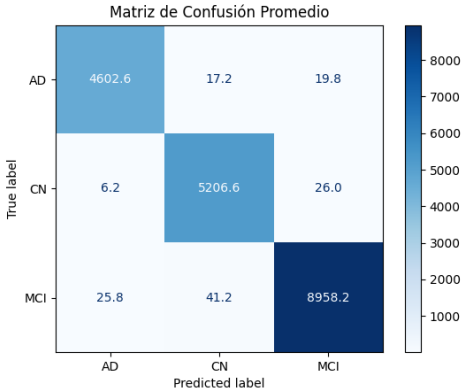
Matrices de Confusión



(a) Plano Axial

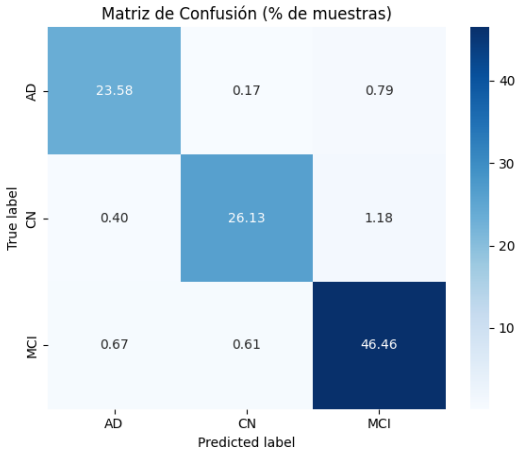


(b) Plano Coronal

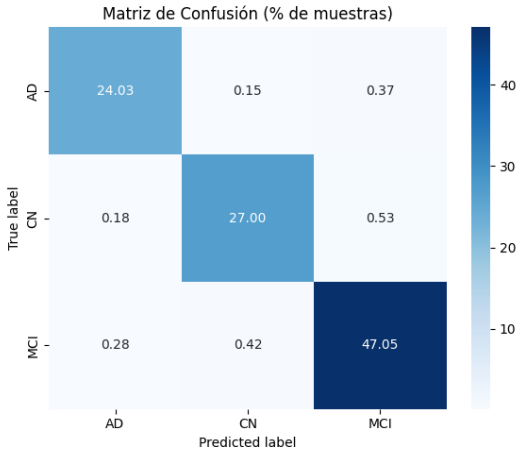


(c) Plano Sagital

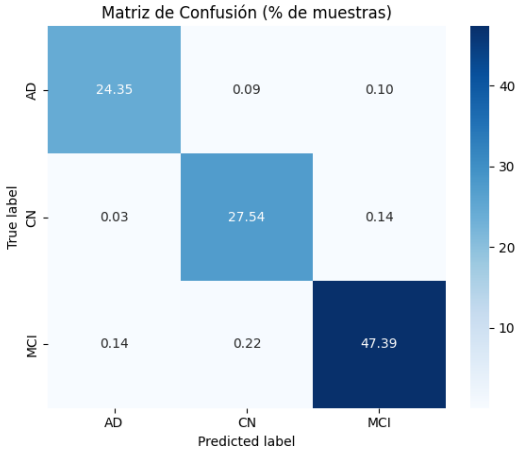
Figura 3.9: Matrices de confusión promedio obtenidas del conjunto de test para los tres planos del modelo DenseNet121 (K=5). (a) Axial,(b) Coronal y (c) Sagital.



(a) Plano Axial



(b) Plano Coronal



(c) Plano Sagital

Figura 3.10: Matrices de confusión promedio por porcentajes obtenidas del conjunto de test para los tres planos del modelo DenseNet121 (K=5) . (a) Axial,(b) Coronal y (c) Sagital.

En esta figura 3.9 podemos apreciar las distintas matrices de confusión para los tres ejes. Estos resultados destacan el buen desempeño del modelo, dado a que la mayoría de muestras se concentran en la diagonal. No obstante, no podemos discernir cual de los tres planos se desarrolla con exactitud mejor. Para ello, calculamos el promedio de % de las muestras distribuidas en las matrices de confusión.

En la figura 3.10 podemos apreciar dichos porcentajes, mostrándonos que a pesar de ser muy parecidas, hay planos que atinan más. En términos de rendimiento, el plano Sagital se posiciona como el más efectivo. Al sumar los porcentajes correspondientes a la diagonal principal de la matriz de confusión(que representan las predicciones correctas) se obtiene un total de 98.49 % (47.04 % para la clase AD, 27.51 % para MCI y 23.94 % para CN).

En segundo lugar se encuentra el plano Coronal, con un total de aciertos del 97.64 %, correspondiente a 46.41 % (AD), 27.28 % (MCI) y 23.95 % (CN).

Finalmente, el plano Axial presenta el menor rendimiento, con una tasa de clasificación correcta del 96.13 %, resultado de 45.46 % (AD), 27.12 % (MCI) y 23.55 % (CN).

Estos resultados reflejan una tendencia clara en la que el plano Sagital proporciona una mayor capacidad discriminativa entre clases para nuestro problema, seguido del coronal y, por último, el axial.

Evaluación global de rendimiento

Modelo	Plano	Clase	Accu	Sens	Spec	PPV	NPV	CUI+	CUI-	AUC
DenseNet121 k=5	Axial	ADvsRest	0.9797	0.9609	0.9859	0.9567	0.9873	0.9193	0.9733	1
		CNvsRest	0.9764	0.9429	0.9892	0.9709	0.9784	0.9155	0.9678	1
		MCIvsRest	0.9675	0.9732	0.9622	0.9593	0.9752	0.9335	0.9383	1
	Coronal	ADvsRest	0.9903	0.9790	0.9940	0.9815	0.9932	0.9608	0.9872	1
		CNvsRest	0.9873	0.9744	0.9922	0.9795	0.9902	0.9544	0.9825	1
		MCIvsRest	0.9840	0.9855	0.9828	0.9812	0.9867	0.9669	0.9696	1
	Sagital	ADvsRest	**0.9963**	**0.9920**	**0.9978**	**0.9931**	**0.9974**	**0.9852**	**0.9952**	1
		CNvsRest	**0.9952**	**0.9939**	**0.9957**	**0.9889**	**0.9976**	**0.9828**	**0.9934**	1
		MCIvsRest	**0.9940**	**0.9926**	**0.9954**	**0.9949**	**0.9932**	**0.9875**	**0.9886**	1

Tabla 3.4: Métricas de rendimiento del modelo DenseNet121 (k=5) según plano y clase.

En la tabla 3.4 podemos ver plasmados las diferentes métricas de rendimiento. En primer lugar, discutiremos cual de los tres planos en función de los resultados ha tenido un mejor desempeño, y posteriormente comentaremos los resultados finales de nuestro experimento.

En cuanto al análisis por planos, el modelo DenseNet121 (k=5) mostró un rendimiento notablemente superior en el plano Sagital, obteniendo las medias más altas de forma constante en todas las métricas evaluadas. Para ser exactos, alcanzó la precisión (Accu) más alta para todas las clases del 99.63 % en AD, 0.9952 en CN y 0.9940 en MCI, Asimismo, registró la mayor sensibilidad, con un 99.39 %, y la especificidad más alta, con un 99.78 %.

Este rendimiento sugiere que las representaciones en el plano sagital contienen información diagnóstica más discriminativa para la clasificación entre AD, CN y MCI. En segundo lugar, se posiciona el plano coronal, que contiene algunos de los mejores valores, como una especificidad del 99.40 % para el AD. Aunque los tres planos presentan resultados elevados, el plano sagital destaca particularmente por su equilibrio entre sensibilidad y especificidad, así como por sus valores más altos en los índices compuestos CUI+ y CUI-, reflejando una mayor capacidad para detectar correctamente los casos positivos.

3.2. Comparativa entre nuestro modelo y MyGNG

En este apartado, se presenta los resultados de la comparativa entre MyGNG y el modelo desarrollado en este proyecto DenseNet. MyGNG es un modelo ternario presentado y desarrollado en la investigación [25] que combina un sistema de agrupamiento llamado Growing Neural Gas con un módulo de etiquetado sencillo [25]. Este modelo destaca por el uso de datos como tests neuropsicológicos y características demográficas, para detectar de forma rápida y precisa la enfermedad de Alzheimer en etapas tempranas[25]. Por otro lado, nuestro

modelo se diferencia del anterior por el uso de las imágenes extraídas de las MRI como fuente de entrada, abordando así el problema desde una perspectiva basada en datos visuales. Cabe destacar que ambos modelos fueron entrenados y evaluados utilizando los mismos 819 pacientes, lo que permite una comparación directa y controlada de su rendimiento.

	Métricas de rendimiento					
	Accu	**Sens**	**Spec**	**AUC**	**CUI+**	**CUI-**
MyGNG	0.863	0.9018	0.8036	0.8576	0.7442	0.7305
DenseNet	0,9963	0,9939	0,9978	1	0,9875	0,9952

Tabla 3.5: Comparativa del rendimiento de DenseNet frente a MyGNG

En la figura 3.5 podemos observar las diferentes métricas de rendimiento entre el modelo de MyGNG y el DenseNet121, cuyas métricas vienen dadas por los mejores valores del rendimiento del modelo en todos sus planos. Al comparar el rendimiento de los modelos DenseNet y MyGNG, se observa una mejora significativa en todos los indicadores evaluados a favor de DenseNet. Individualmente,respecto a la precisión DenseNet alcanza un valor de 0.9963, lo que representa un incremento del 13.3 % respecto al modelo de MyGNG. Esta diferencia se traduce en una mayor proporción de clasificaciones correctas globales.

Asimismo, la sensibilidad mejora en 9.21 %, lo que indica que DenseNet es más eficaz a la hora de detectar correctamente los casos positivos.Por otro lado, este comportamiento se complementa con un notable incremento del 19.42 % en la especificidad, lo cual sugiere que también se reduce el número de falsos positivos de forma considerable.

En cuanto al AUC, DenseNet alcanza un valor de 1, frente al 0.8576 del modelo comparativo, lo que supone una mejora del 14.25 % en la capacidad del modelo para discriminar entre clases. Por otro lado, los índices de CUI+ y CUI- muestran mejoras aún más destacadas: 24.33 % y 26.47 % respectivamente, lo cual refuerza la superioridad de DenseNet tanto en la clasificación correcta de los positivos como de los negativos.

Es por ello que estos resultados que estos resultados reflejan que el modelo DenseNet no solo es más preciso, sino también más robusto y equilibrado en su capacidad de clasificación respecto a MyGNG.

3.3. Evaluación del seguimiento clínico en el rendimiento del modelo

En este apartado se presentan los resultados de los modelos que incorporan el historial clínico como parte de los datos de entrada. Se han considerado dos enfoques: en primer lugar, un enfoque parcial, que incluye combinaciones de visitas (por ejemplo, screening y m06), y en segundo lugar, un enfoque completo, que utiliza el historial completo de visitas (screening, m06 y m12).

Todos los experimentos se realizaron utilizando la cohorte previamente mencionada de 819 pacientes, aplicando una validación cruzada k-fold con k = 5. Para cada configuración de historial clínico, se llevó a cabo una comparación con un modelo equivalente que no incorpora dicho historial, es decir, modelos entrenados únicamente con la información de una única visita (por ejemplo, solo m06), frente a modelos entrenados con múltiples visitas (por ejemplo, screening + m06). Esta misma lógica se aplica a la marca temporal m12.

Finalmente, se comparan los resultados obtenidos para destacar y reafirmar la importancia del historial clínico en la mejora del rendimiento del modelo. Cabe a destacar que al igual que en la comparación anterior, dado que los resultados obtenidos para los distintos planos son similares entre sí y conducen a las mismas conclusiones, en esta sección se presentarán únicamente los resultados de las métricas correspondientes al plano Axial. Es por esto que se podrán acceder a las demás en el Anexo (Repositorio GitHub).

3.3.1. Diagnóstico m06

Para este experimento queremos hacer un diagnóstico en la marca temporal m06. Este experimento sigue la siguiente fórmula

$$D(m06) = f(sc, m06)$$

Donde: D(m06): Diagnóstico en la marca temporal de seis meses. f(sc,m06): Conjunto de Datos de entrenamiento pertenecientes a las marcas temporales de sc y m06

Este experimento contó con un total de 555,045 estas se dividen de la siguiente manera: Visita SC que consta de 283,554 imágenes en total y esta se divide:

- ✔ AD: Esta clase contiene 69,594 imágenes (23,198 por plano)
- ✔ MCI: Esta clase 135,378 imágenes (45,126 por plano).
- ✔ CN: Esta clase 78,582 imágenes (26,194 por plano).

Visita m06 que consta de 252,948 imágenes en total y este se divide:

- ✔ AD: Esta clase 60,750 imágenes (20,250 por plano).
- ✔ MCI: Esta clase 120,549 imágenes (40,183 por plano).
- ✔ CN: Esta clase 71,649 imágenes (23,883 por plano).

Donde en el modelo que posee el historial clínico tiene:

- ✔ 129576 imágenes pertenecen al conjunto de entrenamiento.
- ✔ 32394 imágenes pertenecen al conjunto de validación.
- ✔ 16864 imágenes pertenecen al conjunto de prueba.

Y el modelo que sin el historial clínico tiene:

- ✔ 53961 imágenes pertenecen al conjunto de entrenamiento.

- ✔ 13491 imágenes pertenecen al conjunto de validación.
- ✔ 16864 imágenes pertenecen al conjunto de prueba.

Ambos modelos fueron configurados con los siguientes parámetros 3.6 donde podemos observar que la diferencia entre ambas reside en el conjunto de datos de entrada.

Modelo	Tasa de Aprendizaje	Optimizador	Tamaño de Lote	Fuente de Entrenamiento	K-Fold	Preentrenada
DenseNet121k5 m06	1 x 10^{-4}	Adam	64	M06	5	Sí
model diagnosis m06 sc	1 x 10^{-4}	Adam	64	Screening y M06	5	Sí

Tabla 3.6: Tabla de los modelos con y sin historial clínico y sus parámetros de entrenamiento

Tal como hemos comentado anteriormente en este documento, hay desbalanceo de clases que fueron solventadas con el uso de validación cruzada estratificada (Stratified kfold), sin embargo, esta es diferente a la anterior, ya que al trabajar con dos conjuntos de datos, se ha de operar de la forma que se muestra en la siguiente figura 3.11.

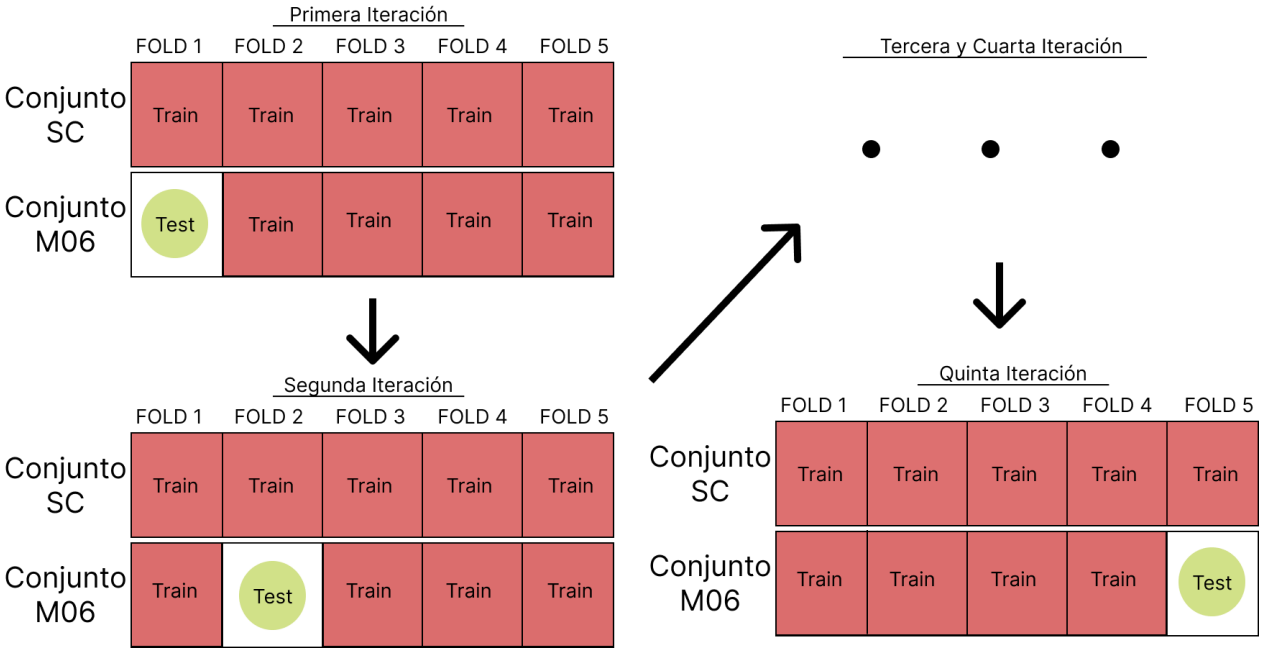


Figura 3.11: Imagen de ejemplo de StratifiedKFold utilizado en el modelo del historial clínico

Como observamos, el conjunto de datos SC siempre forma parte del conjunto de entrenamiento, y en función de la iteración en la que nos encontremos, se utilizará un fold distinto como conjunto de prueba, mientras que el resto de los datos se incorporan al conjunto de entrenamiento.

Desde el punto de vista computacional, esto se consigue agregando todas las imágenes y etiquetas del sc al conjunto de entrenamiento tal y como se puede observar en la figura 3.12

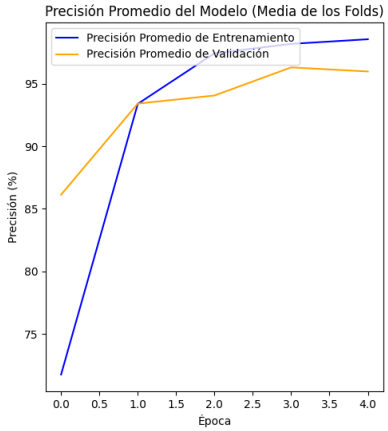
```
for fold, (train_idx, test_idx) in enumerate(kfold.split(image_paths, labels)):
    print(f"\n====== Fold {fold+1}/{k_folds} ======")

    train_paths = [image_paths[i] for i in train_idx]
    train_labels = [labels[i] for i in train_idx]
    test_paths = [image_paths[i] for i in test_idx]
    test_labels = [labels[i] for i in test_idx]


    #-----------------------------------------------------------------

    train_paths += sc_paths
    train_labels += sc_labels

    #-----------------------------------------------------------------
```

Figura 3.12: Ejemplo de división de datos en la validación cruzada con K-Fold, donde se observa la asignación de conjuntos de entrenamiento y prueba.

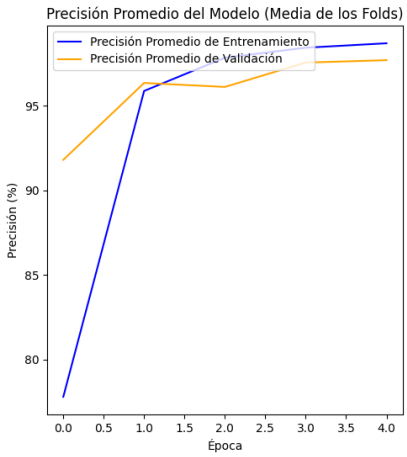
No obstante ante esto, hubo una desviación en el objetivo, ya que se vió prudente realizar un nuevo experimento, donde se entrenaría a un nuevo modelo en base al conjunto de datos exclusivo de la marca temporal de m06, para posteriormente evaluar, si el acceso al resto del historial clínico del paciente (El conjunto de datos de la visita SC), afecta significativamente a la hora de clasificar y diagnosticar AD, MCI y CN.

De este modo, a continuación se mostrará los resultados de ambos experimentos para llevar a cabo una comparación y análisis de los resultados obtenidos de las diferentes métricas.

Perfiles de las métricas precisión y pérdida durante el entrenamiento



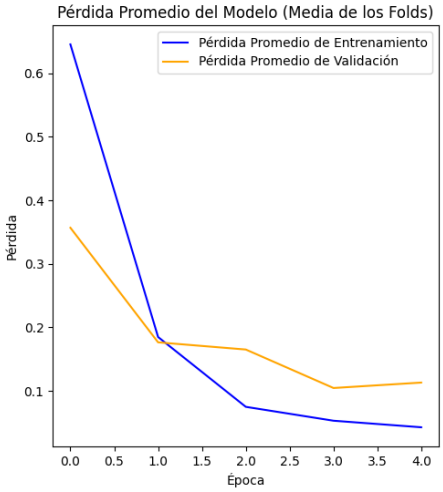
(a) Conjunto de datos m06



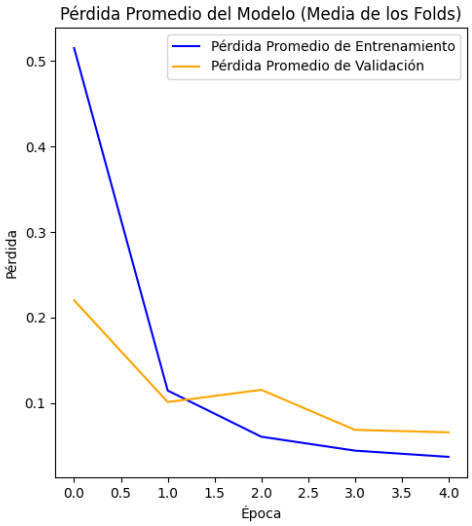
(b) Conjunto de datos sc y m06

Figura 3.13: Evolución de la precisión promedio (entrenamiento y validación) a lo largo de 5 épocas para los modelos con historial clínico parcial (sc,m06) frente a los que no incorporan historial(m06). (a) Conjunto de datos m06 y (b) conjunto de datos sc y m06.

En la figura 3.13 podemos observar las métricas de precisión de ambos modelos con distintos conjuntos de entrenamiento, pudiendo observar claramente, que ambos comparten un mismo comportamiento, no obstante, el conjunto de datos de sc y m06 tiene una curva de validación que sigue más de cerca a la curva de entrenamiento. Esto indica que el modelo tiende menos al sobreajuste que su contraparte, lo que se traduce como una mejor capacidad de generalización , no obstante ninguno de ellos presentan síntomas significativos de sobreajuste ni infraajuste.



(a) Conjunto de datos m06



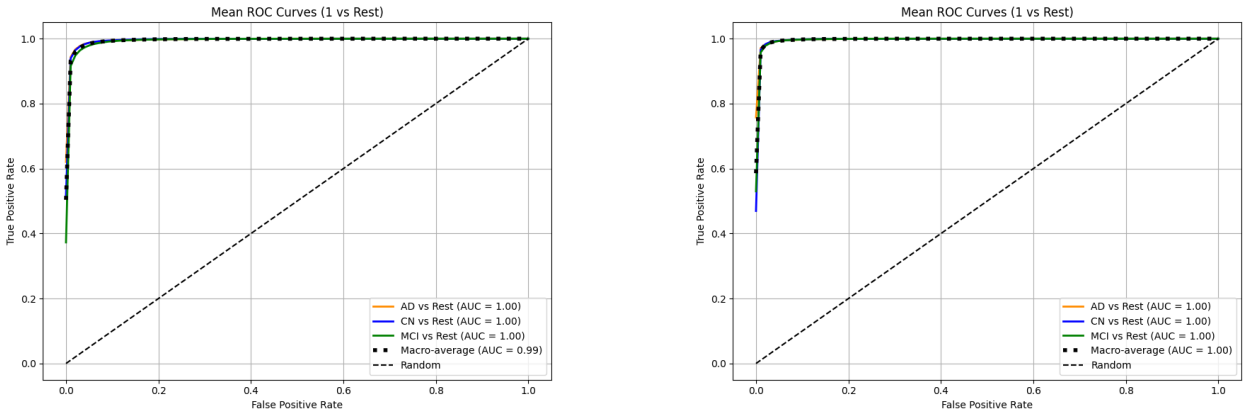
(b) Conjunto de datos sc y m06

Figura 3.14: Evolución de la pérdida promedio (entrenamiento y validación) a lo largo de 5 épocas para los modelos con historial clínico parcial (sc,m06) frente a los que no incorporan historial(m06). (a) Conjunto de datos m06 y (b) conjunto de datos sc y m06.

Por otra parte, la figura 3.14 presenta las curvas de pérdidas. Asimismo, la curva de pérdida de ambas presenta un leve pico en la segunda época que desciende con el paso de las mismas hasta valores contantes.

A diferencia de las curvas de precisión, estas curvas son prácticamente iguales, de modo que no podemos discernir a priori cual de los dos modelos con diferentes conjuntos de datos presenta unas mayores prestaciones

Curvas ROC y AUC

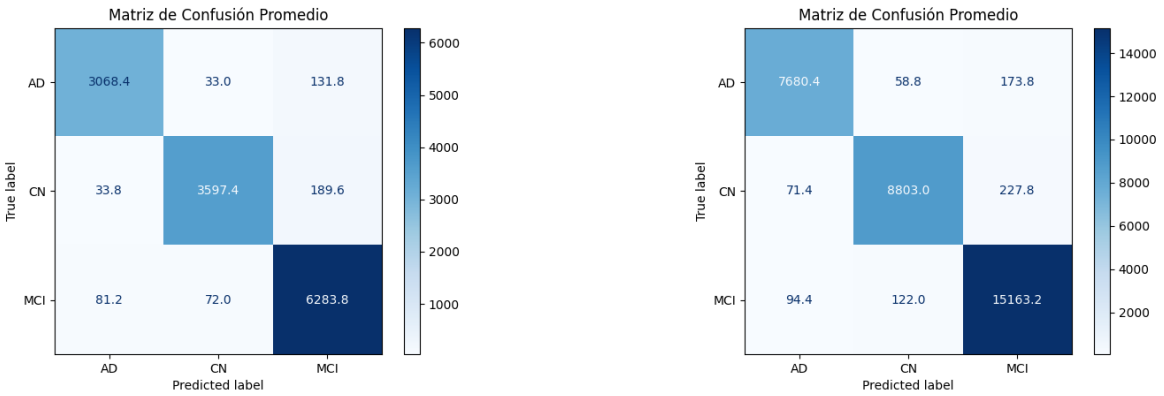


(a) Conjunto de datos m06 (b) Conjunto de datos sc y m06

Figura 3.15: Comparativa de las curvas ROC bajo la estrategia 1 vs Rest de los modelos con historial clínico parcial (sc,m06) frente a los que no incorporan historial(m06). (a) Conjunto de datos m06 y (b) conjunto de datos sc y m06.

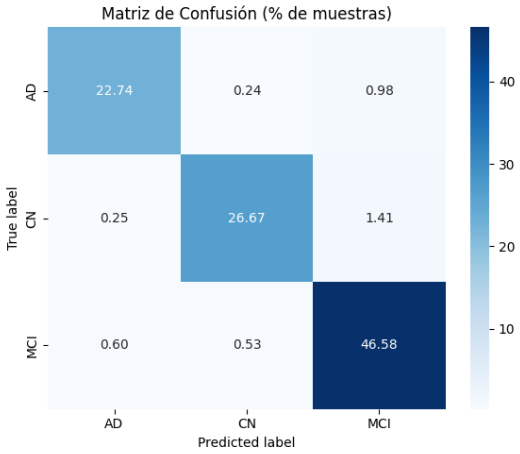
Como podemos observar en las diferentes gráficas y curvas ROC, ambos modelos rinden espectacularmente bien, no mostrando diferencia alguna entre ellos, llegando en todos los casos a un valor de AUC de 1 de media a lo largo de los 5 folds de entrenamiento. Esto mismo ocurre con los resultados del enfoque binario los cuales se pueden observar en nuestro Anexo (Repositorio GitHub). Volviendo de nuevo, a no dar ningún dato significativo que de pie a la comparación, de ambas. Demostrando así que con estos datos, ambos modelos aparentemente rinden de manera similar.

Matrices de Confusión

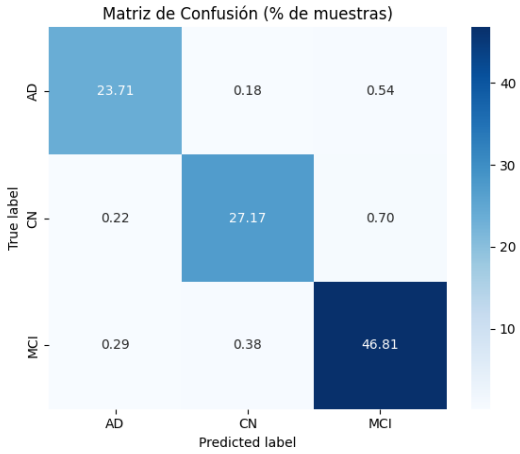


(a) Conjunto de datos m06 (b) Conjunto de datos sc y m06

Figura 3.16: Matrices de confusión promedio obtenidas del conjunto de test de los modelos con historial clínico parcial (sc,m06) frente a los que no incorporan historial(m06). (a) Conjunto de datos m06 y (b) conjunto de datos sc y m06.



(a) Conjunto de datos m06



(b) Conjunto de datos sc y m06

Figura 3.17: Matrices de confusión promedio por procentajes obtenidas del conjunto de test de los modelos con historial clínico parcial (sc,m06) frente a los que no incorporan historial(m06). (a) Conjunto de datos m06 y (b) conjunto de datos sc y m06.

Si observamos las matrices de confusión, en un principio podríamos afirmar que ambas se comportan de igual manera, no obstante, si hacemos el calculo para averiguar el porcentaje de aciertos respecto al número total de muestras. Podemos observar numéricamente, las diferencias entre ellas.

De este modo, si observamos la figura 3.17a y obtenemos los valores de su diagonal, observamos una proporción de clasificaciones correctas de 22.74 % de AD , 26.67 % de CN, 46.58 % de MCI, siendo un total de 95.99 % de aciertos. Mientras que su contraparte en la figura 3.17b presenta valores de 23.71 % para AD, 27.17 % para CN y 46.81 %para MC siendo un total de 97.69 % suponiendo así una mejora respecto al total de 1.70 % respecto al conjunto de datos de m06. Esto afirma y sugiere que el modelo con historial clínico presenta mejores clasificaciones que el modelo que no tiene historial.

Evaluación global de rendimiento

Modelo	Plano	Clase	Accu	Sens	Spec	PPV	NPV	CUI+	CUI-	AUC
DenseNet121 m06	Axial	ADvsRest	0.9793	0.9490	0.9888	0.9639	0.9840	0.9147	0.9730	1
		CNvsRest	0.9757	0.9415	0.9891	0.9716	0.9772	0.9148	0.9666	1
		MCIvsRest	0.9648	0.9762	0.9544	0.9513	0.9778	0.9287	0.9332	**1**
	Coronal	ADvsRest	0.9882	0.9738	0.9928	0.9768	0.9918	0.9512	0.9846	1
		CNvsRest	0.9837	0.9732	0.9879	0.9695	0.9894	0.9435	0.9774	1
		MCIvsRest	0.9800	0.9787	0.9811	0.9794	0.9805	0.9586	0.9620	1
	Sagital	ADvsRest	**0.9956**	**0.9884**	**0.9979**	**0.9932**	**0.9963**	**0.9817**	**0.9942**	**1**
		CNvsRest	0.9934	0.9852	**0.9967**	**0.9915**	0.9942	0.9769	0.9908	1
		MCIvsRest	0.9922	0.9950	0.9897	0.9888	**0.9954**	0.9839	0.9852	1
model diagnosis m06 sc	Axial	ADvsRest	0.9877	0.9706	0.9932	0.9789	0.9905	0.9501	0.9838	1
		CNvsRest	0.9852	0.9671	0.9922	0.9799	0.9872	0.9477	0.9796	1
		MCIvsRest	0.9809	0.9859	0.9764	0.9742	0.9871	0.9605	0.9638	1
	Coronal	ADvsRest	0.9925	0.9847	0.9950	0.9844	0.9951	0.9693	0.9900	1
		CNvsRest	0.9913	0.9827	0.9946	0.9860	0.9933	0.9690	0.9880	**1**
		MCIvsRest	0.9885	0.9889	0.9882	0.9871	0.9899	0.9762	0.9782	1
	Sagital	ADvsRest	0.9952	0.9882	0.9974	0.9920	0.9962	0.9802	0.9936	1
		CNvsRest	**0.9953**	**0.9932**	0.9962	0.9902	**0.9973**	**0.9835**	**0.9935**	1
		MCIvsRest	**0.9933**	**0.9931**	**0.9935**	**0.9929**	0.9937	**0.9860**	**0.9873**	**1**

Tabla 3.7: Comparación de métricas de rendimiento para DenseNet121 (k=5) en configuraciones *m06* y *diagnosis m06 sc*, segmentadas por plano y clase.

Como podemos apreciar en la tabla 3.7 se observa que la inclusión del historial clínico mejora de manera consistente el rendimiento del modelo, independientemente del plano o la clase. En particular, las métricas como la precisión, la sensibilidad y la especificidad tienden a incrementarse ligeramente cuando se utiliza información longitudinal, mostrando porcentajes sólidos por encima del 98 % en todas ellas. Este comportamiento reafirma la relevancia del historial clínico para la tarea de clasificación, sugiriendo que la evolución temporal del paciente aporta señales diagnósticas adicionales y complementarias.

Asimismo, el plano sagital continúa destacándose como el más informativo entre los tres evaluados, ofreciendo los valores más altos en todas las métricas, tanto en el escenario con m06 únicamente como en el que incluye screening. Esto sugiere que la representación anatómica en ese plano contiene patrones relevantes que favorecen una mejor discriminación entre las clases diagnósticas. Pudiendo diagnosticar y clasificar de manera más eficiente.

3.3.2. Diagnóstico m12

Para este experimento queremos hacer un diagnóstico en la marca temporal m12. Este experimento sigue la siguiente fórmula

$$D(m12) = f(sc, m06, m12)$$

Donde:

D(m12): Diagnóstico en la marca temporal de seis meses.

f(sc,m12): Conjunto de Datos de entrenamiento pertenecientes a las marcas temporales de sc ,m06 y m12.

Este experimento contó con un total de 742,455 imágenes, compuestas por todos los conjuntos de datos mostrados anteriormente y por la visita m12: Visita m12 que consta de 195,953 imágenes en total y este se divide:

- ✔ AD: Esta clase 30,804 imágenes (15,402 por plano).
- ✔ MCI: Esta clase 102,009 imágenes (34,003 por plano).
- ✔ CN: Esta clase 57,738 imágenes (19,246 por plano).

Donde en el modelo que posee el historial clínico tiene:

- ✔ 187,003 imágenes pertenecen al conjunto de entrenamiento.
- ✔ 46,751 imágenes pertenecen al conjunto de validación.
- ✔ 13,731 imágenes pertenecen al conjunto de prueba.

Y el modelo sin historial clínico tiene:

- ✔ 43,936 imágenes pertenecen al conjunto de entrenamiento.
- ✔ 10,984 imágenes pertenecen al conjunto de validación.
- ✔ 13,731 imágenes pertenecen al conjunto de prueba.

Asimismo, al igual que el anterior, ambos modelos fueron configurados con los siguientes parámetros 3.8.

Modelo	Tasa de Aprendizaje	Optimizador	Tamaño de Lote	Fuente de Entrenamiento	K-Fold	Preentrenada
DenseNet121k5 m12	1 x 10^{-4}	Adam	64	M12	5	Sí
model diagnosis m06 sc m12	1 x 10^{-4}	Adam	64	Screening, M06 y m12	5	Sí

Tabla 3.8: Tabla de los modelos m12 con y sin historial clínico y sus parámetros de entrenamiento

Por otro lado, el esquema de validación cruzada estratificada que utilizamos, es levemente diferente al anterior. A diferencia del diagnóstico de la visita m06. Ahora el conjunto de datos correspondiente a las visitas m06 y sc forman parte del conjunto de entrenamiento, tal y como podemos visualizar en la figura 3.18.

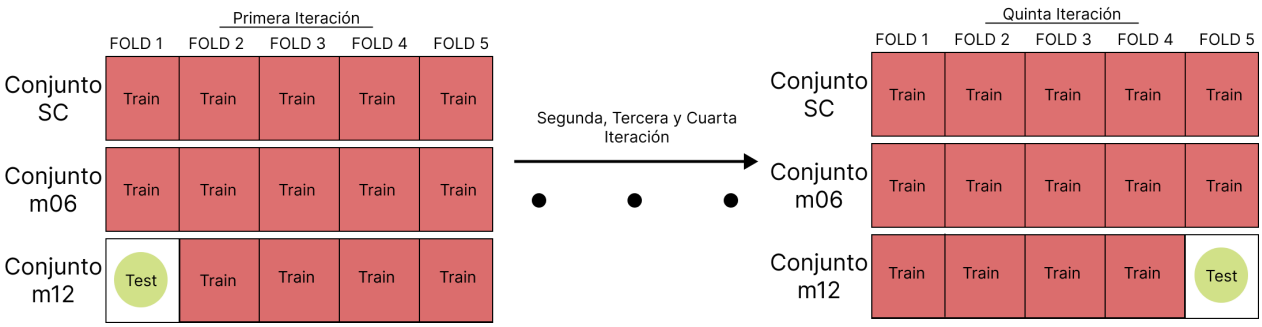
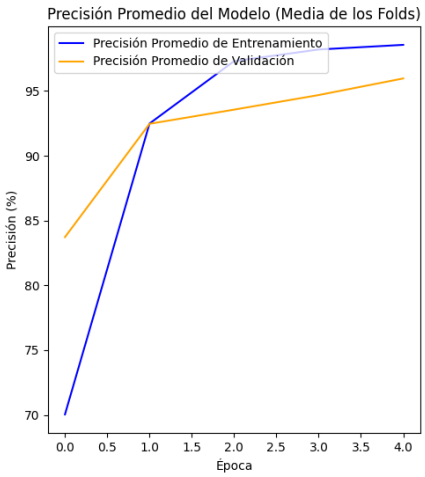


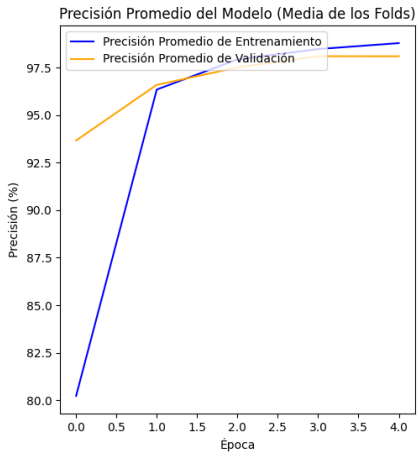
Figura 3.18: Imagen de ejemplo de StratifiedKFold utilizado en el modelo del historial clínico de m12

Además al igual que el experimento anterior, este estudio tuvo una desviación respecto al objetivo original, por los mismos motivos expuestos anteriormente, por lo que se procederá a exponer las gráficas comparativas entre un modelo sin historial clínico en la marca temporal m12 y uno con el historial clínico completo.

Perfiles de las métricas precisión y pérdida durante el entrenamiento



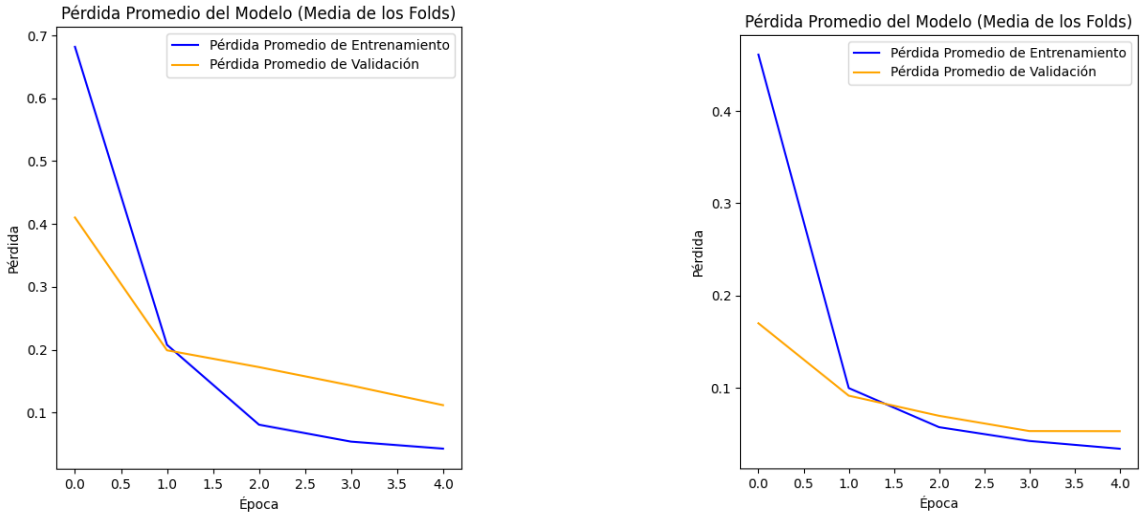
(a) Conjunto de datos m12



(b) Conjunto de datos sc,m06 y m12

Figura 3.19: Evolución de la precisión promedio (entrenamiento y validación) a lo largo de 5 épocas para los modelos con historial clínico completo (sc,m06 y m12) frente a los que no incorporan historial (m12). (a) Conjunto de datos m12 y (b) Conjunto de datos sc,m06 y m12.

Como podemos ver en la figura 3.19b se presenta una curva de precisión que aumenta por encima del 97.5 %, seguida de cerca por una curva de validación de porcentajes de acierto similares. No obstante, este comportamiento difiere en la figura 3.19a ya que la curva de validación no acompaña a la curva de entrenamiento, sugiriendo así que tiene un sobreajuste respecto a los datos de validación. Indicando así, que el modelo con el conjunto de datos completo rinde y generaliza mejor que su contraparte.



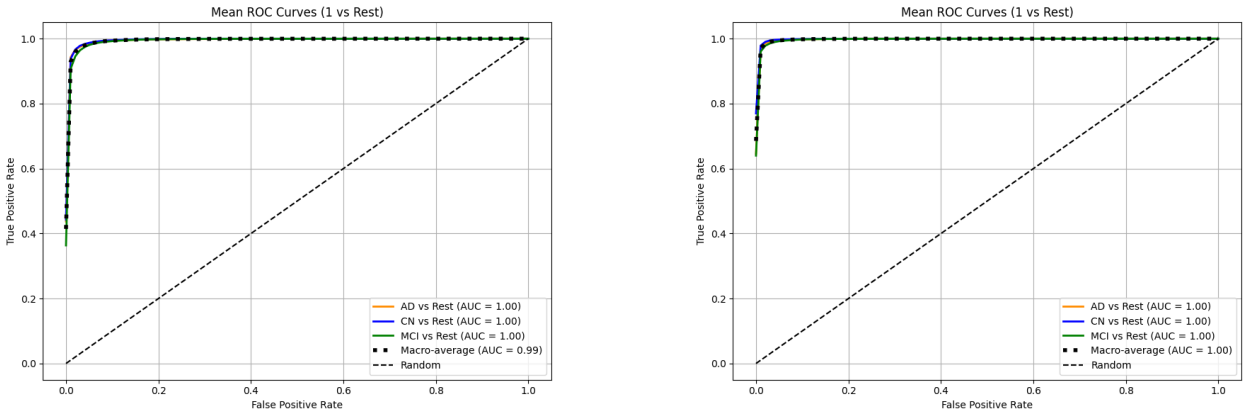
(a) Conjunto de datos m12

(b) Conjunto de datos sc,m06 y m12

Figura 3.20: Evolución de la pérdida promedio (entrenamiento y validación) a lo largo de 5 épocas para los modelos con historial clínico completo (sc,m06 y m12) frente a los que no incorporan historial (m12). (a) Conjunto de datos m12 y (b) Conjunto de datos sc,m06 y m12.

Al igual que las figuras de precisión la figura 3.20 presenta diferencias bastante claras entre ambos modelos. Donde la diferencia reside en la mejor capacidad de adaptación y generalización del modelo, mostrando en la 3.20b una curva de aprendizaje y de validación similares disminuyendo de manera estable y constante durante el entrenamiento. Mientras que en comparativa, el modelo entrenado exclusivamente con los datos pertenecientes a la marca temporal de m12, no presenta una curva que represente tan bien la generalización del modelo a lo largo del entrenamiento.

Curvas ROC y AUC

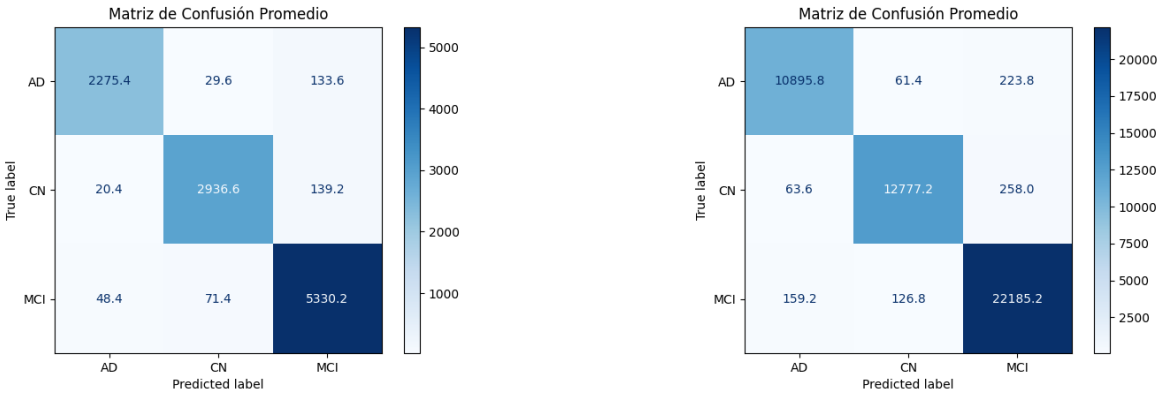


(a) Conjunto de datos m12 (b) Conjunto de datos sc,m06 y m12

Figura 3.21: Comparativa de las curvas ROC bajo la estrategia 1 vs Rest obtenidas de los modelos con historial clínico completo (sc,m06 y m12) frente a los que no incorporan historial (m12). (a) Conjunto de datos m12 y (b) Conjunto de datos sc,m06 y m12.

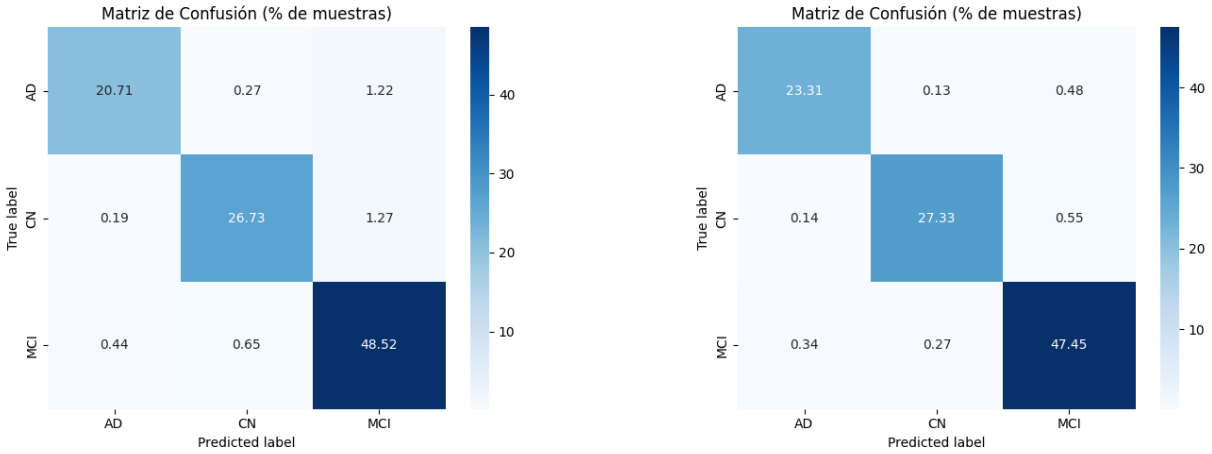
En el apartado de las ROC, todas las figuras difieren en los mismos valores, mostrando al igual que en el experimento anterior, el buen desarrollo y consolidación de los modelos respecto a su tarea. Por lo que al contar con valores de AUC iguales a uno, de nuevo, representa el comportamiento casi perfecto de ambos modelos.

Matrices de Confusión



(a) Conjunto de datos m12 (b) Conjunto de datos sc,m06 y m12

Figura 3.22: Matrices de confusión promedio obtenidas del conjunto de test de los modelos con historial clínico completo (sc,m06 y m12) frente a los que no incorporan historial (m12). (a) Conjunto de datos m12 y (b) Conjunto de datos sc,m06 y m12.



(a) Conjunto de datos m12

(b) Conjunto de datos sc,m06 y m12

Figura 3.23: Matrices de confusión promedio por porcentajes obtenidas del conjunto de test de los modelos con historial clínico completo (sc,m06 y m12) frente a los que no incorporan historial (m12). (a) Conjunto de datos m12 y (b) Conjunto de datos sc,m06 y m12.

En cuanto a las matrices de confusión estas presentan resultados interesantes de cara a la comparativas entre ambos modelos, al igual que en los anteriores análisis las matrices de confusión predeterminadas muestran que ambos modelos operan correctamente. No obstante si hacemos la misma operación comentada anteriormente veremos las principales diferencias entre ellas.

En estas podemos observar, que por primera vez en todos estos experimentos el conjunto de datos base, sin historial clínico, presenta un mejor porcentaje de aciertos en la clase MCI con un porcentaje de 48.52 % frente al 47.45 % del modelo con historial. No obstante, este es en lo único que destaca, ya que los porcentajes de las clases AD y CN son superiores en su contraparte.

Evaluación global de rendimiento

Modelo	Plano	Clase	Accu	Sens	Spec	PPV	NPV	CUI+	CUI-	AUC
DenseNet121 m12	Axial	ADvsRest	0.9789	0.9331	0.9919	0.9707	0.9811	0.9057	0.9732	1
		CNvsRest	0.9763	0.9485	0.9872	0.9668	0.9799	0.9169	0.9674	1
		MCIvsRest	0.9643	0.9780	0.9507	0.9513	0.9777	0.9304	0.9295	1
	Coronal	ADvsRest	0.9841	0.9591	0.9914	0.9703	0.9881	0.9306	0.9797	1
		CNvsRest	0.9839	0.9677	0.9902	0.9745	0.9875	0.9430	0.9779	1
		MCIvsRest	0.9770	0.9814	0.9727	0.9724	0.9816	0.9544	0.9548	1
	Sagital	ADvsRest	0.9926	0.9797	**0.9963**	**0.9873**	0.9941	0.9672	0.9905	1
		CNvsRest	0.9893	0.9715	0.9961	0.9897	0.9892	0.9614	0.9853	1
		MCIvsRest	0.9876	**0.9944**	0.9808	0.9810	**0.9943**	0.9755	0.9753	1
diagnosis m06 sc m12	Axial	ADvsRest	0.9891	0.9745	0.9937	0.9800	0.9920	0.9550	0.9858	1
		CNvsRest	0.9891	0.9754	0.9944	0.9855	0.9905	0.9613	0.9849	1
		MCIvsRest	0.9836	0.9873	0.9802	0.9787	0.9881	0.9663	0.9685	1
	Coronal	ADvsRest	0.9917	0.9818	0.9948	0.9832	0.9943	0.9654	0.9892	1
		CNvsRest	0.9922	0.9860	0.9946	0.9862	0.9945	0.9724	0.9892	1
		MCIvsRest	0.9882	0.9882	0.9883	0.9874	0.9890	0.9757	0.9774	1
	Sagital	ADvsRest	**0.9956**	**0.9957**	0.9956	0.9859	**0.9987**	**0.9817**	**0.9942**	1
		CNvsRest	**0.9958**	**0.9899**	**0.9981**	**0.9952**	**0.9960**	**0.9852**	**0.9942**	1
		MCIvsRest	**0.9945**	0.9934	**0.9956**	**0.9952**	0.9939	**0.9886**	**0.9895**	1

Tabla 3.9: Comparación de métricas de rendimiento para DenseNet121 (k=5) en configuraciones *m12* y *diagnosis m06 sc m12*, segmentadas por plano y clase.

A lo largo de este experimento hemos podido observar, que el modelo que cuenta con la información clínica presenta mejores datos, Como podemos ver en las métricas de rendimiento en la tabla 3.9 se puede ver que de forma global, los resultados reflejan un rendimiento altamente preciso en ambas configuraciones, con valores cercanos a la perfección en métricas como la precisión, la sensibilidad, la especificidad y el AUC. No obstante, se observa una mejora sistemática al incorporar el historial clínico completo, destacándose en todas las clases, obteniendo prácticamente los mejores resultados en todas las métricas. Esta mejora sugiere que la progresión temporal de los datos aporta señales adicionales relevantes para el diagnóstico.

Una vez más, el plano sagital se consolida como el más informativo, alcanzando los valores más altos en la mayoría de las métricas. Por ejemplo, con historial completo, el modelo logra una sensibilidad del 99.57 % para la clase AD, y una precisión superior al 99.5 % en todas las clases, mostrando un equilibrio notable entre sensibilidad y especificidad.

Estos resultados reafirman la importancia de integrar información longitudinal en el entrenamiento de modelos de diagnóstico, ya que permite capturar mejor la evolución de la enfermedad y mejora la capacidad del modelo para distinguir entre las diferentes clases diagnósticas. Además, la consistencia del desempeño en los tres planos evidencia la robustez de la arquitectura propuesta.

	Accu	Sens	Spec	PPV	NPV	CUI+	CUI-	AUC
diagnosis m06 sc	0.9953	0.9931	0.9962	0.9929	0.9973	0.9860	0.9936	1
diagnosis m06 sc m12	0.9956	0.9957	0.9981	0.9952	0.9987	0.9886	0.9942	1

Tabla 3.10: Tabla comparativa entre los diagnósticos m06 y m12

Finalmente, si observamos la tabla 3.10 podemos observar ambos diagnósticos y sus métricas de rendimiento. Los resultados siguen la misma tendencia observada en experimentos anteriores, mostrando una ligera mejora en todas las métricas cuando se utiliza el modelo que incorpora el historial clínico completo, en comparación con el modelo cuya fuente de datos contiene dicha información de manera parcial.

Capítulo 4

Conclusiones y trabajo futuro

4.1. Conclusiones

- ✔ Los objetivos propuestos han sido alcanzados, así como los objetivos derivados:
  - Se ha desarrollado un modelo de inteligencia artificial basado en DenseNet capaz de procesar imágenes MRI para el diagnóstico anticipado del Alzheimer, cuya arquitectura base es DenseNet121 preentrenado.
  - Se ha realizado un estudio comparativo entre el modelo desarrollado y el modelo MyGNG [25].
  - Se han implementado modelos cuyos datos de entrada se componen del historial clínico completo del paciente (sc, m06 y m12) y se ha comparado frente a modelos que sólo utilizan una visita como conjunto de datos de entrada, denotando la importancia del historial clínico en la clasificación del sistema.
- ✔ Los resultados del sistema software desarrollado han sido superiores en todas las métricas al 97 %, destacando entre ellas un 99.25 % de precisión, y un 99.87 % de especificidad, los cuales consideramos buenos resultados en comparación con el estado de la cuestión.
- ✔ El modelo construido en base a la arquitectura DenseNet121 ha alcanzado resultados mejores respecto a MyGNG con un 98.28 % de precisión superando a MyGNG un 13.9 %, Además, supera a MyGNG en sensibilidad con un incremento del 8.1 %, en especificidad con un aumento del 22.8 %, así como en el AUC, donde se observa una mejora del 15.5 %. Asimismo, los índices CUI+ y CUI- han mejorado en un 27.3 % y 33.2 % respectivamente, en comparación con los resultados de MyGNG[25].
- ✔ Con relación al estudio realizado de cómo afecta la transferencia de aprendizaje entre los modelos DenseNet, los resultados muestran que el modelo DenseNet121 preentrenado presenta frente al no preentrenado una mejora en todas las métricas de rendimiento. Especificamente en la precisión (Accu), la sensibilidad (Sens), la especificidad (Spec), el valor predictivo positivo (PPV), el valor predictivo negativo (NPV), así como en los

índices CUI+ y CUI-. Además se observa que el valor de la AUC es de 0.99 en todos los planos y estados cognitivos. Por tanto, el modelo seleccionado como base del resto de experimentos fue el preentrenado, ya que refleja mejor rendimiento independientemente del plano.

- ✔ Los resultados también muestran que el sistema cuya fuente de datos incorpora el historial clínico completo del paciente, es decir, las imágenes correspondientes a todas las visitas (Screening, 6 meses y 12 meses), presentan una precisión (Accu) del 0.9956 respecto al modelo sin historial que decrece 0.9926 en la clasificación de ADvsRest. La sensibilidad (Sens) mejora de 0.9797 a 0.9957 y la especificidad (Spec) de 0.9963 a 0.9956 respecto con los modelos que solo utilizan información de una única visita. Este hecho, también se aprecia en la comparación entre el historial clínico parcial y completo que mejora de la precisión de 99.53 % del modelo parcial a 99.56 % del modelo completo. Esto denota, que la agregación del historial clínico completo como entrada de datos del sistema resulta en un sistema de software más preciso y robusto.
- ✔ A lo largo de todos los experimentos y en base a sus resultados, se observa que el plano Sagital es el plano que obtiene mejores resultados frente a los demás, representando así que es el plano con mayor impacto en la clasificación final.
- ✔ El desarrollo del sistema se vio muy favorecido por la metodología seguida, que fue la metodología en espiral, la cual permitió que el TFG se desarrollase conforme a lo establecido, haciendo hincapié en la utilidad de la selección de una metodología adecuada de cara al desarrollo e investigaciones.

4.2. Trabajo futuro

En vista de los resultados obtenidos, el rendimiento del sistema desarrollado es muy alto. Esto constituye una oportunidad para el desarrollo de un sistema clínico completo para el diagnóstico anticipado de la enfermedad del Alzheimer, que ayude a los facultativos a clasificar de manera automática y eficiente a sus pacientes, mejorando así la calidad de vida de los mismos al tomar medidas durante las étapas más tempranas de la enfermedad, así como aliviar el peso de la enfermedad de sus seres queridos.

Asimismo, esta arquitectura podría abarcar otros ámbitos médicos donde se solventen la necesidad de la asignación y clasificación automática de muestras clínicas, tales como la detección de células cancerígenas en las muestras de sangre, detección de micro fracturas en los huesos, etc. Promoviendo el buen uso de la IA, respetando la información privada de los pacientes dentro de los códigos morales y éticos.

Bibliografía

[1] Real Academia Española. *Demencia*. https://www.rae.es/diccionario-estudiante/demencia. 2024.

[2] Zeinab Breijyeh y Rawan Karaman. «Comprehensive Review on Alzheimer's Disease: Causes and Treatment». En: *Molecules* 25.24 (2020), pág. 5789.

[3] M. Herrera-Rivero et al. «Enfermedad de Alzheimer: inmunidad y diagnóstico». En: *Revista de Neurología* 51.3 (2010), págs. 153-164.

[4] S. Tiwari et al. «Alzheimer's disease: pathogenesis, diagnostics, and therapeutics». En: *International Journal of Nanomedicine* 14 (2019), págs. 5541-5554. DOI: 10.2147/IJN.S200490.

[5] S. Esmaeilzadeh et al. «End-To-End Alzheimer's Disease Diagnosis and Biomarker Identification». En: *Proceedings of Machine Learning in Medical Imaging (MLMI 2018)*. Vol. 11046. Lecture Notes in Computer Science. Epub 2018 Sep 15. Springer, 2018, págs. 337-345. DOI: 10.1007/978-3-030-00919-9_39.

[6] Laura Alejandra Mayorga Cadavid y Andrés M. Pérez-Acosta. *Una aproximación de la literatura científica sobre la relación entre reconocimiento de emociones, deterioro cognitivo y demencias*. https://www.urosario.edu.co. Universidad del Rosario (Bogotá, Colombia). Categoría: Divulgación. Etiquetas: investigación, literatura, emociones, envejecimiento, deterioro cognitivo. Mayo de 2018.

[7] A. Ward et al. «Rate of Conversion from Prodromal Alzheimer's Disease to Alzheimer's Dementia: A Systematic Review of the Literature». En: *Dementia and Geriatric Cognitive Disorders Extra* 3.1 (2013), págs. 320-332. DOI: 10.1159/000354370.

[8] G. Livingston et al. «Dementia prevention, intervention, and care». En: *The Lancet* 390.10113 (2017), págs. 2673-2734. DOI: 10.1016/S0140-6736(17)31363-6.

[9] Román Alberca Serrano y Secundino López-Pousa. *Enfermedad de Alzheimer y otras Demencias*. 4.ª ed. Editorial Médica Panamericana, 2011, pág. 640. ISBN: 9788498355345.

[10] Reisa A Sperling et al. «Toward defining the preclinical stages of Alzheimer's disease: Recommendations from the National Institute on Aging-Alzheimer's Association workgroups on diagnostic guidelines for Alzheimer's disease». En: *Alzheimer's & Dementia* 7.3 (2011), págs. 280-292. DOI: 10.1016/j.jalz.2011.03.003.

[11] Waleed Al Shehri. «Alzheimer's Disease Diagnosis and Classification Using Deep Learning Techniques». En: *PeerJ Computer Science* (dic. de 2022). Research article; temas: Artificial Intelligence, Computer Vision, Neural Networks.

[12] A. P. Porsteinsson et al. «Diagnosis of Early Alzheimer's Disease: Clinical Practice in 2021». En: *The Journal of Prevention of Alzheimer's Disease* 8.3 (2021), págs. 371-386.

[13] Ylermi Cabrera-León et al. «Neural Computation-Based Methods for the Early Diagnosis and Prognosis of Alzheimer's Disease Not Using Neuroimaging Biomarkers: A Systematic Review». En: *Journal of Alzheimer's Disease* 98.3 (2024). Creative Commons License (CC BY-NC 4.0), págs. 793-823. DOI: 10.3233/JAD-231271.

[14] Alzheimer's Disease Neuroimaging Initiative. *ADNI*. https://adni.loni.usc.edu/. [En línea]. 2024.

[15] Alberto Serrano-Pozo et al. «Neuropathological alterations in Alzheimer disease». En: *Cold Spring Harbor Perspectives in Medicine* 1.1 (2011), a006189. DOI: 10.1101/cshperspect.a006189.

[16] Laith Alzubaidi et al. «Review of deep learning: concepts, CNN architectures, challenges, applications, future directions». En: *Journal of Big Data* 8.1 (2021), pág. 53. DOI: 10.1186/s40537-021-00444-8. URL: https://doi.org/10.1186/s40537-021-00444-8.

[17] Fan Zhang et al. «A Single Model Deep Learning Approach for Alzheimer's Disease Diagnosis». En: *Neuroscience* 491 (2022), págs. 200-214. ISSN: 0306-4522. DOI: https://doi.org/10.1016/j.neuroscience.2022.03.026. URL: https://www.sciencedirect.com/science/article/pii/S030645222200149X.

[18] Gao Huang et al. «Densely connected convolutional networks». En: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2017.

[19] PyTorch Developers. *PyTorch Documentation*. https://pytorch.org/docs/stable/index.html. 2025.

[20] Scikit-learn Developers. *Scikit-learn: Machine Learning in Python – User Guide*. https://scikit-learn.org/stable/user_guide.html. 2025.

[21] TorchVision Developers. *TorchVision Documentation*. https://docs.pytorch.org/vision/stable/index.html. 2025.

[22] Xiangrui Li et al. «The first step for neuroimaging data analysis: DICOM to NIfTI conversion». En: *Journal of Neuroscience Methods* 264 (2016), págs. 47-56. ISSN: 0165-0270. DOI: https://doi.org/10.1016/j.jneumeth.2016.03.001. URL: https://www.sciencedirect.com/science/article/pii/S0165027016300073.

[23] Neuroimaging Informatics Technology Initiative. *NIfTI-1 Data Format*. https://nifti.nimh.nih.gov/nifti-1/. 2025.

[24] NiBabel Developers. *NiBabel: Access a cacophony of neuro-imaging file formats*. https://nipy.org/nibabel/. 2025.

[25] Ylermi Cabrera-León et al. «Towards an Intelligent Computing System for the Early Diagnosis of Alzheimer's Disease based on the Modular Hybrid Growing Neural Gas». En: *Journal Title* XX.X (2024), págs. 1-14. DOI: 10.1177/ToBeAssigned.

[26] Alzheimer's Disease Neuroimaging Initiative (ADNI). *ADNI Data Search – LONI Image and Data Archive*. https://ida.loni.usc.edu/pages/access/search.jsp?project=ADNI&tab=advSearch&page=SEARCH&subPage=NEW_ADV_QUERY. 2024.

[27] Anitha Raju et al. «Deep Learning Image Processing Technique for Early Detection of Alzheimer's Disease». En: *International Journal of Advanced Science and Technology* 107 (oct. de 2017), págs. 85-104. DOI: 10.14257/ijast.2017.107.07.

[28] Skelarn. *Skelarn Documentation*. https://scikit-learn.org/stable/modules/cross_validation.html#stratified-k-fold. [En línea]. 2024.